Software Architecture of Software-Defined Radio (SDR)

Bambang Riyanto T.[1], Armein ZR. Langi[2], Adit Kurniawan[2],
Eko Marpanaji[2], Andri Mahendra[2] , Thay Liung[2]
ITB Research Center on ICT
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
Tel. +62-22-425-4034
[1]briyantot@lskk.ee.itb.ac.id, [2]langi@dsp.itb.ac.id, [2]adit@ltrgm.ee.itb.ac.id
[2]ekoaji332@students.itb.ac.id, [2] mahendra@dsp.itb.ac.id,[2] thay148@yahoo.com

Abstract— **This paper addresses software architecture of Software-Defined Radio (SDR) and observation of SDR software performance which is implemented on SDR platform using Universal Software Radio Peripheral (USRP) from GNU Radio to perform radio functions with GMSK and DQPSK modulation scheme. Its performance is observed in Packet Error Rate (PER) and evaluated in terms of Eb/No or S/N ratio, carrier frequency, bit rate, gain, filter coefficient, and of size of payload from delivered data. Based on this results, the smallest PER could be obtained by setting Eb/No value > 20 dB for both GMSK and DQPSK modulations, optimum bit-rate 256 kbps for GMSK modulation and 200 kbps for DQPSK modulation , carrier frequency higher than 0.1 MHz for GMSK modulation and higher than 0,3 MHz for DQPSK modulation, BT of Gaussian filter larger than 0.25 for GMSK modulation, and roll-off factor (alpha) for DQPSK modulation equal to 0.4, optimum gain is 400 for GMSK modulation and 500 for DQPSK modulation, payload size of GMSK modulation larger than 1024 byte, and optimum payload size of DQPSK modulation is 4092 byte.**

Index Terms— **Differential quadriphase shift keying, gaussian minimum shift keying, modulation, packet error rate, root raised cosine, software-defined radio, usrp.**

I. Introduction

Software-Defined Radio (SDR) represents communication technology that has been developing in order to anticipate change of compatibility issues among standards. By using SDR, it is expected that the hardware investment caused by the change of communication standard can be reduced so that low cost communication service can be achieved. The existence of SDR technology has been much awaited for developing future communication system which is predicted to emphasize on wireless communication, including development of mobile phone and third generation (3G), Next generation networks (NGN), and Rural-Next Generation Network (R-NGN). Due to its flexibility, reprogrammability, scalability, and low investment/cost, SDR technology is expected to be suitable for telecommunications system in rural areas through-out Indonesia.

Commercial wireless communication industry is currently facing problems due to constant evolution of link-layer protocol standards (2.5G, 3G, and 4G), existence of incompatible wireless network technologies in different countries inhibiting deployment of global roaming facilities and problems in rolling-out new services/features due to wide-spread presence of legacy subscriber handsets[1].

SDR technology promises to solve these problems by implementing the radio functionality as software modules running on a generic hardware platform. Further, multiple software modules implementing different standards can be present in the radio system. The system can take up different personalities depending on the software module being used. Also, the software modules that implement new services/features can be downloaded over-the-air onto the handsets. This kind of flexibility offered by SDR systems helps in dealing with problems due to differing standards and issues related to deployment of new services/features.

Development of communication system based on SDR requires thorough investigation on SDR architecture along with the required hardware and software. While SDR technology offers many advantages, at present no commercial SDR appearing on the market. This is mainly due to the huge of computing resources needed to run the software performing various radio functions. The proposed research will investigate the software architecture of software-defined radio as apart of main research in the development of computing platform of SDR using commodity component aimed at developing wireless communications in rural areas. This research is also continuation of preliminary research that was carried out under Riset Unggulan ITB 2006 which focused on hardware SDR architecture.

This paper describes results of the research in investigation of SDR software architecture and SDR software development for communication system. It is supposed that the software architecture of SDR resulted from this research can be used as basic concept in designing SDR platform because computation complexity and computation capacity can be determined from this software algorithm. Expected bit rate of SDR system is 256 kbps as mentioned in PPTIK road map with bit error rate (BER) less than 10^{-3}.

This research will answered the research question how to develop a SDR systems software which has bit rate 256 kbps with bit error rate less than 10^{-3}, and what kind of software architecture which is suitable for integration with recent communication protocol such as TCP/IP. The result of this research will give us a chance in developing of wireless communication system which cable network connection is impossible to be used due to geographical problems. Expected result of this research is a architecture of software prototype of digital wireless communication system based on SDR which is needed in developing of rural area system communication. Future work of this research is software integration with recent communication protocol such as TCP/IP protocol or VoIP protocol for IP based voice communication.

The rest of this paper is organized as follows. Section 2 presents the architecture of SDR systems. Section 3 presents system configuration used in this research. Section 4 presents software implementation (especially for modulation and demodulation) and test result. Finally, Section 5 is conclusion.

II. SDR Architecture

Software-defined radio (SDR), sometimes shortened to software radio (SR), was introduced for the first time in 1991 by Joseph Mitola[2]. The word of SDR was used to show a radio class that could be re-configured or re-programmed[3], thus resulted a kind application of wireless communication with mode and frequency band determined by software function. Ideally, SDR offers flexibility, re-configurability, scalability and as multi mode as possible.

SDR architecture is developed based on conventional radio functions. The difference is all functions of signal processing on conventional radio are carried out fully by hardware while the functions of signal processing on SDR are carried out as much as possible by software. The major key in building SDR is the placement of ADC and DAC components as a divider between analog and digital domain, thus the signal processing can be carried out using software.

