Volume-3, Issue-11, Nov Special Issue -2014 • ISSN No 2277 - 8160

Research Paper | Systems

# Management of Software–Defined-Radio (SDR) Through Software Communication Architecture (SCA)

**Gp Capt N THINAKARAN** — Research Scholar, Dept of ECE, Periyar Maniammai University, Thanjavur, Tamil Nadu

**Dr S JAYAKUMAR** — Research Supervisor, Periyar Maniammai University, Thanjavur, Tamil Nadu

**ABSTRACT**

*Present day Radios are deployed in civil and military organisation locally to meet the communication requirements of local subscribers in different frequency and different modes of operations. For different frequency bands, different Hardware is deployed and managed at site by specialist personnel. With the advent of Software-Defined-Radio(SDR), where in most of the functions are developed in software thru' Digital Signal Processing(DSP) Techniques using either DSP Chips or Field Programmable Gated Arrays(FPGA) or General Purpose Processor(GPP). For each mode and frequency band of operations, a waveform can be developed to bundle and reside in SDR. SDR is beneficial in remote programming and deployment. Software Communication Architecture(SCA) is published by the Joint Program Executive Office(JPEO) of the Joint Tactical Radio System(JTRS). With the adherence of Software Communication Architecture(SCA), which is standardized by Object Management Group(OMG), the SDR can be managed remotely thru' existing secure network. In this paper, I have discussed the SDR Technology, the architecture of SCA standards and how its Components are managed at Core Framework. To test these aspects, few Tools for development are suggested.*

**KEYWORDS : SDR, SCA, JPEO, JTRS**

## Introduction

1. Since World War II, the human beings have begun to use Radio communication sets. It has evolved through various developments in different frequency of operation and different modes of operations. Earlier devices were developed for their own countries utilisation. With global competition and industrialization, the radio sets have crossed across the border. In the last decade or so, every military has started interacting with all friendly foreign countries for conduct of Exercises to share the challenges and solutions there of globally. While conducting such exercises with military equipments, Communication equipments such as Radio stand aloof with **incompatibility** and **Inter-operability** problem between countries. In Radio Communication, countries use **different Frequency** ranges from HF(3 to 30 Mega Hertz) to UHF(300 MHz to 3 Giga Hertz) and **different modulation techniques** for **voice** and **data** transmission. In order to solve these challenges, an Unified Communication Architecture is required. Since anything and everything is Software definable in this world and due to rich Mathematics background in Communication field, it is desirable to explore and follow an International Standard in Communication called SCA. Radios which can be implemented using Software are called SDR. In this paper, the SDR Technology and the SCA architecture, the management aspects and few development Tools are discussed.

## Software Defined Radio(SDR)

2. The SDR Forum which is working alongwith IEEE P1900.1 group has defined SDR[1] as "Radio in which some or the entire physical layer functions are software defined". Radio is a device which transmits and / or receives signals in the Electro-magnetic Spectrum to transfer information such as voice, music, and data. Scientifically the frequencies are grouped under the following frequency bands:-

| ACRONYM | Description | Frequency Band |
|---|---|---|
| ELF | Extremely Low Frequency | 3KHz to 30 KHz |
| LF | Low Frequency | 30 KHz to 300 KHz |
| MF | Medium Frequency | 300 KHz to 3 MHz |
| HF | High Frequency | 3 MHz to 30 MHz |
| VHF | Very High Frequency | 30 MHz to 300 MHz |
| UHF | Ultra High Frequency | 300 MHz to 3 GHz |
| SHF | Super High Frequency | 3 GHZ to 30 GHz |
| EHF | Extremely High Frequency | 30 GHz to 300 GHz |

3. In SDR Transmitter the information is formatted into Digital input, $m_i$, in the form of message symbols. Then these symbols could be converted into bit stream, $u_j$, by source encoding, encrypting(for secrecy), channel encoding and multiplexing(for multi signals) process. Depending upon the source coding and multiplexing, the data rate of streams can be controlled. After that desired Pulse coding could be undertaken using modulation such as PAM, QAM, PSK, FSK, OFDM or its variants to form Electromagnetic waves suitable to antenna characteristics. After it travels through media, the signal is received at the receiver.

