SOFTWARE RADIO TECHNOLOGY AND CHALLENGES

John Chapin, CTO
Alok Shah, Director of Engineering
Vanu, Inc.
{jchapin, abshah}@vanu.com

ABSTRACT

This paper provides an overview of software radio and its current state in the industry. Software radio is a technology in which all of the waveform processing, including the physical layer, of a wireless device moves into software. If designed properly, this approach leads to dramatically improved device flexibility, software portability, and reduced development costs. Of course, such a technology brings with it numerous challenges, from hardware components to power constraints to the regulatory environment.

KEYWORDS

Software radio, wireless communications, interoperability, multi-standard devices, software portability.

INTRODUCTION

In a combat zone in Grenada in 1983, a US officer stepped into a phone booth and placed a call to North Carolina. He asked his contact there to get a message to a unit near him in Grenada, one that belonged to a different branch of the US military.[1] This was the officer's highly unorthodox (and now legendary) solution to a problem that bedevils military and public safety operations worldwide: radios that cannot communicate with each other. The different US military branches had acquired radios that used incompatible communications standards.

Software radio was developed to solve this interoperability problem. Traditional radios use hardware circuits, fixed at the time of manufacture, to perform the high-speed signal processing tasks that convert back and forth between user data and the radio waveform. Software radio exploits advances in components such as digital signal processors and field-programmable gate arrays to make the hardware generic, and move all of the waveform-specific tasks into software. One software radio device can support a variety of communications standards, just as one PC can run a variety of software applications. Software radio has a number of benefits in addition to improving interoperability.

- *Reduced cost to upgrade fielded systems* .The communications standard used by a device is field-upgradeable through software downloads, over the air if desired. The economics of expensive infrastructure systems are particularly improved, since the cost of the hardware and deployment can now be amortized over a longer lifetime.

- *Shared infrastructure.* A single hardware platform can be shared dynamically among multiple uses, with channel resources shifting among different communications standards as the load shifts. This significantly reduces the cost of infrastructure to support a mixture of legacy and newly deployed fixed-standard radio devices, as long as the duty cycles of the individual devices are acceptably low.
- *Better support for users with special needs.* A developer can modify the communications standard of a device without investing in a new hardware design. Users who need relatively small volumes of devices, for whom the cost of custom hardware is prohibitive, gain the ability to improve their operations with devices optimized to their special needs. For example, a user may need to gather telemetry from equipment that uses a locally-developed or obsolete radio system. A manufacturer may need to improve communications in a facility whose particular electromagnetic noise environment makes standard waveforms perform poorly.
- *Reduced standards risk.* A user can deploy expensive infrastructure or large numbers of mobile devices without locking in the communications standard that will be used. This insulates the user from potential changes in the standard and from market uncertainty. For example, an auto manufacturer who wishes to offer telematics (internet to the car) must commit to a hardware platform years in advance of vehicle introduction. Without software radio the manufacturer has to guess which data standard consumers will desire, which is a risky bet given the rate of change in the telecom industry.

The variety of clear benefits of software radio technology has led to substantial governmental funding for a decade, and commercial investment in the last several years.

HARDWARE TECHNOLOGY

Figure 1 shows the components of a generic digital radio platform.