SDR architecture can be viewed in term of hardware and software. Placement of ADC/DAC determines the analog and digital world. Analog signal is processed by hardware, while digital signal is processed by software. Hence, the ADC/DAC will seperate the analog signal and digital signal processing and usually called as digital access point. With digital access point, all radio functions can be processed through software. Fig. 1 shows the hardware and software architecture of SDR system.

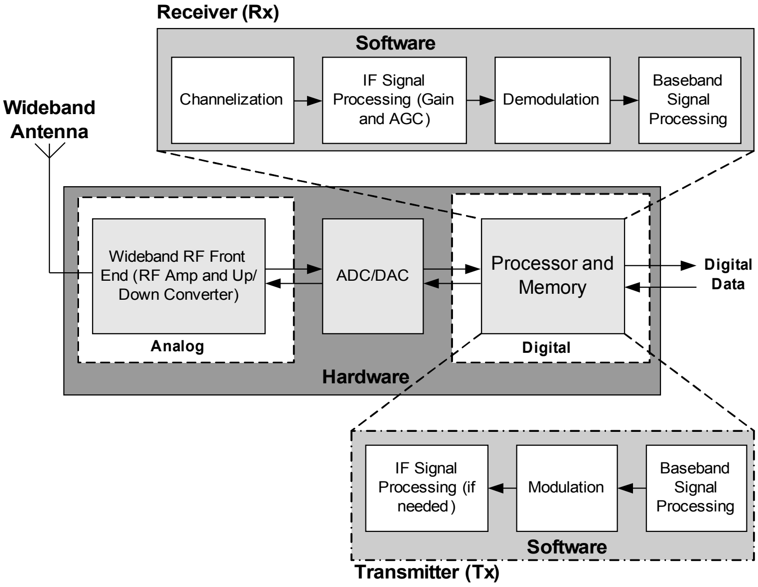

Receiver (Rx)
Software
Channelization
IF Signal Processing (Gain and AGC)
Demodulation
Baseband Signal Processing
Wideband Antenna
Wideband RF Front End (RF Amp and Up/ Down Converter)
ADC/DAC
Processor and Memory
Digital Data
Analog
Digital
Hardware
IF Signal Processing (if needed)
Modulation
Baseband Signal Processing
Software
Transmitter (Tx)

Figure 1. SDR Hardware and Software Architecture

Based on Fig. 1, the SDR architecture (both hardware and software) is divided into two components, transmitter (Tx) and receiver (Rx). In the hardware architecture, the main issues in developing of SDR platform architecture are ADC/DAC sampling rate and computation capacity of processor. To reduce computation complicity, one places the ADC/DAC component in IF section, so the hardware architecture is more realistic. The limitation of ADC sampling rate forced us to implement under sampling method[4] which is more difficult than Nyquist sampling. Nevertheless, this is one method that we can adopt to solve the sampling rate problem.

The software architecture consists of two processes, i.e. transmitting process and receiving process. In transmitter side, the process starts from base band signal processing to process the incoming data or information (in digital communication, the information usually refer as a bit stream), modulating process to modulate the information with IF carrier signal, converting the IF signal into RF signal, and finally sending the RF signal into communication channel (air) through the antenna. In receiver side, the process starts with selecting and down converting the RF signal into IF signal, digitizing IF signal, demodulating to extract the information signal from IF signal, and finally base band processing to obtain the real information which is sent by the transmitter.

In SDR systems, all processes is covered by software. Thus, the software implementation to perform radio functions in the SDR system offers some advantages especially in term of flexibility, reconfigurability, and reprogrammability. In terms of flexibility, for example, the SDR system can anticipate the change of modulation scheme without hardware replacement because the modulation is performed through software. In order to allow software integration with recent protocol communication standard, the developed software architecture is designed into layers in which each layer defines a certain function. The approach to designing reconfigurable radio using hardware paging is formalized in the layered radio architecture. Hardware paging refers to hardware modules being paged in and out of the system in a manner similar to software paging performed with the use of virtual memory. The functionality of the radio is divided into layers, where each layer attached modifies the header and passes the information to the next layer. Once the processing is complete, the information is sent back through the layers in the same way[5].

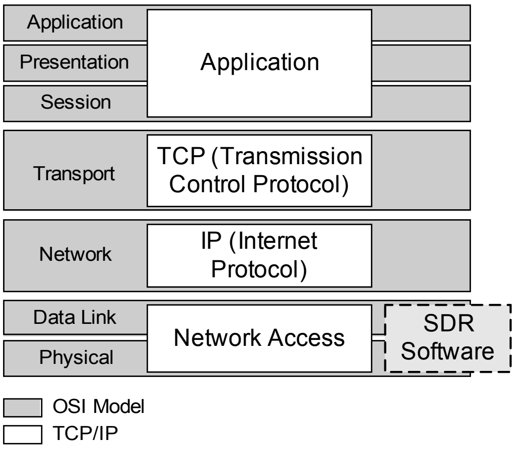


Figure 2. Software Architecture Layers

Principally, if we want to integrate the SDR software with common standard protocol, the designed SDR software must reside in one of that protocol layer. For example, the integration of SDR software with computer network protocol, the designed SDR software must reside in the Network Access layer of TCP/IP layers or Data Link layer of standard OSI model as depicted in Fig. 2. This layer provides the means by which access to the physical network medium is negotiated, as well as error checking and other basic functions.

Based on Fig. 2, the SDR software receives packets from IP layer and process the IP packet through radio functions as information inputs. Based on software SDR architecture, the radio functions consists of base band processing, modulation or demodulation, IF signal processing. While RF signal processing is performed by hardware.

