



## Software Defined Radio (SDR) and Low Field Magnetic Resonance Imaging (MRI) -A Review

Abhishek S J<sup>1</sup>, Anirudh T S<sup>2</sup>, Murigendrayya Hiremath<sup>3</sup>

<sup>3</sup>Assistant Professor

<sup>1,2,3</sup>Department of Telecommunication Engineering,  
Dayananda Sagar College of Engineering, Bangalore, Karnataka, India

### ABSTRACT

The need of low-field MRI systems is mainly for biomedical applications and also for food quality measurements. For example, it is used to measure the quantity of fat and moisture in samples of minced beef and meat. Benefits of using low field MRI system are higher accuracy in detection, reduced system cost, size and its complexity. The key element of MRI system is the digital receiver. The architecture of receiver is inspired from advance digital signal processing techniques that are exponentially applied in the mobile radio and cellular based systems. In the area of Telecommunications the set of these techniques is commonly called “Software Defined Radio” (SDR) or “Digital Receiving”. Its basic idea is to perform digital to analog conversion as close to the receiving coil as possible. All the signal processing can be then realized in the digital domain. This has the advantage of noise and distortion reduction associated with analog mixing stage and better out-of-band noise rejection by the decimation and the digital filters. Moreover, the parameters of the system are fully programmable. They could be modified by the software and without changing the hardware architecture.

**Keywords:** SDR, MRI, low-field

### 1. LITERATURE REVIEW

Aktham Asfour, Kosai Raof, Jean-Paul Yonnet, have presented the concepts of Software Defined Radio and Magnetic Resonance Imaging systems in their paper “Software Defined Radio (SDR) and Direct Digital

Synthesizer (DDS) for NMR/MRI Instruments at Low-Field”, sensors, 2013 [1]. This paper mainly focuses on the fully digital radio frequency components for the design of MRI systems using low-field. “Basics of Software Defined Radio and Application” by National Programme on Technology and Enhanced Learning (NPTEL) under Dr. Meenakshi Rawat, describes various components of SDR’s with the understanding of their limitation and application of Software Defined solutions in order to overcome such limitations [3]. Dr. Meenakshi Rawat is currently an Assistant professor, Department of Electronics and communication engineering, IIT Roorkee. She is currently handling four projects with funding from defense, private telecommunication industries and one international funding related to SDR for 5G communication [4]. Programmable Platform for Experiments and Academic Research on SDR (PEARS) is an SDR development platform which is a powerful combination of configurable hardware and open source software [5]. GNU is a free and open-source software development framework that enables users to design, simulate and deploy Software Defined Radios and other general processing systems. Devarpita Sinha, Anish Kumar Verma, Sanjay Kumar, have presented the concepts of Software Defined Radio in their paper “Software Defined Radio: Operation, Challenges and Possible Solutions” [6]. The vision of SDR is implementing a single radio device containing software controlled digital hardware that can emulate any radio signal of evolving or already existing wireless standards simply

by updating software without replacing the underlying hardware platform.

Héctor Miyashiro, Melissa Medrano†, José Huaracaya†, Jinmi Lezama, presented the concepts of different SDR kits present in the market and their features in their paper” Software Defined Radio for hands-on Communication theory” [7]. This paper gives the knowledge of using an SDR, Classification, and its method of selecting the software and hardware components based on their specifications and our needs.