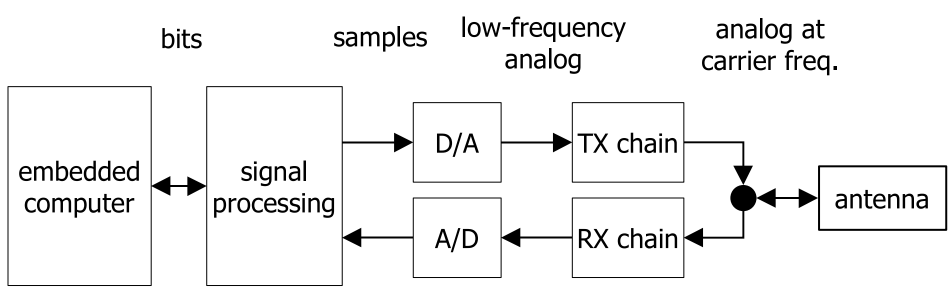


Figure 1. Hardware components of a digital radio

The embedded computer handles the higher layers of the communications task, such as encoding voice or other analog signals as a stream of bits, packetization, routing, and retransmission after error. It exchanges a stream of bits with a signal processing stage, sometimes called a modem, which implements the physical layer of the communications task. For transmission, the signal processing stage computes a sequence of digital samples that represent the data bits according to the rules of the waveform in use. For reception, the signal processing stage analyzes the received samples and estimates the user data bits that were transmitted. The A/D and D/A data conversion stages perform the usual function. The transmit chain includes frequency upconversion from the low intermediate frequency output by the D/A converter to the high carrier frequency, as well as power amplification and filtering to eliminate out-of-band radiated energy. The receive chain includes filtering to reject interfering signals, low-noise amplification and frequency downconversion.

When changing from a legacy fixed-function digital radio to a software radio, potentially every component in the radio except the embedded computer must evolve, depending on the desired level of flexibility and agility.

- *Signal processing.* The signal processing stage must be software reprogrammable to implement the different algorithms required for different communications standards. This is a challenge because modern standards have extremely high computational requirements. Hence there has been a great deal of interest in using field-programmable gate arrays and other types of reconfigurable hardware as the computation engine. An AMPS/GSM dual mode cellular base station that uses digital signal processors for computation has been commercially available from AirNet for some years.[2] The most common computation engine in military software radios is one or several dedicated DSPs.[3] General-purpose processors can also be used for the task.[4]
- *Data conversion.* Radios that support multiple waveforms whose bandwidths are not the same need A/D and D/A data conversion stages with adjustable bandwidths. This is normally accomplished by combining moderately-adjustable data converters with a highly flexible digital down-converter (DDC) and digital up-converter (DUC) components. A DDC accepts a high sample-rate input stream representing a wide bandwidth signal and extracts one or a few desired bandwidth-selectable channels within it, generating a low sample-rate stream that is easier for the reprogrammable signal processing stage to handle. The DUC performs the inverse transformation for transmissions.
- *TX chain.* The transmit chain of a software radio must be tunable across a range of center frequencies and bandwidths. This is challenging because the high linearity and low level of out-of-band emissions expected by users and regulators are normally achieved through careful analog design at a known frequency and bandwidth. One promising solution under research is to predistort the signal generated by the signal processing stage with the inverse of the nonlinearities introduced by the transmit chain.
- *RX chain.* Similarly, the receive chain of a software radio must be tunable across a range of frequencies and bandwidths. The biggest challenge in the receive chain is to provide a highly selective filter that sharply rejects interfering signals just outside the selected band while having a widely tunable center frequency. The US military has funded a project (through DARPA) towards a novel MEMS-based solution.[5] The eventual result of this line of research could be a chip with many different microscopic tuning forks etched onto it to act as the filter resonators.
- *Antenna.* The antenna needs decent gain in all bands of interest. For highly flexible software radio systems, for example covering low VHF up to the commercial PCS bands at 1.9 GHz, no technology is yet known that can affordably provide sufficient gain in a single antenna across such a wide range. Therefore the most flexible systems include multiple antennas and switch among them as required.

Not all software radios need advanced technology in all of these component areas. Simpler systems use the same data conversion, transmit and receive chains, and antenna as fixed-function digital radios, but replace the dedicated circuitry of the signal processing stage with a software reprogrammable processing engine of some kind. Such a system is appropriate for applications where the frequency band and bandwidth are constrained by external factors (normally regulatory), but the user needs the flexibility of transmitting or receiving multiple waveforms.

SOFTWARE TECHNOLOGY

From the perspective of the potential user of software radios, the most important software issue is whether the implementation of the communications standard is portable across platforms and across hardware generations. If the software is not portable, the user loses the benefits of reconfigurability and field upgrades. If software is portable, the user additionally avoids lock-in to a particular hardware platform that will rapidly become obsolete as technology advances. Portability also significantly reduces the costs associated with deploying a waveform to multiple platforms. This is important for users that contract for the implementation of communications standards, such as a facility operator that needs a custom waveform to support legacy or locally-developed telemetry.

