MT REVIEW

FLEX-3000 SDR Transceiver Review, Part 1

The future of radio in your shack today

By Larry Van Horn, N5FPW
Monitoring Times Review Editor

In the past two annual ham radio issues of *Monitoring Times* (the May editions), I have authored a couple of features about how the world of digital communications has been embraced within the amateur radio community.

Digital communications have rapidly grown to be among the most popular modes of operation in the HF bands. Thanks to increasingly powerful computers, we continue to expand the boundaries of digital communications used within the amateur radio service. Digital modes that were invented by hams and tested in the amateur radio bands have now found their way into other portions of the radio spectrum, and are even being used by government and military agencies on a regular basis.

But one area in which the digital revolution has not made significant inroads into our radio hobby community is in the equipment through which we communicate. The next logical step in our digital evolution is to incorporate digital electronic techniques in the radios we use for transmitting and receiving radio signals.

But, to some, especially "us" older radio hobbyists, that word "digital" and "equipment" can strike fear in our hearts. The thought of using some sort of fancy, all digital radio to receive and transmit radio communications seems like something written in a SciFi movie script for most of us. So, try as we may, sometimes it is still hard to get an old dog to try a new trick when the words "digital" and "equipment" are spoken in the same breath.

I am happy to report that recently I definitely learned a few new tricks when the opportunity to review the FlexRadio Systems FLEX-3000™ SDR transceiver came my way.

But first, before I dive deep into this two-part equipment review on the FLEX-3000, a bit of education on this SDR subject is definitely in order.

❖ SDR – what is that?

SDR is an acronym for software defined radio (SDR). The technical staff at FlexRadio provides the following excellent definition:

"A software defined radio is one where the radio frequency (RF) signal is converted to a digital bit stream and all of the modulation and demodulation of the signal is done with digital signal processors (DSPs). An SDR performs significant amounts of signal processing in a general purpose computer, or a reconfigurable piece of digital electronics. The goal of this design is to produce a radio that can receive and transmit a new form of radio protocol just by running new software."

In a nutshell, a software-defined radio system is a radio communication system where components that have typically been implemented in hardware (e.g., mixers, filters, amplifiers, modulators/demodulators, detectors, etc.) are instead implemented using software. While the concept of SDR is not new, the rapidly evolving capabilities of digital electronics are making practical many processes that were once only theoretically possible.

A basic SDR system may consist of a personal computer (PC) equipped with a sound card, or other analog-to-digital converter, preceded by some form of RF front end hardware. Significant amounts of signal processing are handed over to a general-purpose processor, rather than being accomplished using special-purpose hardware. Such a design produces a radio that can receive and transmit widely different radio protocols (sometimes referred to as a waveforms) based solely on the software used.

So using the concept above, in theory, if you wanted to communicate using the JT65 mode, or any of the other dozens of digital modes, that digital protocol could be part of a transceiver's software package. So all you would have to do would be to select a button marked JT65 and tune in a signal to either decode or transmit it to another station via our digital radio described above.

Unfortunately we aren't quite there yet, since no one has incorporated that capability in an all-in-one software package to date. But we are there in a fundamental sense, in that I can run a third party piece of software to do the decoding job while running my SDR software for my radio, simultaneously on one computer.

The basic SDR concept which led to the development of the first SDR experimenter's kit for ham radio was first described by FlexRadio Systems CEO Gerald Youngblood, K5SDR, in *QEX* magazine during the summer of 2002.

Gerald's four-part article on the concepts and techniques was used to develop the world's first SDR for ham radio operators and it is still the quintessential primer on SDRs. You can check out all four of these highly informative, light reading primers online at the FlexRadio company website (Adobe Acrobat PDF format) via the following links:

- Part one introduces DSP and how it is applied to SDRs along with describing a transceiver architecture. **www.flex-radio.com/Data/Doc/qex1.pdf**
- Part two describes the initial software engineering needed to define an SDR. **www.flex-radio.com/Data/Doc/qex2.pdf**
- Part three illustrates the use of DSP along with using a PC sound card to define a functional SDR. **www.flex-radio.com/Data/Doc/qex3.pdf**
- Part four is a detailed description of the three board stack that was to become the ground breaking SDR-1000. **www.flex-radio.com/Data/Doc/qex4.pdf**

So, now that we have been introduced to the SDR concept, what is all this hype I have mentioned about the FlexRadio 3000a SDR? It is time to take a closer look.

❖ SDR versus Traditional Transceiver

What are the advantages of a FlexRadio SDR over a traditional transceiver?

