



SDR— Boosting the Flexibility of RF Communication

Simon Holt, Mouser Electronics



Abstract

Software-defined radio (SDR) represents a break from traditional radio frequency (RF) design techniques. Older, conventional systems could only implement fixed functions and encompass a limited feature set. By taking advantage of flexible RF front-ends and high-performance digital hardware, developers can exploit novel techniques to eke more capacity out of the wireless spectrum and build highly differentiated systems. Using SDR, engineers can build radios that can react more readily to interference, employ more sophisticated channel-coding schemes to increase data rates and take advantage of other advanced RF technologies. Access to an ecosystem of hardware and software, including open-source libraries, means it is easier than ever to gain experience in SDR design.



Since the launch of the 2G cellular networks more than 20 years ago, RF design has evolved significantly. In contrast to those early networks, agility is now a key requirement for system design, as operators and users seek to make the most of the RF spectrum available to them.

When the first digital cellular networks were launched, makers of terminals, base stations and other RF equipment needed to support only a limited selection of frequency bands in each region. When 3G appeared, terminal manufacturers had to take into account a larger range of frequency bands - so they could sell devices that could work in most locations around the world. The arrival of 4G and the Long-Term Evolution (LTE) protocol saw the number of possible frequency bands increase to more than 40. The imminent arrival 5G networks will increase the level of complexity still further, with an even wider choice not just of operator-licensed frequency bands but unlicensed spectrum.

Even within a single mobile band, there are numerous protocols that can be employed to transfer data and Wi-Fi is a prime example. For instance, it needs to coexist with Bluetooth and many other protocols in the increasingly overcrowded 2.4GHz band. These protocols do not differ just in the binary packet formats they employ but in how the data is converted into analogue signals, amplified and subsequently transmitted. Each of those decisions affects the design of the transmit and receive subsystems until the digital symbols injected into the transmit stream are finally decoded and recovered.

A key aspect of RF communication is the modulation of a stream of data symbols onto a carrier signal. The carrier signal typically oscillates at a frequency that is much higher than the rate at which the data symbols are updated. This makes it possible to constrain the bandwidth of the RF signal to within a range of frequencies. The symbols occupy sidebands: frequency ranges that may lie above or below the core carrier frequency. These sidebands comprise the Fourier components of the data symbols when modulated onto the carrier.

The modulation may use one of a variety of schemes. The simplest is amplitude modulation, as used in the earliest broadcast radios. This alters the intensity of the carrier and has the effect of spreading the range of frequencies occupied by the signal. Frequency modulation modifies the frequency of the carrier directly. Phase modulation alters the amplitude of the carrier in such a way that it shifts the effective phase of the signal. Quadrature amplitude modulation uses two carrier signals that are separated by a phase angle of 90°. Although both signals are transmitted together on the one channel, they can be extracted by coherent demodulation as two independent signals because the two carriers are orthogonal to each other.

In principle, it is possible to create the modulated signal directly with a digital-to-analogue converter (DAC) and then use an analogue-to-digital (ADC) in the receive chain to recover the carrier information. After this, software algorithms can be employed to analyse the signal so as to recover the original data symbols. It is only comparatively recently that this digital approach to RF processing - one that allows generation and analysis to be performed in software - has become feasible for a wide range of radio systems.



Traditionally, because most signals call for the ability to handle inputs operating at frequencies of hundreds of Megahertz and into the Gigahertz domain, RF processing has demanded the use of fixed-function circuitry. Direct conversion was impossible because ADCs and DACs were unable to operate at such high frequencies.

Techniques such as heterodyning were used to provide the digitiser with more manageable frequencies. Heterodyne conversion dates back to the early 20th Century and relies on the combination of two high-frequency signals producing two more centred on different frequencies. Typically, one output is at the sum of the two mixed frequencies; the other is at the difference between the two. In the case of down conversion, a local oscillator is tuned to generate an intermediate frequency from the difference signal. The sum is filtered out.

Although the heterodyne technique is simple to understand and is readily implemented in the analogue domain, it tends to rely on precisely matched components tuned to a comparatively limited range of input frequencies. The filters are also chosen for specific frequency ranges. Although devices such as surface-acoustic wave (SAW) filters can offer strong rejection characteristics, suitable for cutting interference from signals at adjacent frequencies, they need to be chosen for specific applications and offer only restricted tune-ability. Very often, the filters will be switched in and out based on the needs of the receiver. To handle a large range of frequencies and signal types, a traditional transceiver design based on a heterodyne architecture demands a large number of discrete components and filters, which increases PCB area utilised and the overall cost.

In recent years, a number of design techniques have come to the fore that have made it possible to increase the flexibility of RF transceivers and move towards the point where direct sampling can be carried out with almost all the processing required being performed within the digital domain. This presents engineers with true SDR functionality. Programmable RF technology provides the means by which to build highly flexible communications systems that can cope with a multitude of frequencies and protocols, as well as delivering enhanced transmission efficiency.

Digital domain processing can improve the overall quality of the transmitted signal. For example, the data sent to a DAC that is used to generate a signal ready for local-oscillator mixing can be processed to prevent leakage of unwanted DC components into the output. More advanced applications can include the generation and processing of complex protocols, such as those that use spread-spectrum techniques to increase security or maximise the bandwidth from the available spectrum.