III. System Configuration

System configuration which used in this research is shown in Fig 3. SDR platform in this research use USRP GNU Radio peripheral that function is up/down converter as a front end and personal computer (PC) as the processor. System configuration of SDR using PC as the processor was often referred as Software Radio or shortened by SWR. Pursuant to configuration, hence modulator and demodulator scheme can be implemented in software, while A /D and D/A converter remain in hardware form which located in USRP. USRP and PC is connected using port USB version 2.0.

Mainboard specification of USRP which used in this research is: (a) USB 2.0 port for connection to computer; (b) 12-bit ADC with speed of sampling 64 MSPS so that with principle of aliasing can conduct digitalization process with range frequency of aliasing - 32 MHz until 32 MHz; (c) 14-bit DAC with clock frequency 128 MSPS so have Nyquist frequency equal to 64 MHz; and (d) analog signal which yielded limited to 10 mWatt. While daughterboard using Basic Tx and Basic Rx so that there is no process of up/down converter and transmitter frequency limited 50 MHz maximum.

Tabel 2. Computer spesifications that used in this research

No	Component	1st Computer	2nd Computer
1.	Processor	AMD Athlon XP1800+, 1.53 GHz, fsb 533 MHz	Intel Pentium 4, 2.93 Ghz, fsb 533 MHz
2.	RAM	DDR 333 MHz, 256 Mbyte	DDR 400 MHz, 2 x 256 MByte
3.	Operating System	Linux Fedora Core-4, 2.6.11-1.i369	Linux Fedora Core-4, 2.6.11-1.i369

Specification of PC which used in this research is summarized at Table 2. Computer that used must have USB version 2.0 port to support connection with USRP board. Programming language used Phyton and a few block functions written in C++.

System performance can be measured using one of the parameter which was often used in QoS, that is:
BER < 10^{-3}, PLR or PER < 10^{-2}, spread delay < 100 ms, and Gos > 95% [3].

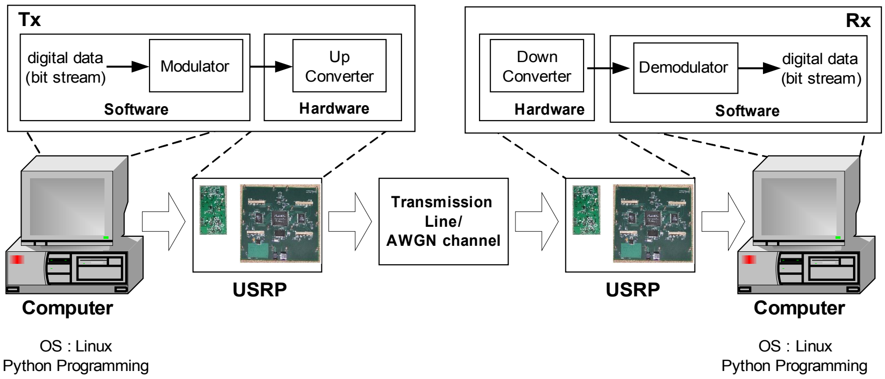


Fig. 3. System configuration of SDR platform

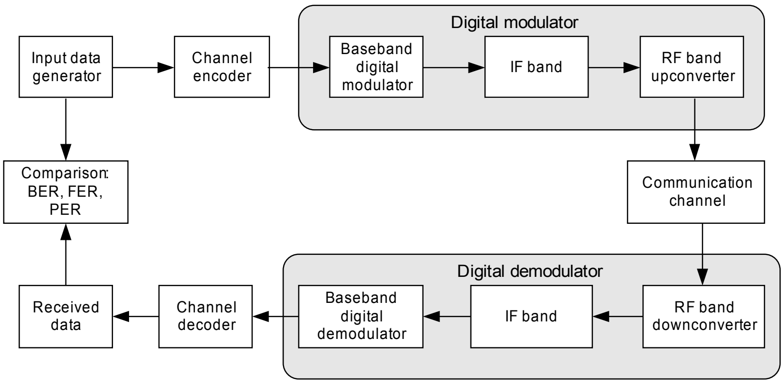


Fig. 4. Method of performance measurement

In this research, performance modulation scheme is measured in Packet Error Rate (PER) because data are transmitted in the form of packet. Transmitting data as a packet will facilitate of SDR software integration with TCP/IP protocol for next development. The method of performance measurement is depicted in Fig. 4.

IV. Implementations And Test Result

In this research, we measured the performance of GMSK DQPSK modulation scheme in order to investigated the SDR software using GNU Radio open source. The SDR software are implemented on SDR platform using *Universal Software Radio Peripheral* (USRP) from GNU Radio as shown at Fig.3. The Board is a front side of SDR system and has functions to execute digitalization function (ADC-DAC) and filter channelization process. The digital signal processing used personal computer with USB port as a link between USRP and PC. Operating system used Linux with python programming language to develop the application.

GMSK and QPSK or DQPSK are chosen in this research because these modulation scheme are widely used in recent wireless and cellular communication systems such as Satellite, CDMA, and cable modem. Software to perform radio functions can be developed using open source from GNU Radio with some modification.

Chen (2007) investigated a performance of *Quadriphase Shift Keying* (QPSK) modulation scheme to send a video digital data. All algorithms are modeled and simulated in Matlab, and the future works of that research is integrated those model into GNU Radio SDR platform using USRP[6].

This experiment would observe the performance of digital communication based on SDR system using USRP board with various of modulation scheme. The transmission channel of coaxial cable RG58 was used to connect transmitter and receiver because of limitation of Basic TX output power. The experiment was carried out to observe Packet Error Rate (PER) stated by the percentage of exactly accepted packet toward variation of Eb/No, bit rate, carrier frequency, filter coefficient, gain, and also payload size.

The result of performance measurement of each modulation is described as follow.