There are many advantages, so let me count the ways of just a few of them to illustrate why FlexRadio SDRs are so cool to own and operate.

First, the DSP code is not "fixed" in firmware like its hardware-based transceiver cousins. That makes upgrading the DSP hardware easy. New radio or operating features are easily implemented with a software upgrade.

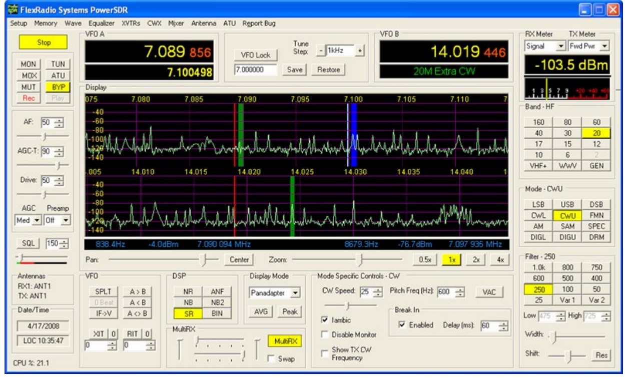
That means that the radio is constantly being improved. It never becomes obsolete. Try doing that with a conventional hardware transceiver – You can't!

Also, this DSP software code is open source. It is not proprietary so you can do a bit of experimenting on your own if you have the knowledge to do such things. More about this subject later.

There is only a single step or conversion from RF to baseband audio. Thus, you will have less equipment noise generated due to eliminated multiple IF conversions. Also, there will be low distortion. Distortion is introduced at every conversion stage in a radio, and since we only have one in the FlexRadios – low distortion.

Another plus of this single conversion is that roofing filters are not required to improve performance. A "roofing filter" is simply a filter in the radio's first intermediate frequency (IF) stage through which all signals must pass before they will be "seen" by later receiver stages. Remember, as I mentioned above we have done away with the need for a "roofing filter" because we have done away with the traditional IF stage found in conventional radios.

Finally, 99 percent of the signal path is entirely in the digital domain. In the Flex-Series of radios the receiver performance is directly related



to the dynamic range of the analog-to-digital converters (ADCs) utilized. Radio frequency signals are down converted to the audio frequency band, which is sampled by a high performance audio frequency ADC. By using these embedded high performance ADCs, we have a product that provides higher dynamic range and is more resistant to noise and RF interference.

Since most of our signal path is in the digital realm, the FlexRadio SDR software performs all of the demodulation, filtering (both radio frequency and audio frequency), signal enhancement (equalization and binaural presentation).

So, what are the major differences between an SDR and a traditional radio that you are going to notice?

There are no knobs and buttons on the transceiver to manipulate. All of the radio control is done via software, so functions such as changing frequency, selecting filters, changing bands are no longer initiated on the radio hardware itself. The hardware is less complex due to the elimination of circuits that would normally be in traditional radios. Basic radio functions are now handled by the SDR software. Also, since very high quality A/D and D/A converters are used, SDRs outperform all traditional radios on both transmit and receive.

Oh, and if that no knob thing is a sticking point for you old timers, you can get a knob as an accessory. I'll have more about that in part two of this review.

❖ Things You Need to Know

Before you jump out there and slap down that credit card to by one of these modern digital marvels, there are a few things you need to know and decide, in order to operate your FLEX-3000 or any of the other FlexRadio SDRs.

The first and most important decision you will need to make before you buy any of the SDRs is what computer are you going to use to run the SDR software. Since we rely on the computer to replace a significant portion of the hardware that is traditionally in a hardware-based transceiver, you won't be able to pick up just any old hamfest flea market special to do this part of the computing for you. You're going to need a computer with some real computing power if you are going to effectively run the main star of the FlexRadio show – their FlexRadio PowerSDR™ software package.

The FlexRadio PowerSDR software provides all DSP and hardware control functions for FlexRadio System's fully software defined radios and is released under the open source General Public License (GPL).

Written in a combination of ANSI C and C# computer languages, the FlexRadio PowerSDR software is easy to learn and modify. Yes, I said modify. The source code for this program is openly available to encourage amateur SDR research and experimentation. For those of us who are not computer programmers, (and for whom the mere thought of programming makes us break out in a cold sweat), the complete application is provided in compiled form so that you can simply download, install and run it. Wipe the sweat off your brow now.

FlexRadio PowerSDR software will run on a variety of personal computers. The transceiver's digital-to-analog converter (DAC)/ADC can operate at 48-, 96- or 192-kHz ,which is known as the sampling rate. This sampling rate is hardware dependent, so a particular hardware platform may not support all sampling rates.