4. In SDR the receiver, the received signal from Antenna is demodulated after carrier synchronization and automatic Gain control, using appropriate Demodulation techniques such as OFDM, QAM, FSK, PSK and its variants, to make the bit stream. From Bit Stream, after symbol synchronization and frame synchronization, the message is reconstructed successfully. The SDR Tx-Rx is given in the figue 1. Below.

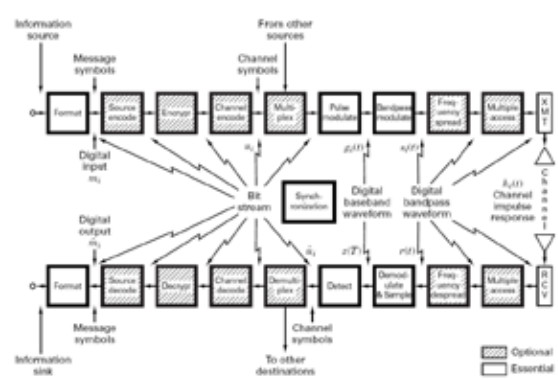



**Figure . 1. Block diagram of SDR Tx and Rx.**

5. Most of the above functions can be executed in a Digital Signal Processor(DSP) or Field Programmable Gate Array(FPGA) or General Purpose processor(GPP) using Algorithms, middleware, Rate Convertor software. The SDR Forum has characterized this SDR architecture[2] with the data flow mentioned in Figure. 2. Given below.

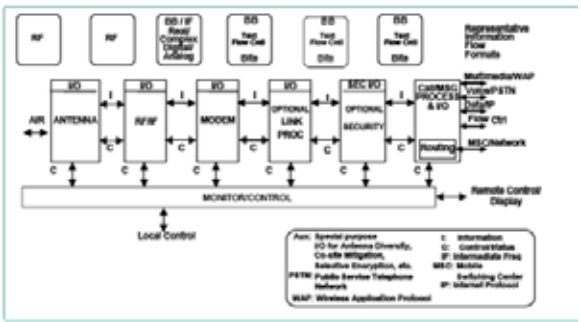



**Figure. 2. SDR Forum's SDR Architecture with Data Flow.**

6. Pure Software Radio is yet a distant dream due to high architectural Complexity and high Flexibility. Software Defined Radio(SDR) is the one which has medium architectural Complexity and more than medium Flexibility. These radios are reconfigurable one where in the software can be upgraded or added with other modes of radio or frequency of operation. A typical SDR development kit is shown below fig. 3 with all front panel interfaces. This has USB, Ethernet and HDMI Interfaces for interacting and viewing the performance, with RF Interfaces for connecting the antennae. The schematic diagram of such Universal Software Radio Peripheral(USRP)[3] is given in fig.4. The Software Development Kit module such as GNU Radio can be used to configure the functions given in Fig.5 into the software Objects shown in fig.6.




