# Real-Time Data Transfer Based on Software Defined Radio Technique using Gnu radio/USRP

**Ponnaluru Sowjanya, Penke Satyanarayana**

***Abstract:*** ***Software Defined Radio (SDR) offers a extensive radio communication platform that uses software updates to make use of fresh technology. From SDR, the idea of an Orthogonal frequency division multiplexing (OFDM) has evolved to personalize SDRs. The channel dispersiveness causes Inter Symbol Interference (ISI) but OFDM is more resistant at these condition because of this reason it is widely used in wireless communication systems. OFDM is having a good performance in terms of Bit Error Rate (BER) and high spectrum efficiency, so it is considered as a key role for next generation wireless communication system. In this paper, three different types of data are transferred in a real time SDR of OFDM transceiver using GNURadio/Universal Software Radio Peripheral (USRP). OFDM is extremely sensitive for synchronization errors such as time and frequency offsets and to estimate channel condition. Therefore, a standard algorithm is applied to solve synchronization and channel estimation problems in SDR based OFDM system. This testbed is implemented using two USRPs of model N210 as transmitter and receiver with an open source of GNURadio as a software. The implementation of OFDM is evaluated for different types of information like text, audio and Image. This evaluates the BER v/s SNR for real time data transmission in SDR Environment.***

***Keywords:*** ***SDR, OFDM, GNURadio, USRP N210, Channel Estimation, Synchronization.***

## I. INTRODUCTION

Wireless communication technology has recently seen spectacular advances in devices, protocols, and applications. Future wireless technologies will require low energy consumption and high data rates for data-heavy applications [1]. As the number of user's increases on commonly accessed mobile bandwidths, overcrowding will become a more serious challenge, and more flexibility in transmission standards will be needed to process the contention inherent in multiple accesses. Wireless communication standards are frequently developed to meet these challenges. Long-Term Evolution (LTE) is the latest standard for mobile phone technology, and many research projects are focused on prototyping and testing wireless protocols. Orthogonal Frequency Division Multiplexing (OFDM) is a commonly used signal modulation scheme by which a stream of symbols is divided into many slowly modulated narrowband subcarriers that are very close together to attain high data transmission rates. This arrangement makes the signal less sensitive to frequency-selective fading.

An OFDM wireless communication system forms several parallel nonselective frequency channels, thereby reducing inter-symbol interference (ISI). The reliability of OFDM systems depends on the decoder's method for obtaining channel state information (CSI) of the dispersive fading channels and compensating for channel effects like less delay spread and less Doppler spread. Several authors have tested real-time implementations of OFDM using various channel-estimation schemes [2–5]. The physical layer-capture effect, carrier sensing, and preamble detection have been studied experimentally, focusing on signal capture and the Least-Square (LS) scheme for channel estimation [2]. Another group proposed and tested a channel-estimation scheme with pilot subcarriers and achieved low complexity, buffer-free data flow, high accuracy, small pilot bandwidth usage, and excellent stability [3–4]. These tests were implemented in real-time optical OFDM transceivers based on Field-Programmable Gate Arrays (FPGAs), which allow a three-level variable-power loading scheme. For high-speed signal processing GPUS are used as modern processor, for this purpose a high-performance software radio platform (CUDA-based software radio (CuSora)) was developed [5]. In this paper we developed a testbed which represents an SDR based OFDM for real time testing. With this setup we can analyze and compare the specifications of the system in real environment. GNURadio is an open-source signal processing framework. It allows the implementation of a General-Purpose Processor on a normal PC. It is one of the best tools which supports the hardware connectivity in SDR platform. It covers a wide range of projects to build a variable software radio system using many libraries presented in it. In [6] using GNURadio/FPGA reconfigurable hardware, author presented a prototype in 3-30MHz frequency band of an RF signal receiver in SDR platform. In [7] author proposed a wide-band spectrum adaption at frame-level which can be easily implemented into devices in different protocols. And they demonstrated their model feasibility using GNURadio/SDR platform. In [8] a compatible resolution that irrespective to the channel condition the energy-constrained devices will scale down their sample rates, which is named as Samples Wi-Fi.

Revised Manuscript Received on October 30, 2019.
* Correspondence Author

**Ponnaluru Sowjanya*,** Electronics and Communication Engineering, Koneru Lakshmaiah Education Foundation, Vaddeswaram, AP, India.

**Penke Satyanarayana,** Electronics and Communication Engineering, Koneru Lakshmaiah Education Foundation, Vaddeswaram, AP, India.

To evaluate this, they used GNURadio/USRP for their SDR platform. In [9] and [10] in different methods they proposed an anti-jamming mechanism and they tested their own proposed method using GNURadio. In [11] to enable Spread Spectrum communication with out pre-shared secrets author introduced a mechanism named as TREKS. And they evaluated this mechanism using a real-time testbed containing GNURadio and USRP. In [12] they developed an SDR based test bed to study the OFDM performance in 802.11 protocol. For this purpose, they have chosen the GNURsdio/USRP as their perfect platform. In [13] they tested two different solutions for integrating the Graphics Processing Units in a Software Defined Radio environment. And concluded that only one of the methods which they proposed is compatible to the environment. In [14] a new PHY/MAC protocol is designed which is named as Diversity-aware Wi-Fi and tested and compared this new protocol with the existing solutions using USRP/GNURadio platform. By considering these diverse applications, we can believe that GNURadio is one of the best SDR based software.OFDM is an advanced modulation scheme, but due to the worst channel conditions the system suffers poor performance in terms of BER. The channel conditions play a major role in synchronization and channel estimation. In OFDM by sending a known data with the transmitting packets we can assume the channel conditions and compensate them to original data. The whole process was analyzed in this paper and tested them for different data types. For this purpose, we created a real-time setup as shown in Fig. 1. The rest of the paper is presented as follows. Section 2 details the channel-estimation and synchronization algorithms that were used. Section 3 explains how we implemented this network in a real-time environment using GNURadio and USRP. Section 4 presents the experimental results of the testbed, and Section 5 presents our conclusions.