Although it is at an early stage, SDR offers many opportunities for developing cognitive radio systems. These can sense the RF usage environment around them and adjust their operation to avoid interference in a way that maximises usable bandwidth. The protocols can dynamically adjust channel frequencies, bandwidth and signalling techniques to avoid potential jammers. Other possibilities that are starting to be explored in relation to 4G and 5G are those of beamforming and the use of multiple-input, multiple-output (MIMO) antenna arrays. MIMO makes it possible to exploit spatial diversity. Spatial diversity increases the overall quality of the incoming signal, making it possible to decode data from low-power sources that suffer from problematic interference. MIMO is also used during transmission to alter the signal sent over each antenna, with the objective of improving the chances of successful decoding by the receiver. With larger antenna arrays, MIMO makes it possible to use beamforming - dynamically steering signals towards the intended receiver. As well as improving reception, beamforming reduces the chance of interference with other users and improves transmission security.

With much of the functionality moved to software, developers can take advantage of a growing ecosystem of open-source tools and libraries to build up experience in SDR and avoid the need to reinvent the wheel for commonly used RF functions. Initiatives such as GNU Radio provide readymade blocks that can be readily tuned to the target application. The GNU Radio platform features filters, channel codes, synchronisation elements, equalisers, demodulators, vocoders, decoders and many other types of functional block. It also defines ways to connect the blocks and manage the flow of data between them.

In terms of hardware, there are two main groups of components required for effective SDR design. One group represents the processing horsepower required to manipulate RF signals in the digital domain. Highly parallelised microprocessors and field-programmable gate arrays (FPGAs) have been developed that possess the throughput necessary to deal with signals in the multi-Megahertz domain. Xilinx has incorporated ADCs and DACs into its MPSoC UltraScale FPGAs in order to make it easier to deploy SDR solutions. The elevated levels of integration not only reduce PCB complexity but curb power consumption as well. Conventional designs use ADCs and DACs with JESD204 serial links to pass digitised data to and from FPGAs and multicore processors. The digital FPGA cores found in devices like the UltraScale now include DSP blocks that are optimised for RF processing, such as digital mixing and filtering. Typically, the fabric comprises multiple DSP blocks that perform the multiply-accumulate operations common to many filtering and signal processing tasks.

The FPGA array provides the flexibility to split samples for processing by multiple DSP blocks and interleave the results for the output stream at high speed. The programmable logic also enables highly customised processing for special-purpose operations. Higher level protocols can be handled by Arm processors that coexist with the programmable logic fabric on the same die. Fabrics such as these, that offer support for a wide range of digital formats, make it possible to handle advanced applications like massive MIMO. These systems are sensitive to phase noise, but higher precision floating-point arithmetic helps overcome the problems caused by this noise.



For front-end processing, manufacturers have taken advantage of the integration possibilities provided by advanced CMOS processors to make it easier to down sample from a full RF signal to one that is suitable for passing to ADCs operating at a high speed, but not at the frequency of the RF source. These devices contain front-end circuitry that can tune the local-oscillator and heterodyne processes in the analogue domain to the needs of different RF signals and the channels contained within them. An example of this is the AD9363 from Analog Devices. This can process RF signals in the 325MHz to 3.8GHz range, which covers most licensed and unlicensed bands, working with channel bandwidths of less than 200kHz up to 20MHz. Each receive subsystem includes independent automatic gain control, DC offset correction, quadrature correction and digital filtering, eliminating to have these functions performed in the digital domain. Two ADCs per channel digitise the received I and Q signals, then passes them through configurable decimation filters and 128-tap finite impulse response (FIR) filters to produce a 12-bit output signal and feed into digital processing.

There is a growing choice of development boards and kits aimed at SDR that provide the user with an easy way to gain experience in the various techniques and technologies. They can support different combinations of processors, FPGAs and host computers, providing RF access through front-end devices such as the AD9363. One example is the ADALM-PLUTO SDR Active Learning Module. This provides users with a combination of the AD9363, software processing and programmable logic for the development, testing and implementation of SDR applications. The module features independent transmit and receive signals capable of operating in full-duplex. It can acquire and generate RF analogue signals from 325MHz to 3.8GHz, passing digital data to and from the intermediate frequency channels at up to 61.44MSamples/s. With driver support for OS X, Windows and Linux, the module lets users work on a variety of compute platforms.

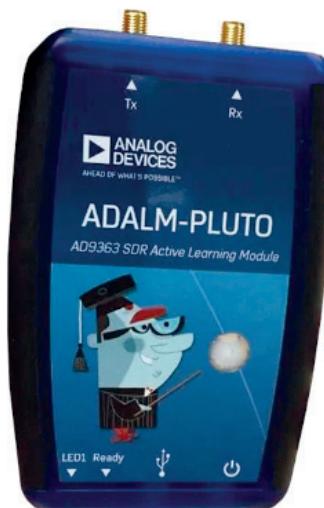


Figure 1: ADALM-PLUTO SDR Active Learning Module from Analog Devices

The Crowd Supply Lime SDR mini-boards are hardware platforms based on a combination of the Intel MAX 10 FPGA and Lime Microsystems' LMS7002M RF programmable transceiver. Despite being small, the Lime SDR mini-boards have been demonstrated as part of a complete LTE 4G base station design, streaming live video between two mobile phones.



Figure 2: Crowd Supply Lime SDR Board.

A number of SDR development kits have been introduced that fit into open-source hardware ecosystems - such as Beaglebone. The KiwiSDR, for example, gives Beaglebone processor RF interfaces in the 10kHz to 30MHz range. For rapid prototyping with personal computers, the HackRF One attaches to a USB 2.0 port and provides access to RF channels of up to 10MHz anywhere between 1MHz and 6GHz.



The growing availability of development boards and kits targeted at SDR deployment demonstrates the way in which these techniques for RF communications are becoming increasingly mainstream. Many more developers are now able to take advantage of SDR and work on customised communications systems for unlicensed bands and, where they have legal access, licensed bands.



mouser.com