**Fig. 3. Photo of SDR Development set USRP E 100.**

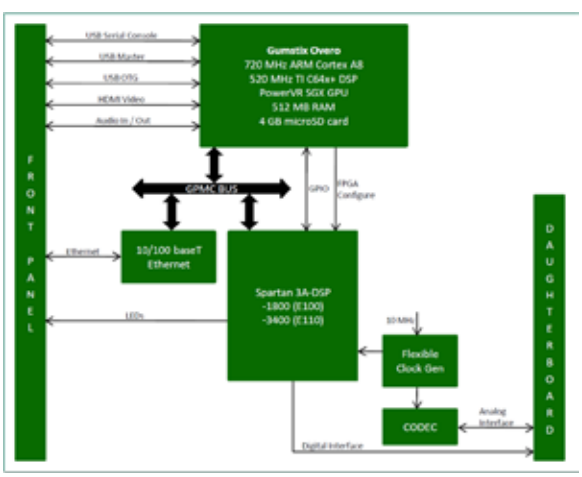



**Fig. 4. Schematic Diagram of USRP E 100.**

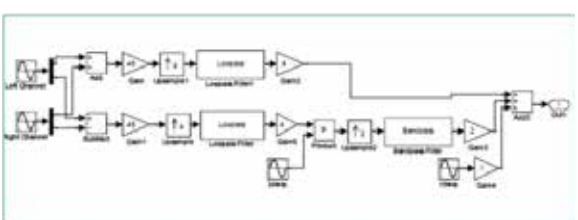


**Fig. 5. Stereo FM Mixing Schematic.**

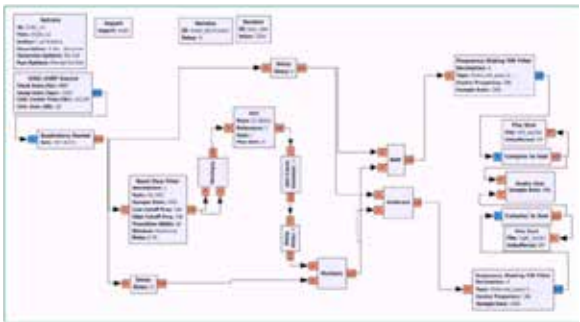


**Fig. 6. SDR Receiver Design in GNU Radio Companion**

7. The present Radios are Hard-wired and works only in **narrow-band** and for few **fixed modes** of Operation only. Thus any feature upgrade makes the Ownership **cost** very **high**. But, by implementing Radios in a Software programmable manner, the Total Cost of Ownership(TCO) comes down tremendously. The existing Hardware Radio **cannot be remotely repairable** and makes the **maintenance cost very high**. With the software Defined Radio adhering to SCA standards will have access anywhere with authentication and remote repair and upgrade is feasible with experts centrally located anywhere in the Globe. The software radio configuration for different modes can be developed and loaded into SDR in the form of Waveforms. Thus friendly countries can have common Waveforms and communicate efficiently with single Software Radio. This will be possible with the advent of advanced Digital Signal Processing(DSP) techniques at Software level running in either General Purpose Processor(GPP) or DSP Chip or Field Programmable Gate Array(FPGA). SDRs have the ability to work beyond simple single-channel, single-mode transreceiver technology, by accommodating arbitrary mode-changes because of the flexibility in determining the channel bandwidth, data rate, and modulation. These characteristics can be modified by direct input media or over-the-air download or through careful signal analysis called Cognitive Radio. A fully implemented SDR will have the capability to navigate a wide range of frequencies with programmable channel bandwidth and modulation characteristics. Thus SDR will qualify for multicarrier/multichannel, multimode and multirate radio. In practical it could be simultaneous operation of AM, FM, GMSK, CDMA or GSM and CDMA etc., If the design of high-linearity devices covering wide bandwidth is solved, modern SDR cost can reduce further.

### Software Communication Architecture

9. Over a period of decade, the SDR designer and developer have started agreeing to a unified architecture called Software Communication Architecture(SCA). This was initially developed at the instance of U.S. Department of Defense's Joint Program Executive Office(JPEO) requirements of Joint Tactical Radio System(JTRS). At present SCA version 4.0.1 was accepted as the last known SCA Specification standards[4] and released in the public domain on 18 Nov 2013 for development of SDR. The main aim of this specification is to provide framework for the deployment, management, intercommunication of software components in embedded, distributed-computing communication systems. As additional-goals, SCA maximizes software application portability, reusability, and scalability through the use of commercial protocols and products. The SCA is characterized by a particular combination of software components and connections. This JTRS SCA Standards or framework enables porting of "Waveforms", reuse of software, extensibility of HW and SW, interoperability between platforms and use of commercial product line. Since SDR is in essence a radio whose output is determined by software, the output is entirely reconfigurable at any given time. This ability to add, remove, or modify the output through reconfigurable and re-deployable software, leads to communication systems capable of multiple mode operations with variable format, data rates and bandwidths. SCA-based system will consist of an Operating Environment(OE) and one or more Applications. The OE provides the capability to manage and execute SCA components and it comprises of the POSIX based Operating System, CORBA/Interface Definition Language(IDL) for Transfer Mechanisms, JTRS Core Framework(CF) Control and XML based Domain Profiles for Platform Devices and Services.