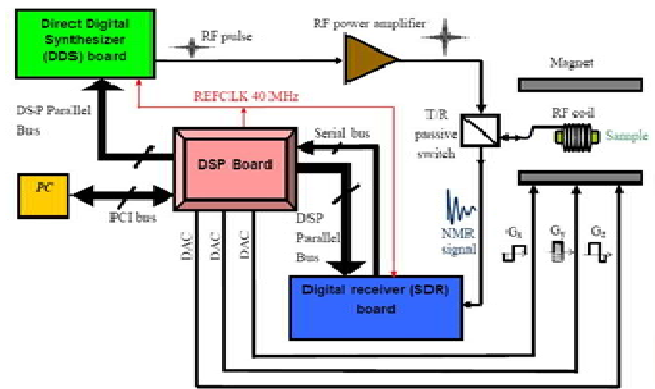
## 2. SOFTWARE DEFEND RADIO FOR MRI SYSTEMS AT LOW-FIELD

### 2.1 Introduction

This paper mainly focuses on the fully digital radio frequency components for the design of Magnetic Resonance Imaging (MRI) systems using low-field. These digital components is mainly dependent on the three main elements- Direct Digital Synthesizer (DDS) for generating a pulse, Software Defined Radio (SDR) for receiving digital MRI signals, Digital Signal Processor for controlling the system. Medical diagnosis and Medical research are the few examples of areas that have been benefited using the MRI signals. Low-field MRI signals are the most common type of MRI signals used due to its low system cost, size and complexity. This is because the low field MRI signals can be generated easily using electromagnets or permanent magnets. Few examples of low-field and mobile MRI systems may include in areas such as material science, quality control of food products, moisture measurement, wood, oil and paper industries, studies on cultural heritage like stones, monuments etc. The low-field MRI spectrometers are not very well available in the market as of now, although the need for these spectrometers is increasing day by day. This paper will mainly focus on a new design of the electronics for transmitting and signal receiving.

### 2.2 Hardware Design

Figure 1 shows the block diagram of DDS and SDR based MRI system at 0.1T. The main element of this block diagram is the Digital signal processor which controls the other two elements- DDS and SDR.



**Figure1.** A block diagram of the DDS and SDR based MRI system at 0.1 T.

The RF pulse is generated by the DDS and amplified using a RF power amplifier after which this signal is sent to a well-tuned RF coil via a transmit/receive (T/R) passive switch. At the end of the RF pulse, the MRI signal is detected by the coil and transmitted to the digital receiver through the same T/R switch. This receiver consists of an amplifier and a Software Defined Radio (SDR) which uses digital quadrature demodulation, filtering and decimation. The samples of the demodulated signals are then transmitted to the DSP, via a serial bus, where more signal processing could be performed if necessary. The DSP, DDS and the SDR are synchronized with a same external 40 MHz reference clock.

### 2.3 Sequencer and Software

The software was developed using the Lab Windows environment and DSP assembly language. Figure 2 illustrates its block diagram. The Graphical User Interface (GUI) allows user to define and edit all the configurations and parameters (frequencies, phases, filters...) of the hardware (DDS and SDR). It also enables user to draw and/or graphically edit a sequence (pulse envelop, gradient waveforms) and to define parameters such as repetition time (TR), echo time (TE) and the Field-of-View, etc. The use of this graphical sequence editor allows avoiding programming the gradient waveforms in text sequence editor. Data from the GUI are stored in ASCII files. These files are then linked with the assembly and the architecture files of the DSP. The assembly compiler provides then an executable file which is loaded in the DSP RAM via the PCI bus. This assembly program or sequence program interprets all the data defined in the GUI and controls all the temporal events of the sequence and the data transfer and commands between the three boards (DSP, DDS and SDR). The use of the assembly language was adequate to optimize the

memory use of the DSP and to facilitate the management of the temporal events within the sequence. In this way, the DSP ensures the role of a high performance pulse programmer (sequencer) which is able to execute each program instruction at every clock cycle with a high temporal resolution (25 ns for a clock of 40 MHz). The generation of gradient signal is achieved with 0.5 μs of temporal resolution.

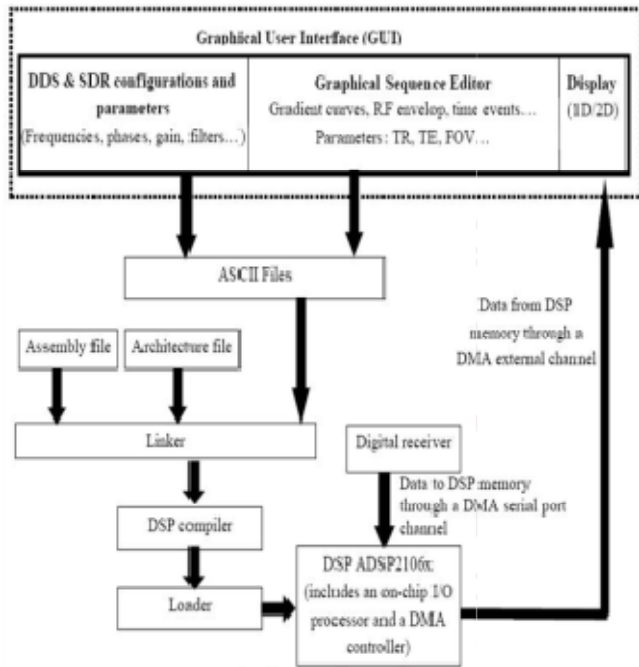


Figure 2. Block diagram of the developed software.