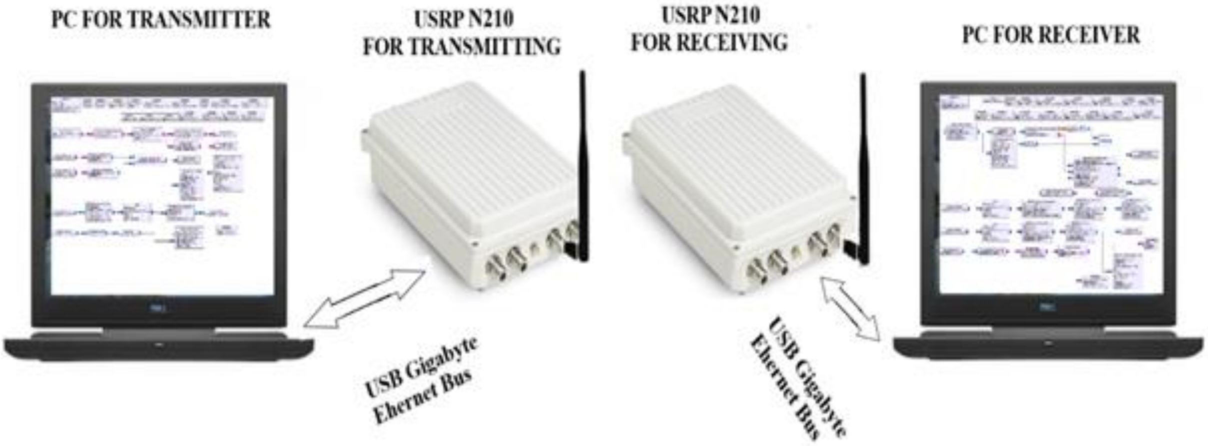

**Fig. 1. Experimental Setup**

## II. BACKGROUND

This OFDM system employs a total of 52 subcarriers, including 48 information subcoarriers and 4 pilots, which allow a coherent detection. The distance between 64 sub-carriers is 312,50 kHz for a complete signal bandwidth of 20 MHz. [15]. Synchronization errors, carrier frequency offset, and symbol timing offset are the main problems at the receiver end. These errors generate ICI and ISI, which affect the orthogonality of the subcarriers. The receiver expects pilot symbols to prevent this problem. The receiver uses autocorrelation and cross-correlation functions to recover data from the transmitted symbol sequence. Some portions of the bandwidth and power are reserved for carrying the known pilot sequence. IEEE 802.11a includes a structured preamble that is used for synchronization. This preamble includes 12 OFDM symbols and is divided into a short training sequence of 10 short symbols and a long training sequence of two long symbols. 12 subcarriers are used for the short OFDM training symbols and are modulated by the elements of the sequence $S–26, 26 = \sqrt{(13/6)} \times \{0, 0, 1+j, 0, 0, 0, −1−j, 0, 0, 0, 1+j, 0, 0, 0, −1−j, 0, 0, 0, −1−j, 0, 0, 0, 1+j, 0, 0, 0, 0, 0, 0, 0, −1−j, 0, 0, 0, −1−j, 0, 0, 0, 1+j, 0, 0, 0, 1+j, 0, 0, 0, 1+j, 0, 0, 0, 1+j, 0,0$. In the short preamble (SP), symbols 1 to 7 are used for Automatic Gain Control, signal detection and Diversity Gain, and symbols 8 to 10 are used for timing synchronization and coarse estimation of the frequency offset.

Then, 53 subcarriers (including a zero value at DC) are used in the long OFDM training sequence, which is modulated by the elements of the sequence $L–26, 26 = \{1, 1, −1, −1, 1, 1, −1, 1, −1, 1, 1, 1, 1, 1, 1, −1, −1, 1, 1, −1, 1, −1, 1, 1, 1, 1, 0, 1, −1, −1, 1, 1, −1, 1, −1, 1, −1, −1, −1, −1, −1, 1, 1, −1, −1, 1, −1, 1, −1, 1, 1, 1, 1\}$. The Long Preamble (LP) is used for fine frequency offset estimation, channel estimation, and channel equalization.

### A. Channel Estimation

Estimating the channel is the characterization of transmission channels according to a predefined mathematical model. Short term or instantaneous CSI and long term or statistical CSI are the two factors that characterize the mathematical channel model. Statistical data like statistical spreading and average channel gain was included in long term CSI. Channel impulse response is the only factor included in short term CSI. In OFDM systems, the channel impulse response that is identified by time-domain channel estimation and channel frequency response that is identified by frequency-domain channel estimation was done before and after DFT processing, respectively [16]. Channel estimation can take three approaches: pilot-aided, blind and decision-directed channel estimation (DDCE).

One of the most standard channel-estimation methods is pilot-aided channel estimated method. In this method, a sender transmits a well-known pilot data, which is used as a testimonial by both sender and receiver. Pilot symbols are slightly complex in computation but used in any wireless communication system. Though, this method degrades the data rate because some symbols are used for pilots instead of data, and the channel bandwidth is limited. Accurately estimating the channel, even when the number of pilots reduced is a challenge in pilot-aided channel-estimation method. Pilot assignment methods use blocks or combs, as shown in Fig. 2. For slow fading channels in which the channel moves slowly, a block-type pilot allocation model is appropriate. Conversely, for fast-fading channel, a comb-type pilot allocation is appropriate because the pilots are uniformly distributed throughout the symbol sequence. Frequency-domain interpolation is necessary to determine the data symbols channel response and make this method more susceptible to frequency-selective channels.