10. The SCA Specification uses Unified Modeling Language(**UML**),

defined by the Object Management Group(OMG), to **graphically** represent interfaces, components, operation scenarios and sequence diagrams. For **textual** representation of the interfaces, the OMG has defined Interface Definition Language(**IDL**). For SCA Domain **Profile** elements, Extensible Markup Language(**XML**) are used, to identify the capabilities, properties, inter-dependencies, and location of hardware devices and software components.

11. The Core Framework(CF) is the main component of SCA, which provides core set of Interfaces defined in IDL. CF Interfaces consist of following set of Interfaces:-

(a) Base Application Interfaces are used by Application Layer and provides Port, Life Cycle, Testable Object, Port Supplier, Property Set, Resource Factory, and Resources.

(b) Framework Control Interfaces provide control of the system by interfacing Application, Application Factory, Domain Manager, Device, Loadable Device, Executable Device, Aggregate Device, and Domain Manager.

(c) Framework Service Interfaces provide interfaces for distributed file access services to software application components.

(d) Domain Profile describes the properties of hardware devices and software components in the system and enables application deployment.

**Management Aspects**

12. In SCA, software components are grouped into three functional groups viz., Radio Management, Devices and Applications. Radio management deals with installing / uninstalling /deploying / configuring applications, health monitoring, introspection and enforcing lifecycle of SCA components. SCA Devices can be started, stopped, configured, queried and tested by the control of Device Manager. An application is composed of Resources which can be deployed, started, stopped, configured, queried and tested.

13. Radio Management is performed by five components viz., Domain Manager, Application Factory, Application, Device Manager and File Manager. Domain manager is the central component for radio management. It accepts registration of every Device Manager, Device, and service. It is responsible for the installation of application and creates an Application Factory. It offers introspection services such as listen to events, get the list of installed and deployed applications. It provides two event Channels, viz., one for every device state changes and another one for radio management events. Domain Manager creates an Application Factory for each type of installed application. Application Factory is responsible for instantiating specific types of applications. It can create several instances of the same type of application. An Application component is used to start/stop the signal processing, change the behaviour of the application through configuration and release the application which terminates components and frees acquired capacity. A Radio node provides set of collocated Devices. A Device Manager is responsible for a single radio node in launch/shutdown by reading the node assembly descriptor and it registers to a Domain Manager.

14. Devices are software proxies for hardware devices and/or can be target for software deployment. There are three types of Devices viz., General Device, Loadable Device and Executable Device. General Device is used for getting access to autonomous hardware device such as modem, audio, GPS Receiver etc., Loadable device is generally used as a proxy for a device like an FPGA. Executable Device is generally used as a proxy for GPP.

15. An application can be viewed as a single Resource which has a certain combination of input, output, and control ports that need to be connected to other components. Its behaviour can be altered by changing the value of its configuration properties such as code rate etc., A Resource has capability requirements such as OS, Processor Devices and Capacity requirements such as MIPS, Kilo Bytes of Memory Devices etc., It executes simple and complex signal processing functions such as FFT. Resource is always a port supplier.

16. File Manager manages multiple distributed File Systems. File System enables remote access to physical files for creation, deletion and copying etc., of files. When one file resides over a span of multiple physical file systems, it is called federated file system. This type of file system is managed using the mount and un-mount operations.

17. The SDR which are compliant to SCA, are remotely accessible for configuration, loading of new waveforms and bug-fixing. Thus new products can be introduced by remote downloads. This saves the operating cost, development cost, capital cost in new product introduction. TCP/IP security and performance are key requirements of SDR Devices.