2.4 Discussions and Validation

The inductance  $L_p$  of the coil and the tuning capacitor  $C$  were about 0.7 μH and 2,080 pF, respectively. No adjustable variable capacitor section was used for fine tuning the coil at this validation stage. Actually, the power supply of the used magnet allows to slightly change the value the magnetic field around 0.1 T. The working frequency of the DDS and the SDR, which is fully programmable, is then easily adjusted to fit the resonance frequency of the coil (when this is inside the magnet and loaded by the sample). In other words, we perform a tuning of the working frequency rather than a tuning of the resonance frequency of the coil. On another hand, this solution avoids the use of somewhat sized variable capacitors.

3. SDR FOR HANDS-ON COMMUNICATION THEOR

3.1 Hardware Design

Software Defined Radio is a platform which is mainly made up of RF front end, ADC/DAC converters and Digital Signal Components as shown in the Figure

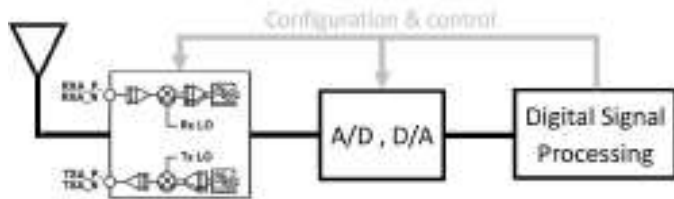
3, The antenna demodulates the signal which is caught from the transmitting antenna into a Intermediate Frequency(IF) or Baseband(BB),which is then sampled using ADC and processed using a DSP processor .The Transmitting operation is exactly opposite of that of receiving. All the processes will be configured using software which can hold control over the Digital elements eg Numerically Controlled Oscillator(NCO), The Programmable Gain Amplifier(PGA).There are different types of SDR which functions differently based on the application or a activity. For educational purpose, the main factors are cost and user accessibility. Table 1 shows and compares the SDR's in the market nowadays. Based on a scenario where a instructor transmit and a student receive we can use HackRF\_one for Tx and RTL-SDR for Rx.

Table 1: SDRs Hardware Comparison

	LIME SDR	USRP B210	BLADE RF	HackRF_one	RTL-SDR
Frequency (MHz)	0.1 - 3800	70 - 6000	300 - 3800	1 - 6000	0.5 - 1750
TX/RX	6 RX 4 TX	2 RX 2 TX	RX TX	RX TX	RX
Resolution (Bits)	12	12	12	8	8
Interface USB	3.0	3.0	3.0	2.0	2.0
Sample Rate(Msp/s)	20	61.44	40	20	2.4
Bandwidth (Mhz)	64	56	28	20	3.2
Open Source	YES	YES	YES	YES	YES
GNU Radio	YES	YES	YES	YES	YES
Price(\$)	299	1200	300-450	240-350	20

**Table 2** :Software Comparison

	GNU Radio Companion	MATLAB Simulink	LabView
Open Source	YES	YES	NO
Price(\$)	0	71	450
size(mb)	324	5000	900



**Figure 3**: SDR Block Diagram

3.2 Software Design

In order to run software, In a SDR, we have a DSP component as a Microprocessor. The programs are implemented using this Software to process signals at IF and BB: where it is configured to control the SDR hardware components. For educational purpose we select the Software which not only has good cost and user accessibility but also the flexibility to develop different applications and its user friendliness. Table 2 provides the Principal Software available in market

3.3 Methodology

In order to implement communication theory into practical's, we have two methods:

**Education Model:** There are few methods of using physical electronic blocks with multi-functional characteristics (eg

Leybold, Promax).They gives very good basic of presenting of concepts but is limited to hardware processing. They are also pretty expensive, cannot be used individually, which is not economic.

**Simulation Model:** The software provides varieties of functions and tools for generating and processing simulated signals. But for usage of this software we should buy license for each computer.

The Methodology involved in Laboratory Sessions include Theoretical and also Practical workshops: The Theoretical part is supported with Simulations which helps in improving the effectiveness of the theory. The study of analog and digital modulation topics, theory summarizes the basic concepts with emphasis

on algorithms and equations that help to identify basic operations and functions such as adders signal sources, FIR filters, etc. needed to design the flow graph in GRC. Designing of discrete blocks than an Integrated which helps us to analyse at each step. On practical part, before implementation, the design is checked in virtual scenario through simulation, which are configure in SDR hardware. The hardware is realized with minor changes in design.