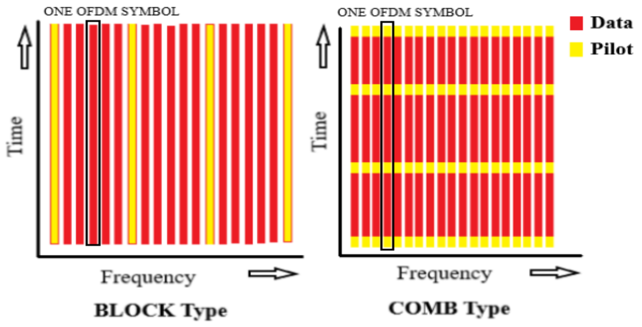

**Fig. 2 Types of pilot assignments**

The block pilot composition is depicted in Fig. 2. In this arrangement, each OFDM symbol has a pilot symbol at the beginning of each subcarrier. Time-domain interpolation is applied to use these pilots in estimating the channel. Since the inverse of the Doppler frequency $f_{dp}$ in the channel gives coherence time, the pilot symbol period must fulfill the following variation:

$$S_t \leq \frac{1}{f_{dp}} \qquad (1)$$

Comb-pilot composition is also depicted in Fig. 2. Each OFDM symbol has periodically inserted pilot tones between the subcarriers that are used for the channel estimation in frequency-domain interpolation. Since the inverse of the maximum delay spread $\sigma_{max}$ , gives the coherence bandwidth, the pilot symbol period must fulfill the following variation:

$$S_f \leq \frac{1}{\sigma_{max}} \qquad (2)$$

Blind channel estimation requires no pilot symbols and instead relies upon received symbols intrinsic data. Though this estimation technique does not waste any of the signal bandwidth, the computations are significantly more complex and lead to increased latency. For example, nearly 100 symbols are required to approximate one channel coefficient. For this reason, in real-world wireless communication systems this blind channel-estimation method is infrequently used. DDCE uses both detected data symbols and pilot symbols for channel estimation.

In our study we are using standard pilot-assisted channel estimation technique i.e. Linear Square (LS) [17]. The simple LS equalizer algorithm is regularly used as a standard model in typical hardware implementations. It treats IEEE 802.11a as block pilots for channel estimation for the long training sequence. Let us assume that the two long-term preambles after finding the starting of the frame are denoted as

$$\underbrace{y_G[n_p-128],\ldots,y_G[n_p-65]}_{Long\ Preamble\ 1},\underbrace{y_G[n_p-64],\ldots,y_G[n_p-1]}_{Long\ Preamble\ 2}$$

and denote them as T1 and T2. Time-domain symbols T1[n] and T2[n] are obtained from these two LPs, for LS channel estimation. Then, their N-point DFTs are computed as follows.

$$Y_1(k) = \sum_{n=0}^{N-1} T_1[n]\, e^{-\frac{2\pi jkn}{N}} \qquad (3)$$

$$Y_2(k) = \sum_{n=0}^{N-1} T_2[n]\, e^{-\frac{2\pi jkn}{N}} \qquad (4)$$

The two-training symbol N-points are equal, i.e., X1(k) = X2(k) = X(k). The LS estimate for H(k) is then given as below:

$$\hat{H}(k) = \frac{(Y_1(k)+Y_2(k))}{2X(k)} \qquad (5)$$

After channel estimation is done, the data in the packet are equalized. The DFT of the received signal is represented as Y(k). By equalizing the received DFT vector, the estimate of the transmitted data is

$$\hat{X}(k) = \frac{Y(k)}{H(k)} \qquad (6)$$

In the frequency domain, each subcarrier follows this simple, one-tap equalizer process. All the symbols in the packet follow this procedure.

### B. Frame Detection

For frame detection it requires n complex multiplications per baseband sample for n length of the correlation sequence. Schmidl-Cox is a more efficient method for frame detection, which performs the autocorrelation of the short preamble [18]. Let the autocorrelation for baseband sample stream YG with a lag of 16 samples which is the length of repeating pattern of short preamble be a. Then

$$a[n] = \sum_{k=0}^{N_{win}+15} y_G[n+k]\overline{y_G}\,[n+k+16] \qquad (7)$$

Here, $\overline{y_G}$ denotes the complex conjugate of $y_G$ and for moving average the parameter $N_{win}$ was used, which was set to 48. According to the size of OFDM symbol average of 64 samples including the lag of the autocorrelation was taken. To calculate the autocorrelation coefficient c, the value was normalized with average input power level p because the absolute value of the autocorrelation depends on the input power level and the gain setting of the device.

$$p[n] = \sum_{k=0}^{N_{win}-1} y_G[n+k]\overline{y_G}[n+k] \qquad (8)$$

$$c(n) = \frac{|a[n]|}{p[n]} \qquad (9)$$

Here $|a[n]|$ denotes the magnitude of a. There will be autocorrelation coefficient plateau during starting of the frame because during short preamble the autocorrelation stays high and the average was not done for the whole preamble length. Once the correlation exceeds a predefined threshold, which defines the sensitivity of the transceiver, then frame detection was triggered.

It causes unwanted signal processing and the receiver could start on noise, if the threshold value was fixed at too low. It causes missing of frames by the receiver and poor performance, if the threshold value was fixed at too high.

So, based on a parameter study a threshold was selected. After frame detection, for aligning the FFT in the receiver and decoding the frame, deriving the position of the OFDM symbol is required. Long preamble is used for deriving the position, by cross correlating detected LP with the known time-domain pattern of the LP. This pattern covers across 64 samples and repeats 2.5 times during the long preamble. The result after cross correlating the LP, was 64 complex multiplications per sample. The correlation was done using a small window of length Npreamble. This was done for 320 samples which was the total length of short and long preamble. Cross correlating the signal with the preamble generates very confined peaks that generates accurate alignment. After cross-correlating, for determining the frame start, the receiver searches for three highest values Np according to their magnitude. This was done by using argument matrix (argmax3).

$$\mathrm{Np} = argmax_{3_{n\in\{0,\ldots,N_{preamble}\}}} \sum_{k=0}^{63} y_G[n+k]\overline{LP}[k], \quad (10)$$

64 samples after the last peak will be the first OFDM data symbol, because this last peak is the highest sample index.

$$\mathrm{np} = \max(\mathrm{Np}) + 64, \quad (11)$$

Here np represents the starting of the frame.

## III. EXPERIMENTAL SETUP

In this section, we describe the components used for experimental setup. First, we describe the technology used for transmitter and receiver, after we describe the implementation of the transmitting and receiving using this setup. The paper setup consists of a transmitting side and a receiving side however; both are designed in the platform of software defined radio.