Achieving portability across platforms from multiple vendors requires standards. Standardization of the embedded computer software and operating environment is well along, while standardization of the signal processing software is much less advanced. The former is the subject of the Software Communications Architecture (SCA), developed under the sponsorship of the US military, specifically the Joint Tactical Radio Systems Joint Program Office (JTRS JPO).[6] Compliance with the SCA is mandatory for the next major set of US military radio procurements. The wide military use of this standard will encourage its adoption for commercial software radios, although it remains to be seen whether the advantages of complying with the standard sufficiently outweigh the associated costs, primarily CPU consumption, for commercial applications.

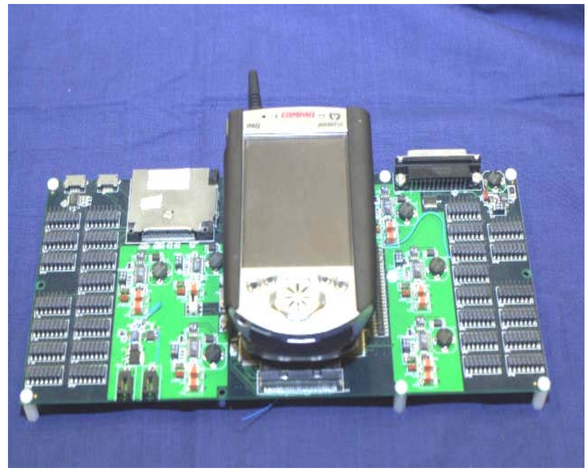
Standardization of the signal processing software operating environment is less advanced because of the high computation requirement of the signal processing task and resulting low level of the software normally written to carry it out. The software may be an FPGA map or other low-level hardware-specific construct, if the signal processing engine is not a DSP or general-purpose CPU. Even if a standard CPU is used, the code may need to be optimized for performance to the point that it is no longer portable. Efforts at standardization are just beginning within the context of the Software Defined Radio Forum.

REPRESENTATIVE IMPLEMENTATIONS

To make the above discussion more concrete, we now describe a few software radio implementations from Vanu, Inc.

Powered User Equipment: For user environments where power is available, a COTS motherboard or custom designed board using COTS components can be used. The current Vanu demo system is a standard PC with a 1 GHz Pentium III, dual-channel PC133 memory, Echotek A/D and D/A cards, and an external RF front end for frequency conversion. This system has been used to demonstrate reception and, where applicable, transmission, of legacy AM and FM broadcasts, family band walkie-talkie, APCO 25 (a North American digital law enforcement standard), 1G and 2G cellular standards, and NTSC television.

Handheld User Equipment: Vanu built a research prototype of a battery-powered handheld system for the DoD JTRS program.[7] It is a custom board with a 200 MHz StrongARM 1110, dual-port RAM, and a FPGA that interfaces to a host PDA and to a separate RF-to--digital board that provides channel selection. The system supports the IS-136 2G cellular standard in addition to less-complex legacy standards. (The wings to the left and right of the iPAQ shown are to support test probes and to create separate power supplies for major system components for test and measurement purposes.) Vanu is presently building a similar system in which the signal processing is performed natively in the PDA's StrongARM processor. The user can simply attach the RF front end to the PDA to turn it into a multi-mode radio.



Infrastructure Equipment: Vanu's proof-of-concept implementation of a cellular telephone base station consists of a COTS 1U rackmount server with two 1.8GHz AMD x86-compatible processors. Scalability is provided by gigabit ethernet links between multiple servers if required. The control component runs on a single processor and communicates to the signal processing components using CORBA. The system uses standard cluster techniques to detect and recover from the failure of any signal processing server.[8][9]

The *same software* runs on all the radically different hardware implementations of the Vanu system architecture. Furthermore, in platforms other than the handheld, all hardware except the transmit front end is acquired as COTS equipment (at the component, board or box level) that are plugged together without any hardware design work. These features make Vanu systems fundamentally different from legacy radio architectures, where the software is intimately tied to the hardware and building a new radio always starts as a hardware design problem.

BARRIERS TO ADOPTION

Software radio has already been adopted in some markets, notably military. In commercial markets, the primary barriers to adoption are cost, power consumption, and FCC certification.