Following concepts are taken into consideration for their importance in our content:

**Basic concepts:** Going through the basics of Digital Signal Processing, Nyquist theorem, Digital

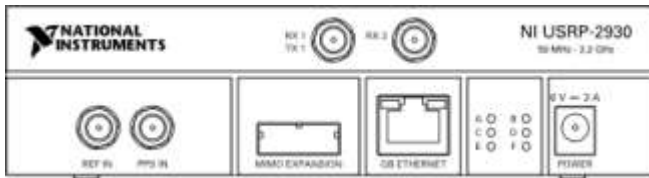
1. filtering, communication, Quadrature modulator and embedded systems.
2. **Introduction to SDR:** Getting familiar with SDR with, Definition, Architecture, Available Hardware and current Trends.
3. **Introduction to GRC:** Every SDR hardware needs a Software platform to implement and it is presented in different software and it's installation is Studied.
4. **Analog Modulation:** Because of its simplicity in designing, simulating and implementing with GRC's and SDR's, this provides a good starting point, hence AM(Amplitude-Modulation),FM(Frequency-Modulation)are often discussed.
5. **Digital Modulation:** In this Information era, Communication starts with timing, frequency and synchronization, tasks needed to avoid imperfections due to Hardware; it can be continued to different binary, M-ary and OFDM modulation that can be analysed through flowcharts and waveforms.

3.4 Summary

The Steps to be followed in order to perform an operation using a SDR, Classification of SDR's and method of selecting the Software platforms and Hardware components based on their specifications and user requirements for a certain goals of applications.

**4. USRP 2930**

4.1 Front Panel



4.2 NI USRP-2930 module LED's

CONNECTOR	USE
RX1 TX1	Input and output terminal for the RF signal. RX1 TX1 is an SMA (f) connector with an impedance of 50 Ω and is a single-ended input or output channel.
RX2	Input terminal for the RF signal. RX2 is an SMA (f) connector with an impedance of 50 Ω and is a single-ended input channel.
REF IN	This terminal is not used for this device
PPS IN	PPS IN is an SMA (f) connector with an impedance of 50 Ω and is a single-ended input channel. PPS IN accepts 0 V to 3.3 V TTL and 0 V to 5 V TTL signals.
MIMO EXPANSION	The MIMO EXPANSION interface port connects two USRP devices using compatible MIMO cable.
GB ETHERNET	The gigabit Ethernet port accepts an RJ-45 connector and gigabit Ethernet compatible cable.
POWER	The power input accepts a 6 V, 3 A external DC power connector.

LED	INDICATION
A	Indicates the transmit status of the module: OFF—The module is not transmitting data. GREEN—The module is transmitting data.
B	Indicates the status of the physical MIMO cable link: OFF—The modules are not connected using the MIMO cable. GREEN—The modules are connected using the MIMO cable
C	Indicates the receive status of the module: OFF—The module is not receiving data. GREEN—The module is receiving data.
D	Indicates the firmware status of the module: OFF—The firmware is not loaded. GREEN—The firmware is loaded.
E	Indicates the reference lock status of the LO on the module: OFF—There is no reference signal, or the LO is not locked to a reference signal. BLINKING—The LO is not locked to a reference signal. GREEN—The LO is locked to a reference signal
F	Indicates the power status of the module: OFF—The module is powered off. GREEN—The module is powered on.

4.3 Back Panel



GPS ANT is the input terminal for the GPS antenna signal. GPS ANT is an SMA (f) connector with an impedance of 50 Ω.

## 5. CDAC DATASHEETS AND PEARS TUTORIAL

### 5.1 Programmable Platform for Experiments & academic research on SDR (PEARS)

1. The SDR development platform is used because of its combination of both Hardware and Open-source characterized software.
2. This software platform enables us to use through wide range varieties of wireless technologies such as AM, FM, QAM.
3. It has an RF front end for radio with interfaces to PC/embedded computer for baseband processing and it also works as standalone radio too.
4. It provides GNU Radio, MATLAB and Simulink interfaces to operate.

### 5.2 GNU Radio: Basic Introduction

It is a free and open -source software development framework that helps users to design, simulate and deploy Software defined radios and other signal processing systems. The part of "GNU project" is licensed under and copyrighted under Free software foundation. and supported globally by the open-source community and widely used with hobbyist, academicians and some commercial events. It wraps up the functionality in easy-to-use reusable code blocks and provides an extensive library for standard algorithms.