### A. Transmitter

Signal processing is carried out as follows in the TX flow-graph. Binary data from a specified file is read by the File Source block. The binary information has been transformed into streamed information via the Tagged Stream block. The stream tagged utilizes tags to define the boundaries of packets. The CRC-32 Stream block attaches a 32-bit cyclic redundancy (CRC-32) control to the tagged stream. The Packet Header block generates one header per packet. Packet configuration and header are displayed in Figure 3. The Repack Bits block converts 8 bits per byte to 2 bits or 1 bit per byte and modulates the payload by either quadrature phase shift keying (QPSK) or binary PSK (BPSK), respectively. A block of Virtual Sink relates to the same ID to the Virtual Source block. BPSK modulates the header and QPSK or BPSK modulates the payload in the block of chunks to symbols. In the Tagged Stream Mux block, the header and payload are concatenated. The OFDM Carrier Allocator block assigns modulated information symbols to the relevant sub-carriers, inserts pilot sub-carriers, and attaches the short preamble and long preamble. The parameters of this block are Occupied Carriers, Pilot Carriers, Pilot Symbols, and Sync Words. The parameter for Occupied Carriers contains the range of subcarriers used. The parameter Pilot Carriers assigns pilot subcarrier IDs and the parameter Pilot Symbols assigns pilot subcarrier modulated symbols. The Sync Words parameter provides the modulated symbol sequences for the short and long preambles which are the same as the IEEE 802.11a standard but has half the length of 802.11a preambles. The FFT block operates in the TX flow-graph as an inverse fast Fourier transform (IFFT). The OFDM Cyclic Prefixer attaches an OFDM symbol with the guard interval (GI). The Constant Source block offers interframe space (IFS) and one OFDM and one IFS are concatenated by the Stream Mux block. This IFS has a quarter of the length of the OFDM symbol in this document. The Multiply Const block multiplies the signal by 1=p 520 as the transmit signal should be within the transmitter's dynamic range. The block Tag Gate removes the tags from the recorded information. The UHD: USRP Sink block controls USRP and UHD for the transmission of digital radio IQ information.

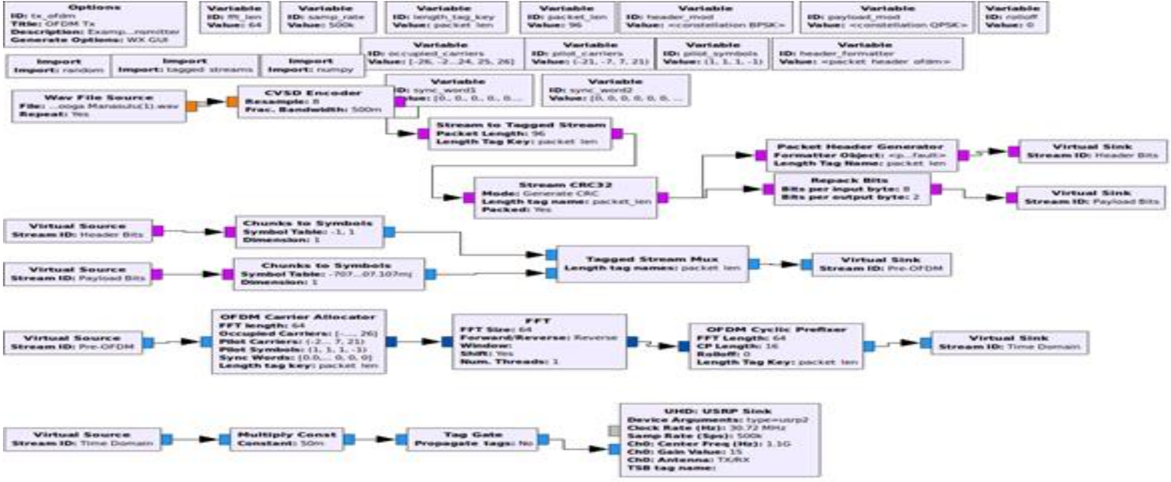

Fig.3. Transmitter Block Diagram

### B. Receiver

Signal processing is performed as follows in the RX flow-graph. UHD: the Source Block of the USRP controls the digital IQ on the radio. The symbol and the frequency offset of the RX signal is removed by the Schmidl & Cox OFDM Synch. block. The block Delay delays 1 OFDM symbol in the RX signal. Frequency offset is modulated by Frequency Mod (FD) via the Frequency Mod block and compensated by a Multiplexing block for the late RX signal. The RX is divided into headers and payload streams via the Demux header / payload block. Signal processing is performed in the header stream before the payload stream. The FFT Block is an FFT that matches a demodulator of the OFDM. The estimate block for each sub-carrier with the prevalent lengthy preamble and the equalizer for the channel state data for the header works with the estimated CSI. OFDM channel estimating block estimates the channel state data (CSI). The OFDM Serializer block is used to remove and convert pilot subcarriers into serial. The Decoder Constellation block demodulates the BPSK header. The Parser Block Packet Header transforms header information as tags into packet information. Except for obtaining estimated CSI from the tags and demodulate payload modulation the signal processing in payload stream is almost identical to that in the header stream. The Bit repack transforms the byte order bits into 8 bits per byte. Stream CRC-32 transmits RX information if there is no CRC-32 error. The packet sequence number on a console is displayed in the Tag Debug block.