A. GMSK Modulation

GMSK is derived from MSK which replace the rectangular pulse with a sinusoidal pulse and apply Gaussian filter for pulse-shaping. The GMSK modulation has been chosen as a compromise between spectrum efficiency, complexity and low spurious radiations (that reduce the possibilities of adjacent channel interference). A Gaussian-shaped impulse response filter generates a signal with low side lobes and narrower main lobe than the rectangular pulse. Because this modulation used Gaussian filter for pulse-shaping then people calls this modulation as Gaussian Minimum Shift Keying (GMSK) modulation. The relationship between the pre-modulation filter bandwidth, B and the bit period, T defines the bandwidth of the system. GSM designers used a BT = 0.3 with a channel data rate of 270.8 kbps. This compromises between a bit error rate and an out-of-band interference since the narrow filter increases the inter-symbol interference and reduces the signal power.

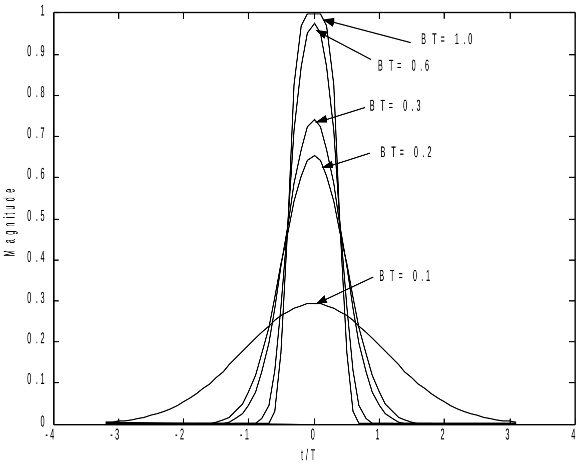


Fig. 5. Gaussian response filter with various of BT product

The response of Gaussian filter is determined by equation[7]:

$$g(t) = \frac{1}{2}\left[\operatorname{erfc}\left(\pi\sqrt{\frac{2}{\log 2}}\,BT_b\left(\frac{t}{T_b}-\frac{1}{2}\right)\right)-\operatorname{erfc}\left(\pi\sqrt{\frac{2}{\log 2}}\,BT_b\left(\frac{t}{T_b}-\frac{1}{2}\right)\right)\right] \quad (1)$$

Based on Equation (1), the filter response of Gaussian filter for several chosen values of BT is computed using MATLAB, and is plotted in Fig. 5.

Fig. 6 shows the principle of a GMSK modulator. The process occurred on modulator starts by changing bit stream $x(n)$ with level 0 and one into bipolar stream bit $x'(n)$ with level -1 and 1. Next, bipolar stream bit is processed by Gaussian filter possessing $g(t)$ filter. This filter response is computed using convolution between bipolar bit streams $x'(n)$ and $g(t)$. The result of convolution is then integrated in order to compute the phase $\varphi(t)$. This phase will determine I-channel and Q-channel signals through equation:

$$I = \cos(\varphi(t)) \quad (2)$$

$$Q = \sin(\varphi(t)) \quad (3)$$

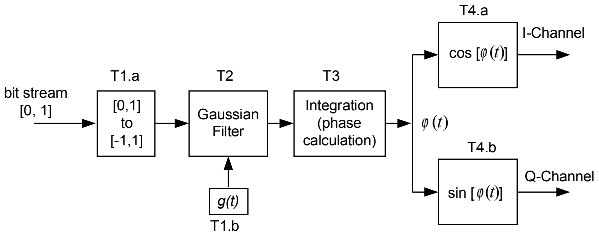


Fig. 6. GMSK Modulator

These I-channel and Q-channel signals will be transmitted to the air and will be processed by GMSK demodulator detailed in the following section.

The process occurred on GMSK demodulator in order to completely regain the information signal of bit stream is shown in Fig. 7. I-channel and Q-channel signals will be processed partly in case of its filtering and synchronization. The next process is partly phase computation for data rows exhibiting odd and even index numbers from the result of synchronization of I-channel and Q-channel signals to detect bit rows with previous level 0 and 1. Finally, the complete bit rows can be gained by joining both of bit rows, and therefore information signal of bit rows is regained according to what is sent by GMSK modulator. The detail algorithm of GMSK demodulator can be seen in[8].

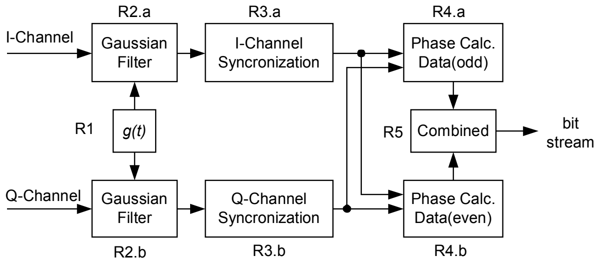
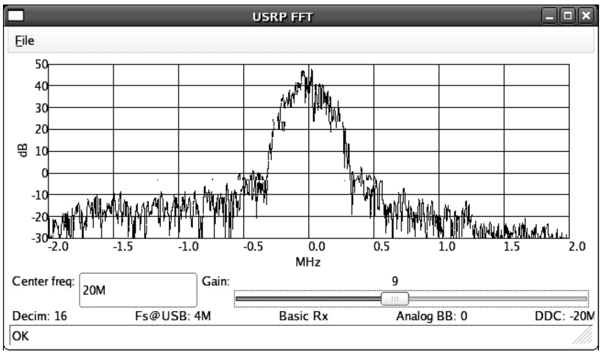


Fig. 7. GMSK Demodulator

Fig. 8 shows the GMSK spectrum, and the result of investigation is shown in Fig. 9 up to Fig. 13. Referred to all curves of GMSK performance, we conclude that the lowest PER value can be achieved by setting Eb/No larger than 20 dB, bit rate equal to 256 kbps, BT larger than 0.25, carrier frequency larger than 0.1 MHz, gain larger than 400 and data size for each packet larger than 1024 bytes.