### 5.3 GNU Radio: Implementation

GNU Radio provides a library of signal processing blocks which can be connected together in the forms of graph and stream of data's. The software structure consists of two levels,

1. Signal processing blocks using c++,
2. Python, the higher language, used to create a network a graph and glue those blocks together.

SWIG (Software wrapper interface Generator) provides interfaces between C++ and python, so that these function scan be called directly in Python.

Flow graphs that depict the connection between the blocks in GNU radio are form of python files.

GNU Radio Companion (GRC) is a graphical tool used to create a python flow graph files graphically as opposed to creating them in code alone. Flow graph when compiled in 'GRC', python code generates

desired GUI widgets, and creates and connects the blocks in the flow graph.

### 5.4 Developing Waveforms Using "PEARS Module" in GNU Radio,

In order to provide application programming interface to PEARS, develop a GNU Radio Module named "SDRP" with a Source code: `.../gnuradio-master/gr-sdrp`. SDRP Module makes use of "LIBIIO", a library written in C++, that abstracts the low-level details of the hardware to interface with IIO Framework.

Libiio composed of following backends:

- "local" backend, which interfaces the Linux kernel through the virtual filesystem,
- "Network" backend, which interfaces the "iiod" server through a network link. The SDRP blocks are just a tiny wrapper around Libiio.

The SDRP Module includes

- SDRP Sink-used to configure the transmission parameters of RF Front end.
- SDRP Source-used to configure the receive parameters of RF Front end.

Above blocks together with several pre-assembled signal processing blocks in GNU Radio facilitates implementation of RF waveforms in PEARS.

### 5.5 PEARS SINK

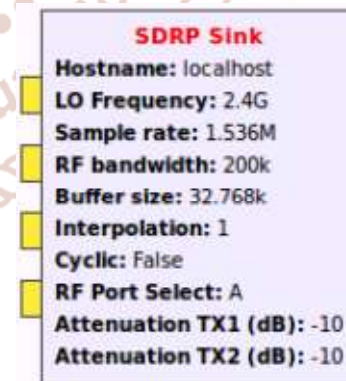


Figure 4: SDRP Sink

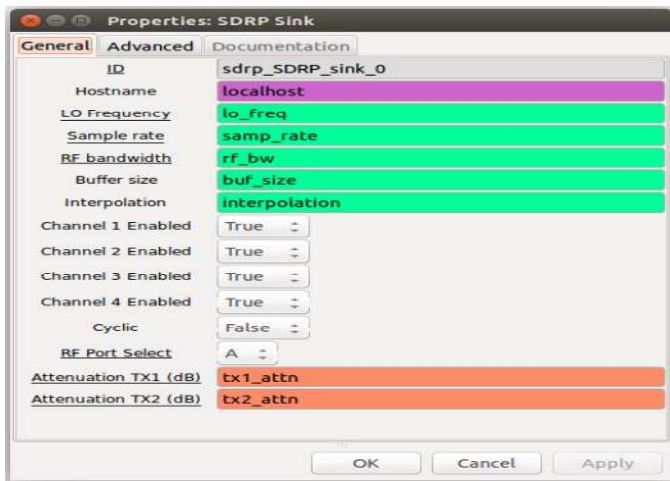


Figure 5: Properties of SDRP Sink

**Host name:** Set to “localhost” if using GNU Radio on the PEARS board. If using GNU Radio on a PC, set this field to IP of the PEARS Board.

**LO Frequency:** Local Oscillator frequency -70 MHz to 6 GHz

**Sample rate:** Frequency at which the hardware will input/output samples. Max 122.88Mpsps.

**RF Bandwidth:** RF TX Bandwidth 200 kHz to 56 MHz

**Buffer size:** Size of the buffer that holds the samples to be written to the hardware.

**Interpolation:** Data input rate of the block = (Sample rate/interpolation).

**Channel enable (1-4):** Enable the TX channels, Channel 1 –TX1-I Channel 2 – TX1-Q Channel 3 – TX2-I Channel 4 – TX2-Q.