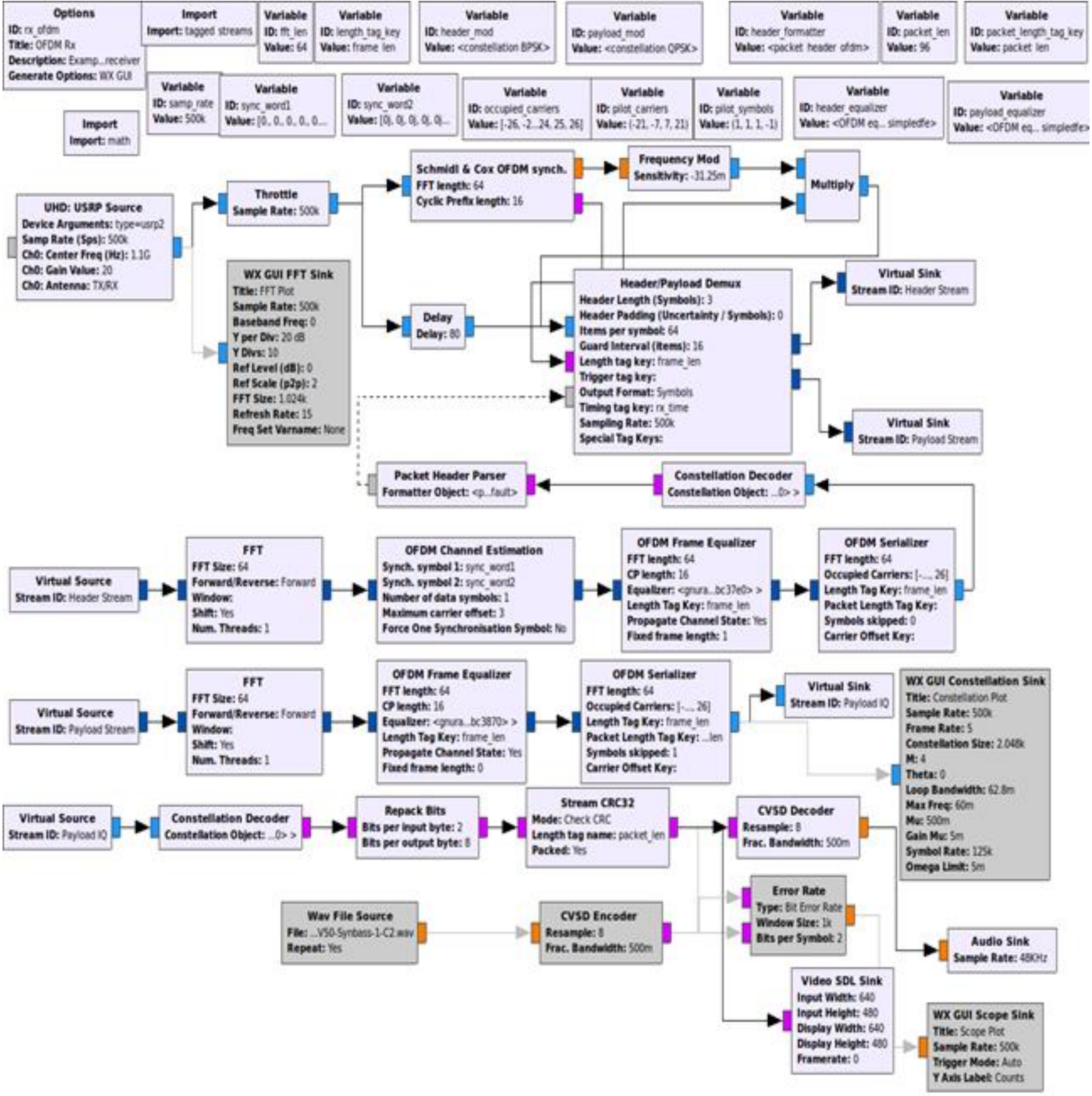

Fig.3. Receiver Block Diagram

### C. Experimental setup

We developed a real-time SDR using two USRPs for the transmitter and receiver connected to a personal computers (PCs) running the GNURadio software. Here, we considered an indoor environment where no other Wi-Fi signals are passing, and the length between and receiver is approximately one meter. The SDR architecture includes three sections for radio frequency (RF), intermediate frequency (IF), and baseband. The RF signal received by a smart antenna is sent to the USRP. GNURadio is used for flexible baseband signal processing. Two PCs were used to run the SDR software for implementing a real-time radio environment. Table 1 and Table 2 list the most important software and hardware components that we used, respectively. The IEEE 802.11a PHY parameters that we used are listed in Table 3.

**Table 2 PC components used in the setup**

| PC Component | Type |
|---|---|
| CPU | Intel core i7-8550U |
| RAM | 16 GB |
| Operating System | UBUNTU 16.04 |
| GNURadio | 3.7.10.1 |
| UHD | Version 003 011 000 000 |

**Table 3 List of the hardware components in our setup**

| Hardware Component | Type |
|---|---|
| SDR Transmitter | USRP N210 |
| SDR Receiver | USRP N210 |
| Transmitting Antenna | Dipole antenna |
| Receiving Antenna | Dipole Antenna |

**Table 4 PHY parameters used in OFDM implementation**

| Parameter | Measurement |
|---|---|
| Bandwidth | 20 MHz |
| OFDM subcarrier | 64 |
| Subcarrier Spacing | 312 KHz |
| OFDM Symbol time | 4 μs |
| Guard time | 1.6 μs |
| Comb-pilot spacing | 4.4 MHz |
| Center frequency | 2.2 GHz |

There are defined variables in the OFDM system, such as sampling frequency, Sync word 1, Sync word 2, pilot symbols, occupied carriers etc. that are used in the OFDM transmitter and receiver. Three kinds of data are examined: picture, audio and text stated in the File Source block. The information from the output is transmitted through a CRC32 error divided into header bits and payload bits which use the BPSK and QPSK modulations. The error is also forwarded to the Tagged Stream Block, where tags are added at even intervals. The output data is sent to the OFDM Carrier Allocator block, the occupied carrier and the Cyclic Prefixed are given here. It is sent with a USRP block, in which the fundamental parameters such as sample rates, frequency, acquisition, etc. are configured. The frequency was 2.2 GHz. In the OFDM receiver, we can visualize four important parts. First, a Schmidl & Cox OFDM synchronizer has been used for the USRP source and the synchronizer, the samples are immediately piped into the header / payload block Demux, and a trigger signal is sent to indicate the start of the frame. The channel estimates are then entered into the header, the FFT transfers OFDM symbols to the frequency domain and the results are transmitted to the Estimated Channel block. The channel estimation and frequency offset of the preamble specimens are performed by this block OFDM. The symbols used to assess the equalization factor are synchronized first, then these symbols are removed and the other symbols are sent from the block. When the OFDM symbol stream is send to the OFDM frame equalizer, it equalizes the raw frequency offset of the carrier in one or two sizes for an OFDM frame. When the OFDM symbolize stream is send out on the OFDM Frame Equalizer, one or two OFDM frames are equalized and the coarse frequency offset of the carrier is corrected. The header bits are then interpreted with the Parser packet header block and the results are feedback to demuxers for the purpose of enabling them to know the payload length and the tagged stream. The demodulation of payload does not need to do this because the data about the channel status is on the marked payload stream. In the end, we acquired the image, audio and text sent in the transmission in an data recipient in the Output File.