Center freq: 20M

Gain: 9

Decim: 16 Fs@USB: 4M Basic Rx Analog BB: 0 DDC: -20M

OK

Fig. 8. GMSK Spectrum

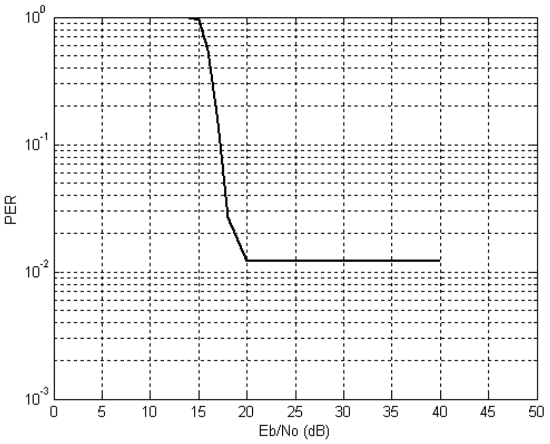


Fig. 9. PER with various of Eb/No

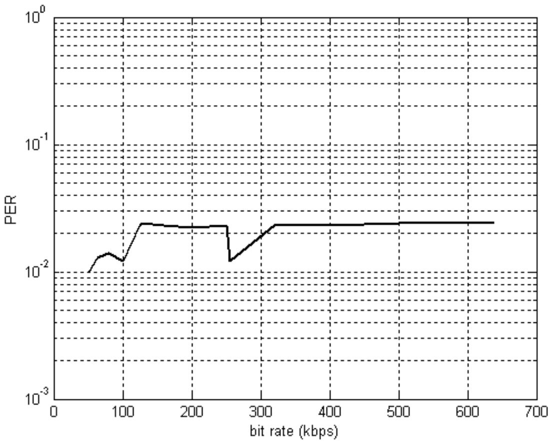


Fig. 10. PER with various of bit rate

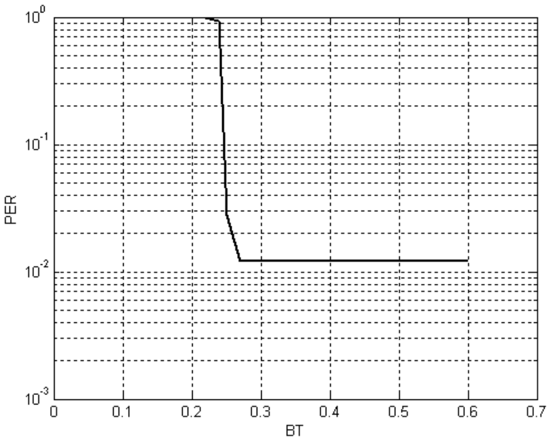


Fig. 11. PER with various of BT

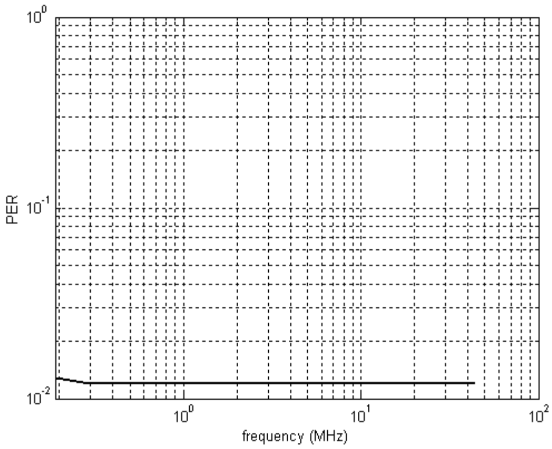


Fig. 12. PER with various of carrier frequency

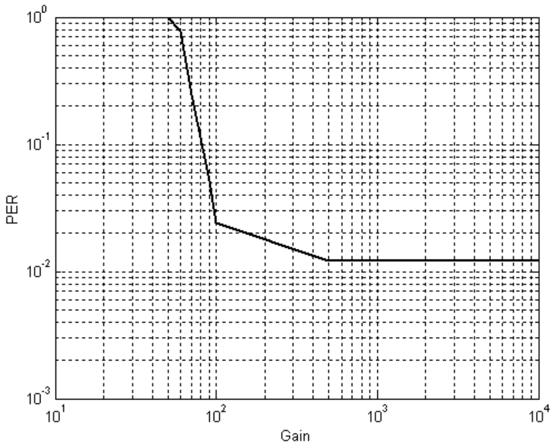


Fig. 13. PER with various of gain

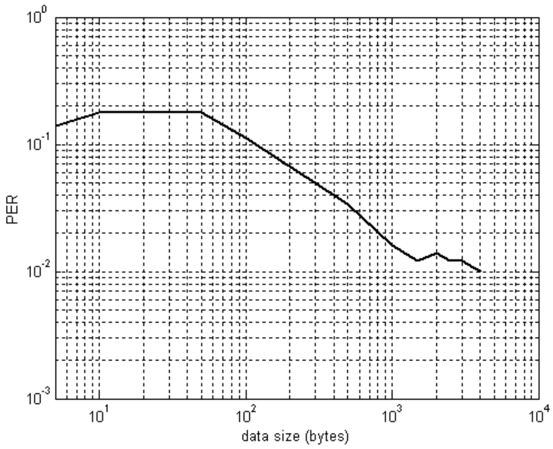


Fig. 14. PER with various of data size

B. DQPSK Modulation

DQPSK modulation is a variant of $\pi/4$-QPSK modulation scheme, which differential encoding is added to the bits encoding prior to the modulation. Differential means that the information is not carried by the absolute state; it is carried by the transition between states. The advantage of using differential encoding is free from phase ambiguity if the constellation is rotated by some effect in the communications channel which the signal passes through. This problem can be overcome by using the data to change rather than set the phase. The QPSK modulation scheme using differential encoding is usually called as $\pi/4$-DQPSK or DQPSK.