**Cyclic:** If true, first buffer of samples will be repeated on the enabled channels until the program is stopped.

**RF\_port select:** Selecting the TX port A or B.

**Attenuation TX (db):** Attenuation in TX channels, -0 to -89.75dB with a resolution of 0.25dB.

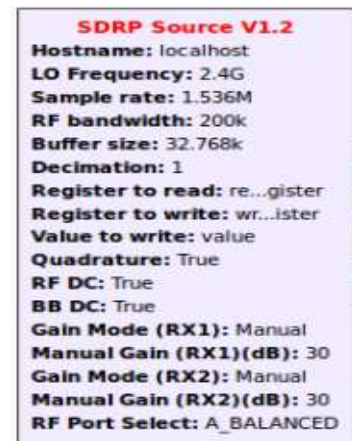


Figure 6: SDRP source

## PROPERTIES:

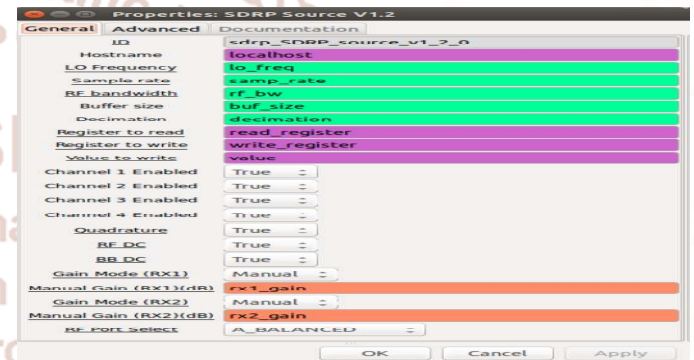


Figure 7: Properties of SDRP Source

**ID:** Unique ID of the PEARS Source GRC block.

**Hostname:** PEARS Sink GRC block set to “localhost” if using GNU Radio on the ARM processor of PEARS. If using GNU Radio on a PC, set this field to IP of the PEARS.

**LO frequency:** Local Oscillator frequency- 70 MHz to 6 GHz.

**Sample rate:** Frequency at which the hardware will input/output samples. Max 122.88Mpsps.

**RF Bandwidth:** RF TX Bandwidth 200 kHz to 56 MHz.

**Buffer size:** Size of the buffer that holds the samples to be written to the hardware.

**Decimation:** Data output rate of the block = (Sample rate/decimation).

**Register\_read:** Address of the AD9361 register in Hex format which is to be read.

**Register to write:** Value to be written to the AD9361 register.

**Channel enable (1-4):** Enable the RX channels, Channel 1 –RX1-I Channel 2 – RX1-Q Channel 3 – RX2-I Channel 4 – RX2-Q

**Quadrature:** If True enables the Quadrature tracking.

**RF DC:** If True enables the RF DC Offset tracking.

**BB DC:** If True enables the Baseband DC Offset tracking.

**Gain mode (RX1):** Select one of the following Rx Gain modes: Manual, Slow attack, Fast attack, Hybrid.

**Manual Gain (RX1) (DB):** Gain in Rx channels, Max gain 73 dB with a resolution of 1dB.

**Gain mode (RX2):** Select one of the following Rx Gain modes: Manual, Slow attack, Fast attack, Hybrid.

**Manual Gain (RX2) (DB):** Gain in Rx channels, Max gain 73 dB with a resolution of 1dB.

**RF Port Select:** Select one of the following RF Ports: A\_BALANCED, B\_BALANCED, C\_BALANCED, A\_N, B\_N, C\_N, A\_P, BA\_P, C\_P.

## References

- 1) Aktham Asfour, Kosai Raouf, Jean-Paul Yonnet, “Software Defined Radio (SDR) and Direct Digital Synthesizer (DDS) for NMR/MRI Instruments at Low-Field”, sensors, 2013.
- 2) Web link: <http://www.mdpi.com/journal/sensors>.
- 3) Web link: [https://onlinecourses.nptel.ac.in/noc18\\_ec01/preview](https://onlinecourses.nptel.ac.in/noc18_ec01/preview).
- 4) Dr. Meenakshi Rawat, Assistant Professor, Department of Electronics and communication engineering, Indian Institute of Technology (IIT), Roorkee, Uttarakhand, India.
- 5) CDAC data sheet ([www.cdac.in](http://www.cdac.in)).
- 6) Devarpita Sinha, Anish Kumar Verma, Sanjay Kumar, “Software Defined Radio: Operation, Challenges and Possible Solutions”, Department of Electronics & Communication Engineering BIT Mesra, Ranchi, India.

7) [7] Héctor Miyashiro, Melissa Medrano†, José Huarcaya†, Jinmi

8) Lezama,” Software Defined Radio for hands-on Communication theory”, Instituto Nacional de Investigación y Capacitación de Telecomunicaciones - INICTEL-UNI, Universidad Nacional de Ingeniería – UNI, 2017.