In the Fig. 4 can be see the hardware implemented to realize the proof of concept, both computers have installed the graphics interface of the GNU Radio and they are connected to their corresponding USRP. The left computer is doing the information transmission, in this case the information is a audio, in the other hand, the right computer is doing the signal reception and can be listened that a good quality audio was received as well as the OFDM spectrum. In the same manner the text and image files are also send to the same setup and we will get the data at the receiver end. In the audio transmission we can listen the song directly at the time of reception. But for image and text the result will be stored in a file and we can compare the result using those files.

The system is tested by increasing the distance between the transmitter and the receiver. At the time also the system can recover the data with some error. The channel estimation and synchronization blocks play a major role in the receiver section.

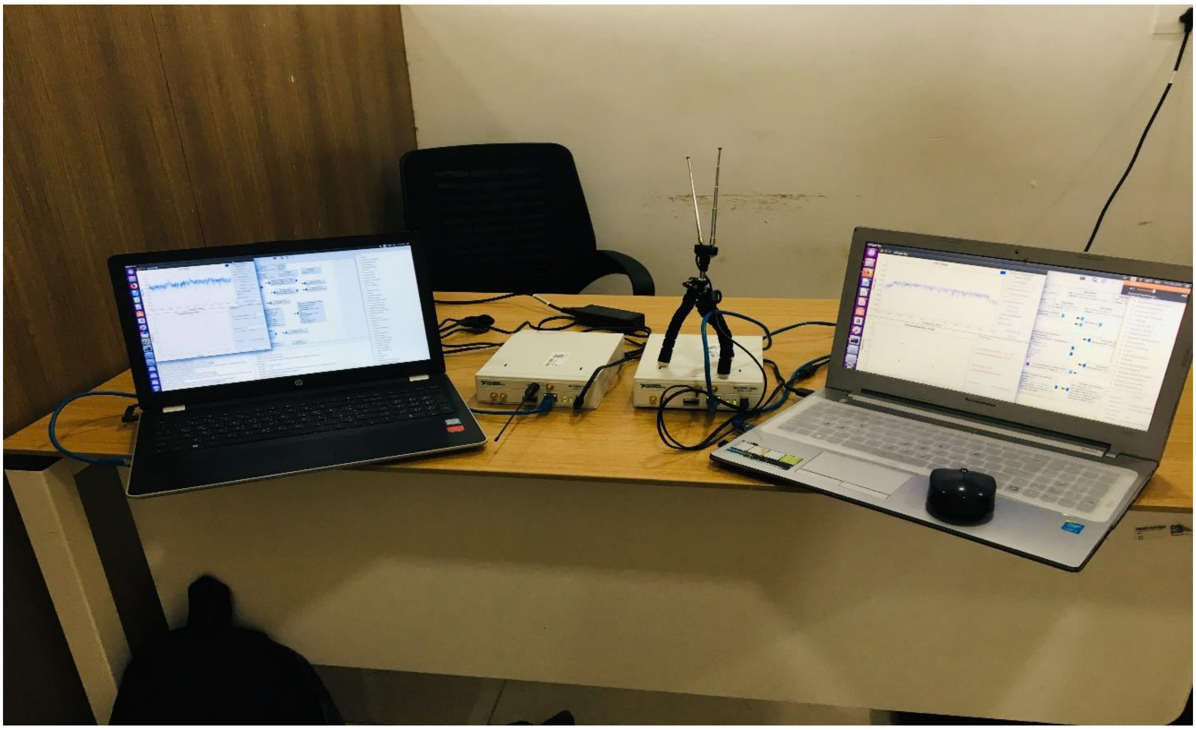

**Fig.4. Photograph of the experimental setup for a real-time radio environment**

## IV. EXPERIMENTAL RESULTS AND DISCUSSION

Table 2 shows the necessary parameters and their values for simulation / implementation purposes. The FFT length 512 is maintained here, giving the highest performance with test and error.

**Table 5 PHY parameters used in OFDM implementation**

| System parameter | Value |
|---|---|
| RX gain | 20 dB |
| Bandwidth | 5MHz |
| FFT size | 64 |
| Guard Interval size | 16 |
| Bit rate | 6 Mbits/Sec |
| Header modulation | BPSK |
| Payload modulation | QPSK |
| Payload size | 96 bytes |
| Distance between antennas | 0.8 |

In this paper we have transferred the three types of data using the SDR based real time setup. We compared the send and received data lively. We done four types payload modulation techniques like BPSK, QPSK, 16QAM and 64QAM. Here we are displaying the FFT plot, scope plot, constellation plot and waterfall for QPSK based received signal in Fig. 5,6,7 and 8. Instead of displaying the plots for all the data types we are considering audio signal as transmitted signals and displaying the plots for the particular audio signal.