- *Cost.* Unsurprisingly, existing mainstream markets focus on applications where the flexibility offered by software radio is not absolutely necessary. In such applications, the additional cost to provide reprogrammable signal processing or agile transmit and receive hardware is difficult to justify. Therefore the initial adoption of software radio is occurring in niche markets. As the volume of software radios shipped continues to increase, the cost can be expected to drop to competitive levels. However, the transmit and receive chain components pose a particularly challenging chicken-and-egg cost problem. Fixed-function radios benefit from highly integrated transceivers at very low unit cost. Most of the technology needed for a highly integrated and low cost yet agile transceiver exists, but the companies that could develop one are waiting for the market to prove its potential before investing the tens of millions of dollars needed to develop such a product. To break the chicken-and-egg cycle, software radio companies are building board-level integrated transceivers and driving their cost down in order to grow the market.

- *Power consumption.* A software reprogrammable processing engine flips more transistors than a dedicated circuit does to perform the same computation. Power consumption can easily be a factor of ten higher when comparing dedicated circuits to portable software for a DSP or GPP. This rules out software radio at present for most battery-powered applications. Hope lies in the high R&D investments that continue to be made by the major manufacturers to improve the battery life of PDAs. This effort will result in components that work equally well to provide a long battery-life software radio.
- *FCC certification.* The FCC is very supportive of software radio, as the flexibility and economic benefits associated with software radios will work to improve the overall efficiency of spectrum usage over time. However, the FCC is cautious about any new technology, and software radio has proved particularly challenging in a number of respects. The FCC is concerned that these devices not make it easier than it is at present for software faults, hackers or viruses to cause a radio to transmit out of compliance. The FCC issued its first Report & Order regarding software radio in September 2001.[10] No devices have yet been certified under the new rules so it is unclear what measures to protect against faults and hackers will be judged sufficient.

SUMMARY

Software radio is ready today for certain commercial applications. In particular, in cases where power is available, and the device either needs to communicate using multiple standards, or needs field upgradeability, or the user needs a specialized waveform in relatively low volume, software radio is likely to be the technology of choice. It seems likely that many telemetry applications would be good candidates. Ongoing progress in component technology, software standards, and the regulatory environment will steadily grow the markets to which software radio is applicable.

REFERENCES

[1] Anno, Colonel Stephen E. and Einspahr, Lieutenant Colonel William E. "Command and Control and Communications Lessons Learned: Iranian Rescue, Falklands Conflict, Grenada Invasion, Libya Raid." United States Air War College Research Report, No. AU-AWC-88-043. pg. 42.

[2] "AdaptaCell™ Broadband, Software-Defined Base Station." AirNet Communications Corporation. http://www.aircom.com/pr_adaptacell.htm

[3] Bonser, Wayne. "US defence initiatives in software radio." Software Defined Radio: Origins, Drivers and International Perspectives. Walter Tuttlebee (Ed.). John Wiley & Sons, Chichester. 2002. Chapter 2.

[4] Bose, Vanu. "Design and implementation of software radios using general purpose processors." MIT Ph.D. thesis. June 1999.

[5] Ohr, Stephan. "Darpa sows seeds of a telecom revolution." EE Times. August 4, 1997. http://www.eetimes.com/news/97/966news/sows.html.

[6] "JTRS Overview." Section 4. http://www.jtrs.saalt.army.mil/overview/section4.html.

[7] Chapin, John. "Handheld all-software radios: prototyping evaluation for JTRS Step 2b." http://www.jtrs.saalt.army.mil/docs/documents/step2b/Vanu.pdf

[8] Chapin, John, Chiu, Andrew, and Hu, Roger. "PC clusters for signal processing: an early prototype." IEEE Sensor Array and Multichannel Signal Processing Workshop. Cambridge, MA. March 2000. http://www.vanu.com/publications/sam2000pcclusters.pdf

[9] "Programmable, scalable wireless information infrastructure." NSF Contract DMI-9960454 final report. July 11, 2000.

[10] Software Defined Radio First Report and Order (FCC 01-264). Federal Communications Commission. September 13, 2001.