

Chapter 2

What Is a Software-Defined Radio?

Historically, radios have been designed to process a specific waveform.¹ Single-function, application-specific radios that operate in a known, fixed environment are easy to optimize for performance, size, and power consumption. At first glance most radios appear to be single function—a first-generation cellular phone sends your voice, while a WiFi base station connects you to the Internet. Upon closer inspection, both of these devices are actually quite flexible and support different waveforms. Looking at all the radio devices in my house, only the garage door opener and the car key fob seem to be truly fixed. With this introduction, clearly a software-defined radio's main characteristic is its ability to support different waveforms.

The definition from wireless innovation forum (formerly SDR forum) states [3]:

A software-defined radio is a radio in which some or all of the physical layer functions are software defined.

Let us examine each term individually:

- The term physical layer requires a bit of background. Seven different *layers* are defined by the Open Systems Interconnection (OSI) model [4], shown in Table 2.1.

This model is a way of subdividing a communications system into smaller parts called layers. A layer is a collection of conceptually similar functions that provide services to the layer above it and receives services from the layer below it. The layer consisting of the first four blocks in Fig. 1.1 is known as the *physical* layer.

- The broad implication of the term *software defined* is that different waveforms can be supported by modifying the software or firmware but not changing the hardware.

¹ The term *waveform* refers to a signal with specific values for all the parameters (e.g., carrier frequency, data rate, modulation, coding, etc).

Table 2.1 OSI seven-layer model

	Data unit	#	Name	Function
Host layers	Data	7	Application	Network process to application
		6	Presentation	Data representation and encryption
		5	Session	Interhost communication
Media layers	Segment	4	Transport	End-to-end connections and reliability
	Packet	3	Network	Path determination, logical addressing
	Frame	2	Data Link	Physical addressing
	Bit	1	Physical	Media, signal, and binary transmission

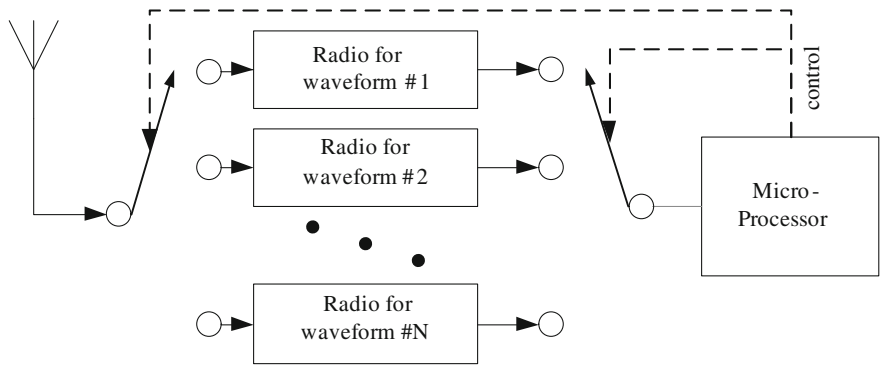


Fig. 2.1 Basic software-controlled radio

According to the strictest interpretation of the definition, most radios are not software *defined* but rather software *controlled*. For example, a modern cellular phone may support both GSM (2G) and WCDMA (3G) standards. Since the user is not required to flip a switch or plug in a separate module to access each network, the standard selection is controlled by software running on the phone. This defines the phone as a software-controlled radio. A conceptual block diagram of such a radio is shown in Fig. 2.1. Software running on a microcontroller selects one of the single-function radios available to it.

A simple thought experiment shows that the definition of a true SDR is not quite as black and white as it appears. What if instead of selecting from a set of the complete radios, the software could select one of the building blocks shown in Fig. 1.1? For example, the software would connect a particular demodulation block to a decoder block. The next logical step calls for the software to configure details of the demodulator. For example, it could choose to demodulate QPSK or 8PSK symbols. Taking this progression to an extreme, the software could define interconnect between building blocks as simple as registers, logic gates, and multipliers, thereby realizing any signal processing algorithm. Somewhere in this evolution, the software-controlled radio became a software-defined radio.

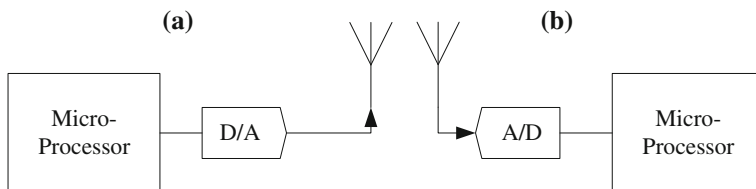


Fig. 2.2 Ideal software-defined radio: (a) transmitter, (b) receiver

The key, albeit subtle, difference is that a software-*controlled* radio is limited to functionality explicitly included by the designers, whereas a software-*defined* radio may be reprogrammed for functionality that was never anticipated.