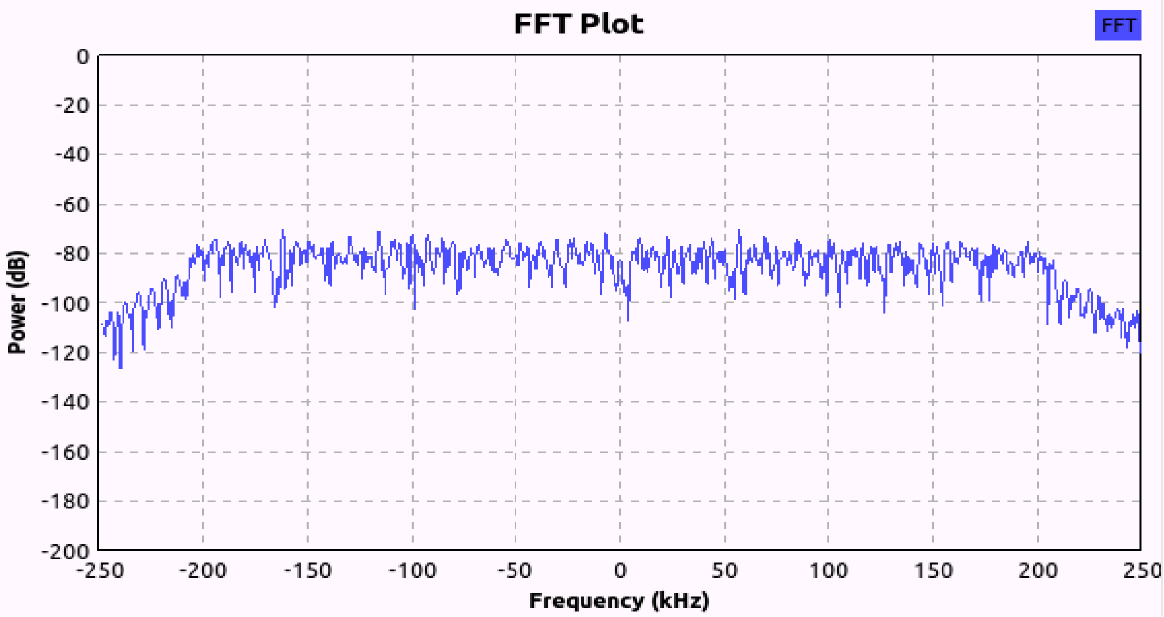

Fig.5. FFT Plot for received audio signal

Fig.6. Scope Plot for received audio signal

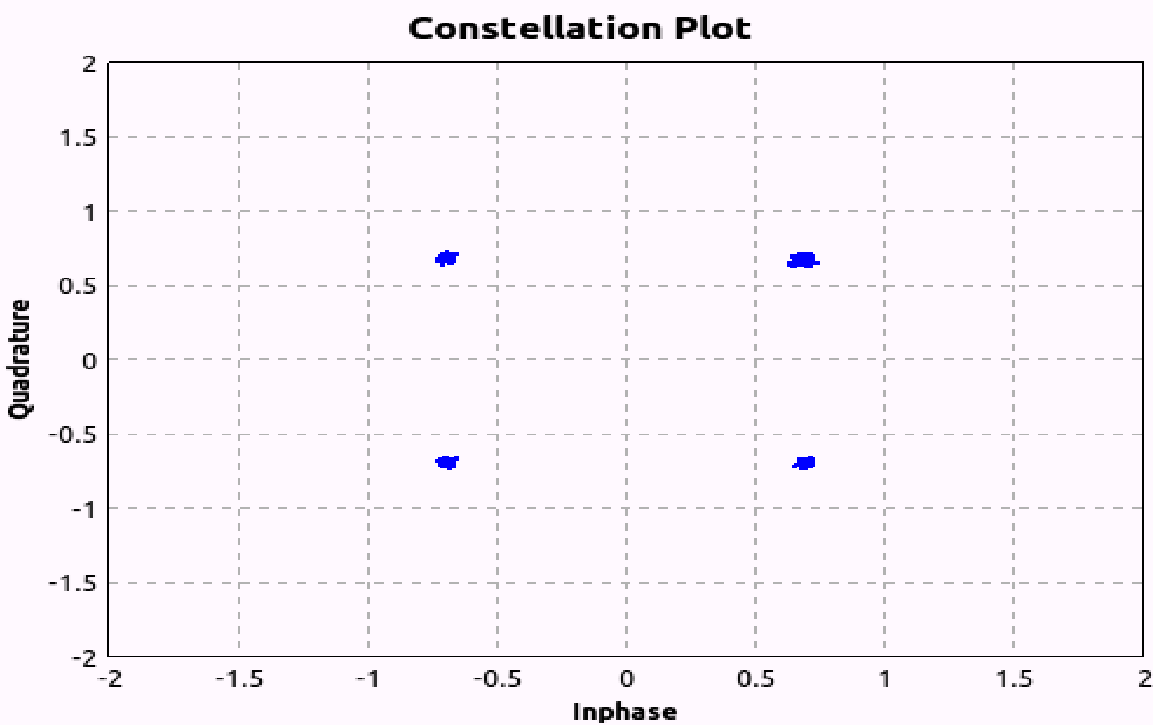

Fig.7. Constellation Plot for received audio signal

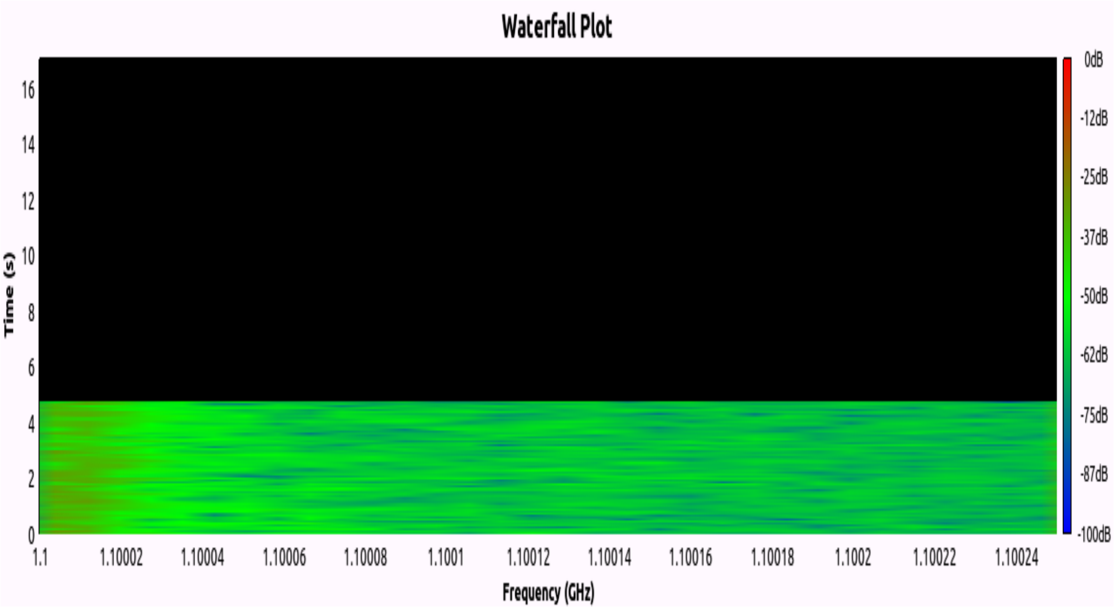

Fig.8. Waterfall for received audio signal

## V. CONCLUSION

The OFDM results prove that SDR is the best solution for next generation technology. By using a low-cost desktop setup for implementing physical layer of 802.11a, it was shown that this is one of the solutions for researchers to test their techniques in real environment. The real time results prove that BPSK is the best modulation technique to transfer the information with less BER even at low SNR. In future using this setup is used to implement a novel technique for channel estimation and synchronization and compare them with standard techniques.