Why is this sampling rate important? The higher the sampling rate, the larger the spectrum bandwidth you can sample and view. Also, higher sampling rates can reduce processing latency. Processing latency is the slight delay between receiving the RF signal and having it converted to an audio frequency (AF), which is what you hear from the headphones or speakers. The higher the sampling rate, the greater the computing resources needed to run at that speed.

There is one other parameter in addition to sampling rate which can affect a computer's performance, and that is the hardware/audio buffers. Smaller audio buffers produce less latency, but require more computing resources.

Fast computers (multiple core central processing units or CPUs) that can be purchased today, can run at the highest sampling rates with the lowest buffer settings. Older computers with less processing power will run very well, but may have to use lower sampling rates and/or larger buffers to achieve acceptable performance.

Since this is a multiple dependency system, it is difficult to state definitively a "recommended" computer configuration. But in a nutshell, just about any new computer you buy today with a dual-core CPU (AMD or Intel) with at least 1 MB of L2 cache RAM will be more than sufficient for running the PowerSDR software package.

Cache memory is different than the main RAM of your personal computer. Cache RAM, the processor's local temporary storage area, is located on the CPU. More cache RAM means less trips to the main RAM, speeding high speed operations like DSP functions.

A great website for checking out your computer's CPU performance is located at **www.cpubenchmark.net**. The higher PassMark rated CPUs will speed up your FlexRadio performance. PowerSDR will run on CPUs with a PassMark rating of 400, but if you are buying a new system, consider purchasing a unit that has a CPU measurement with at least a 1000 rating.

FlexRadio does offer on their website a "Knowledge Center" article that describes several factors that need to be taken into account when choosing a PC to use with their FlexRadio PowerSDR. These suggestions can also be used to evaluate your current computer system. You can access that article online at **http://kc.flex-radio.com/KnowledgebaseArticle50063.aspx**.

The FlexRadio Rule of Thumb is this: "Get the highest performance PC you can afford and get one that allows you to upgrade if necessary."

❖ Firewire Throughput is Critical

One of the more critical elements for determining if a computer will run the PowerSDR software package is the throughput achieved from the Firewire host controller. Most computers today come with an integrated Firewire interface. In some cases, these integrated peripherals haven't been optimized for high throughput data rates.

Using a bus-connected Firewire host controller card that is PCI or PCI-E based is recommended for an optimum data throughput. For laptops, you want to get an ExpressCard Firewire host controller rather than a PCMCIA if at all possible. You can get additional details on this subject on the FlexRadio website at **http://kc.flex-radio.com/KnowledgebaseArticle50179.aspx**.

One final note about Firewire, FlexRadio will be shipping you a 1394a 6-pin to 6-pin Firewire cable about 1 meter in length. You will need to know what sort of 1394 pin you will have on your card or computer and make plans accordingly. I had a 4-pin 1394 connector on my laptop, so I needed to get a 6-pin to 4-pin cable. Tech support at FlexRadio Systems told me that they recommend a cable over a pin converter for stability purposes.

The following cables, connectors and software were included with the FLEX-3000 I reviewed: PowerSDR Software CD with all necessary software and documentation to run your Flex Radio; 13.8 VDC power cable – one end terminated with the FLEX-3000 power connector and the other unterminated; BNC (male) to SO-239 connector for connecting coax with PL-239 connectors; and the aforementioned 1394 Firewire cable.

One final thing you will have to have ahead of time is a microphone. The FLEX-3000 and FLEX-1500 radios use an 8-pin modular connector or "jack" for interfacing a microphone to the transceiver. This connector is a modular "8P8C" (8 position/8 contacts) connector. A corresponding 8P8C plug is needed, commonly referred to as an RJ-45 plug or connector, for connecting a microphone to the radios. This plug is commonly used for twisted pair Ethernet cabling applications. The FLEX-3000/1500 uses the same connector pinout as the Yaesu FT897, FT857 and FT817. You can get more info on this subject online at **http://kc.flex-radio.com/KnowledgebaseArticle50063.aspx**.

If you need a different cable, microphone cables, microphone, or a tuning knob, FlexRadio has an online store where these items can be bought as you make your SDR purchase.

So, as the folks at FlexRadio say, "Tune in the excitement"™ with their radios. We are now to the point where we are ready to tune in the excitement. Join us for Part Two of this review in the June issue of *Monitoring Times,* as we as put the FLEX-3000 on the air during a big international amateur radio DX contest.