**Software Development Tools**

18. SDR can be developed using various combination of Universal Software Radio Peripheral(USRP) and Radio Software Toolkit. Few open source and commercial Toolkit are listed below:-

19. GNU Radio[5] is an open source toolkit of signal processing blocks which are written in C++ for implementing SDRs on external /RF hardware. GNU Radio Companion(GRC) provides GUI that allows flow graphs which are built in Python and signal visualisers to be built out of signal processing blocks. USRP could be from National Instruments to cover frequency range from HF to UHF ranges.

20. Prismtech[6] has introduced a product Spectra CX bundle and alongwith Mathworks' Simulink for the development of SCA compliant SDR. Spectra CX is useful in designing and simulating waveforms, transforming the algorithmic blocks into deployable components, deploy and test Waveforms and applications.

21. National Instruments[7] Labview software and USRP will be another toolkit for development of SDR.

22. SCARI software suite from the Communication Research Centre(CRC) Canada is a high performance SDR waveform development software and debugging toolkit.

23. Pentek[8] company produces various SDR modules for further deployment in various frequency bands. Tiny SDR, Softrock(Flex, HDSDR), SDR Cube, Genesis(G11, GSDR), HPSDR, Zeligsoft Component Enabler(CE) and code Generator, Green Hills Software's INTEGRITY RTOS, Harris dmTK Core Framework and Development Tools, ORBexpress(CORBA middleware)etc., are other available SDRs and SCA Operating Environment and Development Tools.

24. Also some Simulation Package such as SciLab, GNU Radio or MatLab can be used for verifying the SDR. The performance can be verified with parameters such as Signal to Noise Ratio (SNR), Bit Error Rate (BER), Mean opinion Score (MoS), Spectrum Visualisation and Log Spectral Distance (LSD).

**Conclusion**

25. In this paper the Software Defined Radio(SDR) Technology was discussed with the block diagram of SDR system. Few examples were given to illustrate the remote management of SDR and development of Radio blocks through SDR. Then Software Communication Architecture(SCA) was analysed with the standards given in JTRS. In specific core framework was explained with various interfaces. Various management aspects were analysed on a SCA compliant Radio for implementation to meet interoperability and remote configuration. With the standard Ethernet protocols, the SDR technology will be very useful in easy deployment. SDR Development Tool-kit with Hardware and Software were studied and listed.

## REFERENCES

1. SDR Forum Document Number SDRF-06-R-0011-V1_0_0.pdf accessed from http://ww.sdrforum.org/pages/document/Library/documents/ SDRF-06-R-0011-V1_0_0.pdf accessed on 03 Oct 2013. | 2. "Base Station System Structure", SDR Forum Document Number SDRF-01-P-0006-V2.0.0 accessed from http://ww.sdrforum.org/pages/document/Library/documents/ SDRF-01-P-0006-V2_0_0.pdf accessed on 03 Oct 2013. | 3. "Embedded Software Defined Radio Design with the E100 Universal Software Radio Peripheral" by Lee Anthony Kashka", MS Thesis, University of Colorado, 01 Nov 2011. | 4. "Software Communication Architecture Specification version 4.0 dated 28 Feb 2012 and its Appendices "A" to "F" version 4.0.1 dated 01 Oct 2012" by JPEO of JTRS. | 5. "GNU Radio Tutorial" by Tom Rondeau dated 27 May 2012, www/trondeau.com/gnu-radio-conference-2011. | 6. "A Model Driven Tool Integration Process for Rapid SDR Development" Webcast dated 12 May 2013, from www.prismtech.com/spectra accessed on 01 Jan 2014. | 7. "Designing Hands-On Wireless Communications Labs with the NI Universal Software Radio Peripheral and LabVIEW Solutions" accessed on 25 Jan 2014, from http://sine.ni.com/cs/app/doc/p/id/cs-13848.pdf. | 8. "Putting FPGAs to work in Software Radio Systems" sixth edition eBook by Rodger H.Hosking, Year 2012. |