The equation differential encoder to produce symbols is:

$$s_k = s_{k-1} \oplus b_k \qquad (2)$$

where s_k is current output symbol, s_{k-1} is previous output symbol, and b_k is current bit input. The $\oplus$ sign indicate modulo-2 operation. So s_k only changes state (from binary '0' to binary '1' or from binary '1' to binary '0') if b_k is a binary '1'. Otherwise it remains in its previous state. At the receiver, the received signal is demodulated to yield $s_k = \pm 1$ and then differential decoder reverses the encoding procedure (decoding process) and produces:

$$b'_k = s_k \oplus s_{k-1} \qquad (3)$$

since binary subtraction is the same as binary addition. Therefore, $b'_k = 1$ if s_k and s_{k-1} differ and $b'_k = 0$ if they are the same. Hence, if both s_k and s_{k-1} are *inverted*, b_k will still be decoded correctly as b'_k. Thus, the 180° phase ambiguity does not matter.

The DQPSK modulation format uses two QPSK constellations offset by 45 degrees (π/4 radians) as depicted in Fig. 15. Transitions must occur from one constellation to the other, or each DQPSK symbol will reside in one of eight points in the constellation diagram. This guarantees that there is always a change in phase at each symbol, making clock recovery easier. The data is encoded in the magnitude and direction of the phase shift, not in the absolute position on the constellation.

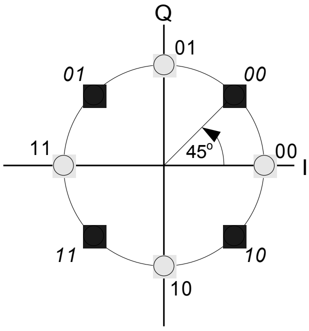


Fig. 15. DPSK constellation diagram with identical Gray coding rotated by 45°

Table 1 summarizes a possible set of relationships between the phase transitions in the DQPSK modulation scheme and the incoming Gray code dibits[7].

As mentioned above, DQPSK used root Nyquist filter or root raised cosine filter as a pulse-shaping filter. This filter is used to reduces a number of spurious signals or to control the shape of digital data.

Table 1. Correspondence between input dibit and phase change for DQPSK modulation

Gray-Encoded Input Dibit	*Phase change, $\Delta\theta$ (radians)*
00	$\pi/4$
01	$3\pi/4$
11	$-3\pi/4$
10	$-\pi/4$

The time response of Nyquist filter is determined by equation[8]:

$$g(t) = \frac{1}{\pi t}\frac{1}{1-\left(\frac{4\alpha t}{T_b}\right)^2}\sin\left\{2\pi(1-\alpha)\frac{1}{T_b}\right\} + \frac{1}{\pi}\frac{4\alpha/T_b}{1-\left(\frac{4\alpha t}{T_b}\right)^2}\cos\left\{2\pi(1+\alpha)\frac{1}{T_b}\right\} \qquad (4)$$

where α is roll-off factor of root raised cosine filter. This parameter controls the shape and bandwidth of the incoming signal.

Fig. 16 shows the time respon Nyquist filter with several chosen of roll-off factor (alpha) values. One features of the Nyquist filter is that we always obtain 0 at nT_b (*n* is integer: 1, 2, 3, ...) in the time domain when alpha = 0. Therefore, when we set the synchronization point at nT_b, one symbol never interferes with other symbols at this point.

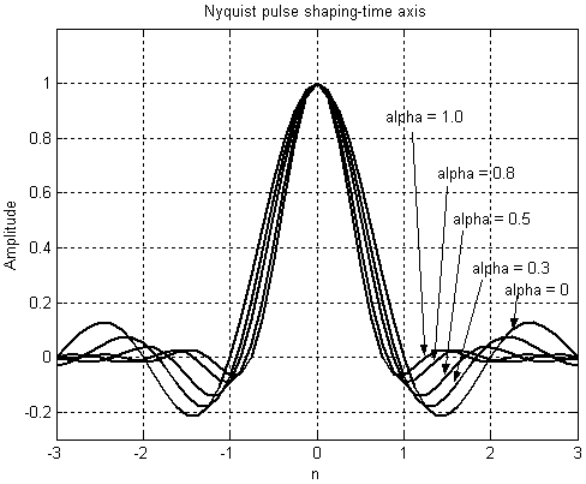


Fig. 16. Time response of Nyquist filter with several chosen roll-off factors.

The advantages of DQPSK are:

- the phase transitions from one symbol to the next are restricted to $\pm\pi/4$ and $\pm 3\pi/4$ so it will reduce the symbol ambiguity
- the signal trajectory does not pass through the origin, thus simplifying transmitter design
- using root raised cosine filtering, it has better spectral efficiency than *Gaussian Minimum Shift Keying* (GMSK), the other common cellular modulation type.