## REFERENCES

1. Albreem MAM. "*5G wireless communication systems: vision and challenges*". I4C, pp. 493–97, April 2015.
2. Lee J, Ryu J, Lee S-J, Kwon T. "*Improved modeling of IEEE 802.11a PHY through fine-grained measurements*", Computer Networks: The International Journal of Computer and Telecommunications Networking, Vol. 64, No.4, pp.641–5, March 2010.
3. Giddings RP, Jin XQ, Tang JM. "First experimental demonstration of 6Gb/s real time optical OFDM transceivers incorporating channel estimation and variable power loading". Optics Express, Vol.17, No.22, pp. 19727-19738, 2009.

4. Jin XQ, Giddings RP, Tang JM. "*Real-time transmission of 3Gb/s 16-QAM encoded optical OFDM signals over 75km SMFs with negative power penalties*". Optical Express, Vol.17, No.17, pp. 14574-14585, 2009.
5. Li R, Dou Y, Zhou J, Deng L, Wang S. CuSora: "*Real-time software radio using multi-core graphics processing unit*". Journal of Systems Architecture, Vol. 60, No.3, pp. 280-292, March 2014.
6. Leonardo A. Agüero Guzmán, Elias M. Ovalle, and Rodrigo A. Reeves, "*Measurement of the ionospheric reflection height of an HF wave in vertical incidence with a resolution of minutes*", IEEE Geoscience And Remote Sensing Letters, Vol. 15, No.11, pp. 1637–1641, November 2018.
7. Wei Wang, Yinejie Chen, Zeyu Wang, Jin Zhang, Kaishun Wu, and Qian Zhang, "*Wideband Spectrum Adaptation Without Coordination*" IEEE Transactions on Mobile Computing, Vol. 16, No.1, pp. 243-256, January 2017.
8. Wang W, Chen Y, Wang L, Zhang Q, "*Sample less Wi-Fi: bringing low power to Wi-Fi communications*", IEEE/ACM Transactions on Networking, Vol. 25, No.3, pp. 1663-1672, June 2017.
9. Suman Bhunia, Edward Miles, Shamik Sengupta, and Felisa Vazquez-Abad, "*CR-Honeynet: A Cognitive Radio Learning and Decoy Based Sustenance Mechanism to Avoid Intelligent Jammer*", IEEE Transactions on Cognitive Communications and Networking, Vol. 4, No.3, pp. 567-581, September 2018.
10. Song Fang, Yao Liu, and Peng Ning, "*Wireless Communications under Broadband Reactive Jamming Attacks*", IEEE Transactions on Dependable and Secure Computing, Vol. 13, No.3, pp. 394-408, June 2016.
11. Cassola A, Jin T, Noubir G, Thapa B "*Efficient spread spectrum communication without pre-shared secrets*", IEEE Transactions on Mobile Computing, Vol. 12, No.8, pp. 1669-1680, 2013.
12. Vilches T, Dujovne D, "*GNURadio and 802.11: performance evaluation and limitations*", IEEE Network, Vol. 28, No.5, pp. 27-31, October 2014.
13. Horrein P-H,·Hennebert C, Pétrot F, "*Integration of GPU computing in a software radio environment*", Journal of Signal Processing Systems, Vol. 69, No.1, pp. 55-65, October 2012.
14. Lee S, Choi J, Yoo J, Kim C-K, "*Frequency diversity-aware Wi-Fi using OFDM-based bloom filters*", IEEE Transactions On Mobile Computing Vol. 14, No.3, pp. 525-537, March 2015.
15. M. Dohler, R. W. Heath, A. Lozano, C. B. Papadias, and R. A. Valenzuela, "*Is the PHY Layer Dead?*" IEEE Communications Magazine, vol. 49, no. 4, pp. 159–165, Apr. 2011.
16. Kim H. *Wireless Communications Systems Design*. John Wiley & Sons; 2015.
17. Van Nee R, Prasad R. *OFDM wireless multimedia communications*. Boston: Artech House; 2000.
18. T. Schmidl and D. Cox, "*Robust frequency and timing synchronization for OFDM*," IEEE Transactions on Communications, vol. 45, no. 12, pp. 1613– 1621, 1997. DOI: 10.1109/26.650240.

## AUTHORS PROFILE

**Ponnaluru Sowjanya** completed her B.Tech (Electronics and Communication Engineering) in Sri Mittaplli Institute of Technology for Women, Tummalapalem, Guntur, India; M.Tech (Embedded Systems) in Nalanda Institute of Engineering and Technology, Kantepudi, Satenepalli, India. At present, she is doing the Ph.D. degree in Electronics and Communication Engineering at Koneru Lakshmaiah Educational Foundation, Vaddeswaram, India. Her research interest includes Wireless Communication and Signal processing.

**Dr. Penke Satyanarayana** received a B.Tech in Electronics and Communication Engineering (ECE) from Koneru Lakshmaiah College of Engineering, Vaddeswaram, Vijayawada, India; M.Tech in Computers and Communications Systems from JNTU, Hyderabad, India; and Ph.D. in ECE from JNTU, Kakinada, India. He is working as a Professor in the Department of ECE, K L E F, Guntur, India. He has published over 20 research papers in reputed international and national journals and conferences. He is a member in Indian Society of Technical Education (ISTE). His research interests include Wireless Communication and Signal processing, Embedded Systems and VLSI.