The ideal software-defined radio is shown in Fig. 2.2. The user data is mapped to the desired waveform in the microprocessor. The digital samples are then converted directly into an RF signal and sent to the antenna. The transmitted signal enters the receiver at the antenna, is sampled and digitized, and finally processed in real time by a general purpose processor. Note that the ideal SDR in contrast with Fig. 1.1, does not have an RFFE and a microprocessor has replaced the generic DSP block. The ideal SDR hardware should support any waveform at any carrier frequency and any bandwidth.

So, what challenges must be overcome to achieve this elegant radio architecture?

- Most antennas are mechanical structures and are difficult to tune dynamically. An ideal SDR should not limit the carrier frequency or bandwidth of the waveform. The antenna should be able to capture electromagnetic waves from very low frequencies (e.g., <1 MHz) to very high frequencies (e.g., >60 GHz²). Section 10.5 detail the challenge of designing such an antenna. Such a wideband antenna, if available, places high demands on the RF front end (RFFE) and the digitizer.
- Selection of the desired signal and rejection of interferers (channel selection) is a key feature of the RFFE. However, the antenna and filter(s) required to implement the channel selection are usually electromechanical structures and are difficult to tune dynamically (see Sect. 10.4).
- Without an RF front end to select the band of interest, the entire band must be digitized. Following Nyquist's criterion, the signal must be sampled twice at the maximum frequency (e.g., 2×60 GHz). Capabilities of currently available A/D converters are discussed in Sect. 10.2.3, and are nowhere close to 120 GHz.
- The captured spectrum contains the signal of interest and a multitude of other signals, as shown in Fig. 2.3. Interfering signals can be much stronger than the signal of interest.³ A power difference of 120 dB is not unreasonable. The

² 60 GHz is the highest frequency used for terrestrial commercial communications in 2011.

³ Consider a practical example of a cell phone transmitting at +30 dBm while receiving a -60 dBm signal.

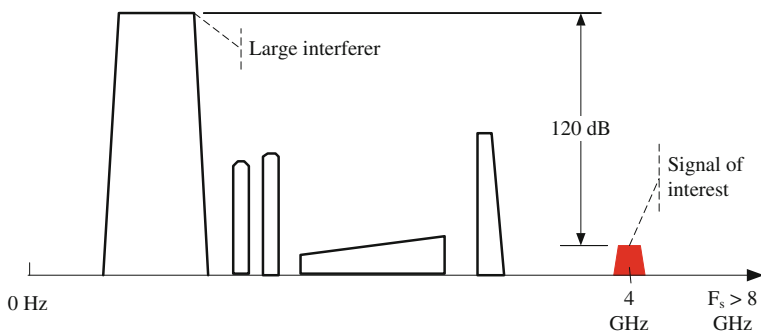


Fig. 2.3 Digitizing the signal of interest and adjacent bands

digitizer must have sufficient dynamic range to process both the strong and the weak signals. An ideal digitizer provides about 6 dB of dynamic range per bit of resolution. The digitizer would then have to provide well over 20 bits of resolution (e.g., 20 to resolve the interferer and more for the signal of interest).

- The digitizer must be very linear. Nonlinearity causes intermodulation between all the signals in the digitized band (see Fig. A-31 in Sect. A.9.6). Even a high order intermodulation component of a strong signal can swamp a much weaker signal.
- In the extreme example discussed so far (a 24 bit digitizer operating at 120 GHz) real-time digital signal processing has to be applied to a data stream at $120 \times 10^9 \times 24 \approx 250$ GB/s. This is beyond the capabilities of modern processors and is likely to remain so in the foreseeable future.

Assuming all of these technical problems were solved,⁴ the same radio could be used to process any existing and expected future waveforms. However, it does not mean that radio is optimal or suitable for a given application. The ideal SDR may be perfect for a research laboratory, where physical size and power consumption are not an issue, but completely inappropriate for a handheld device. The next few chapters will deal with the implementation tradeoffs for different market segments.

Some of the earliest software-defined radios were not wireless. The soft modems used in the waning days of dial-up implemented sophisticated real-time signal processing entirely in the software.

⁴ Note that the A/D converter with these specifications violates the Heisenberg uncertainty principle and is therefore not realizable. The maximum A/D precision at 120 GSPS is limited to ~ 14 bits [157] (see also footnote 6 of Chap. 10)



<http://www.springer.com/978-1-4419-9331-1>

Implementing Software Defined Radio

Grayver, E.

2013, XVIII, 270 p., Hardcover

ISBN: 978-1-4419-9331-1