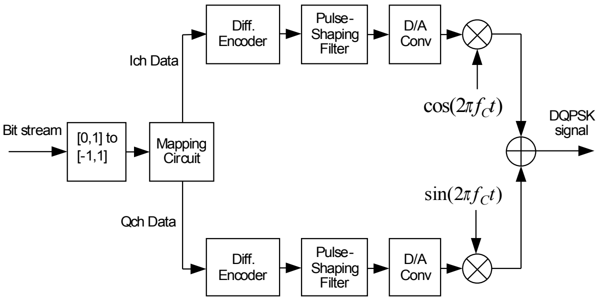


Fig. 17. DQPSK Modulator

Fig. 17 shows the principle of a DQPSK modulator. The process occurred on modulator starts by changing bit stream with level 0 and 1 into bipolar stream bit with level -1 and 1. Next, bipolar stream bit is processed by mapping circuit to produce dibits for *I-channel* (Ich) and *Q-channel* (Qch), and then each dibit data will be processed by differential encoder using the rules of Table 1 separately. The output of differential encoder will be passed through a pulse-shaping filter to reduce its spurious signals. Finally, the output of pulse-shaping filter is converted to analog signal and modulated using RF signal $\cos(2\pi f_C t)$ for Ich signal and $\sin(2\pi f_C t)$ for Qch signal, where f_C is carrier frequency. Both Ich and Qch signal are combine together to produce DQPSK signal.

At receiver or demodulator of DQPSK, process that occurred is reverse at modulator of DQPSK. Fig 18 showing block diagram of DQPSK demodulator.

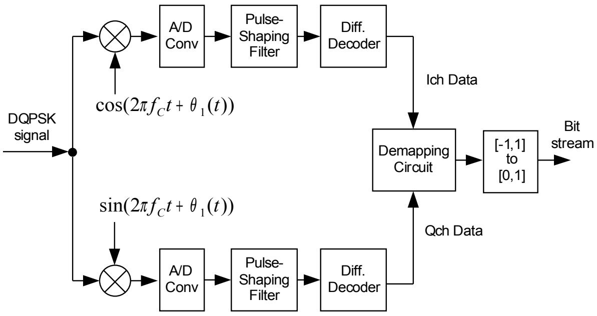
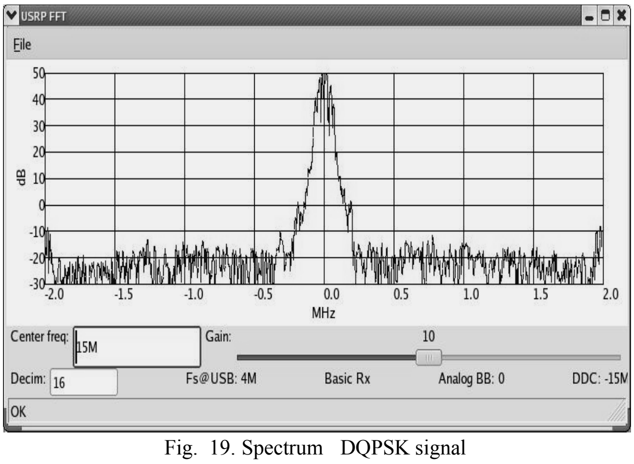


Fig. 18. DQPSK Demodulator



Center freq: 15M

Gain: 10

Decim: 16

Fs@USB: 4M

Basic Rx

Analog BB: 0

DDC: -15M

OK

Fig. 19. Spectrum DQPSK signal

Referred to the figure, originally demodulator of DQPSK will eliminate carrier frequency so that recover Ich and Qch signal. Both signal then converted in the form of digital signal through A/D Converter. Digital signal which yielded then passes through a pulse-shaping filter, and by using differential decoder dibit information transmitted by modulator will be recovered. Dibit information will be dissociated and become consecutive bit through demapping circuit, and bit stream with level -1 and 1 will be returned to bit stream with level 0 and 1. Thus, consecutive information bit data transmitted by modulator of DQPSK can be found again at DQPSK demodulator.

Fig. 19 shows the DQPSK spectrum. The results of DQPSK performance measurement are depicted in Fig. 20 up to Fig. 25.

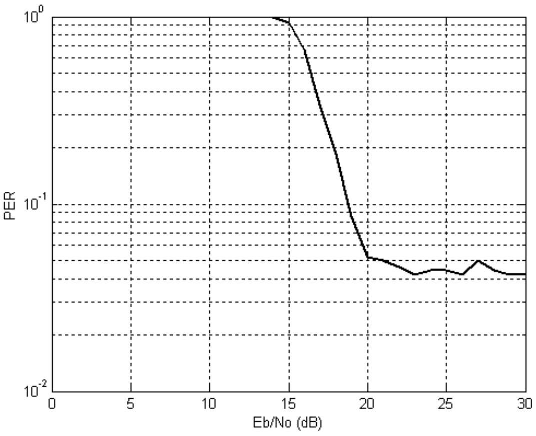


Fig. 20. The effect of Eb/No on PER

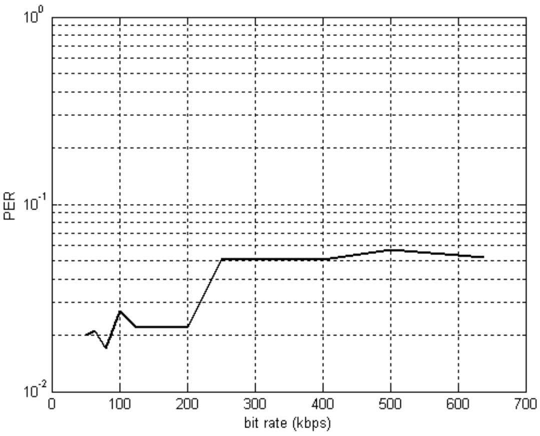


Fig. 21. The effect of bit rate on PER

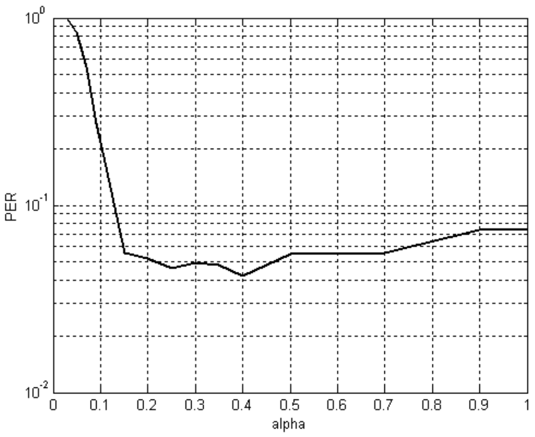


Fig. 22. The effect of roll-off factor on PER

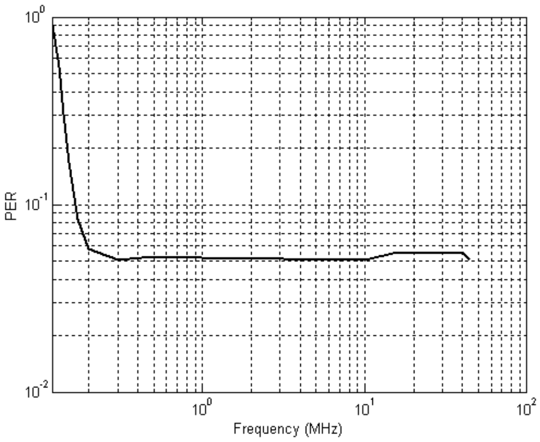


Fig. 23. The effect of carrier frequency on PER

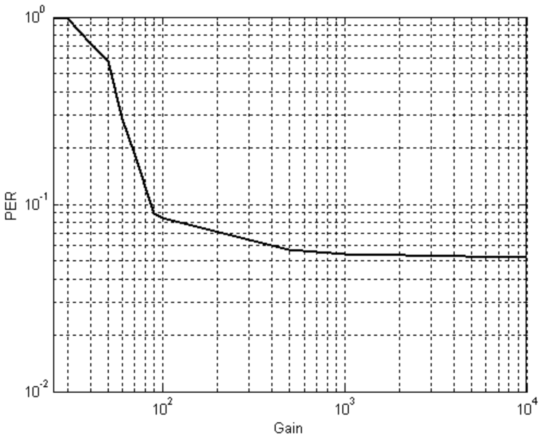


Fig. 24. The effect of gain PER

Based on the DQPSK performance, we conclude that lowest PER can be achieved by setting Eb/No larger than 20 dB, bit rate between 100 kbps up to 200 kbps, roll off factor equal 0.4, carrier frequency between 0.2 MHz up to 0.2 MHz, gain larger than 500, and payload size larger than 1024 bytes.

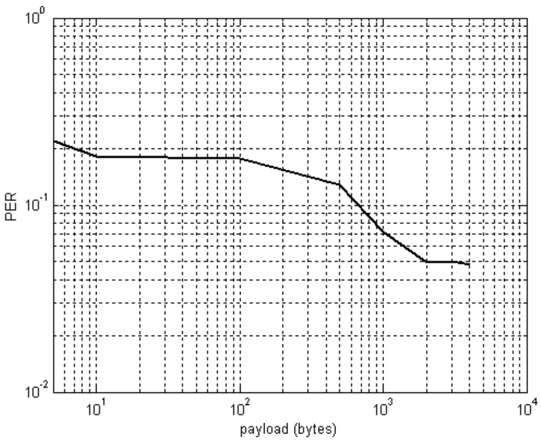


Fig. 25. The effect of payload size on PER

Referred to all the graphs of these GMSK and DQPSK performance result, PER value which obtained with variation of Eb/No, carrier frequency, bit rate, filter coefficient, gain, and payload is very low that is closing to 10^{-2}. It is seem that the lowest PER value of GMSK is lower than the lowest PER value of DQPSK. In spite of comparison with criteria measurement performance where PER < 10^{-2} not yet fulfilled. Hence, modulation scheme implementation of GMSK and DQPSK at SDR platform using USRP of GNU Radio still require to be completed either software and hardware so that we can get better performance of DQPSK modulation scheme on SDR platform using USRP GNU Radio.

V. Conclusion

Referred to this research result we can conclude that: In general can be said that modulation scheme of GMSK and DQPSK which implemented at SDR platform using USRP from GNU Radio have better performance which is seen from PER value. Lowest PER value can be obtained at Eb/No value > 20 dB for both GMSK and DQPSK modulations, optimum bit-rate 256 kbps for GMSK modulation and 200 kbps for DQPSK modulation, carrier frequency higher than 0.1 MHz for GMSK modulation and higher than 0,3 MHz for DQPSK modulation, BT of Gaussian filter larger than 0.25 for GMSK modulation, and roll-off factor (alpha) for DQPSK modulation equal to 0.4, optimum gain is 400 for GMSK modulation and 500 for DQPSK modulation, payload size of GMSK modulation larger than 1024 byte, and optimum payload size of DQPSK modulation is 4092 byte.

Continuation of this research is software development of SDR to improve the performance and also create application that adept to integration with communications protocol for example TCP/IP.

VI. Acknowledgment

We would like to thank to the Dean of STEI ITB and the chief of PPTIK STEI ITB who have given opportunity also donation in order to accomplish this research. This paper is a part of the current research which has been conducting about SDR platform development, and it is using ITB Research fund year 2007 No: 0047/K01.03.2/PL2.1.5/I/2007. We also thank to all people who contribute to this research.

References

[1] White Paper, "Software-Defined Radio." Available at http://www.wipro.com/pdf_files/sdr_wipro.pdf, Agustus, 2002.

[2] J. H. Reed, *Software Radio: A Modern Approach to Radio Engineering*. New Jersey: Prentice Hall, 2002.

[3] J. Mitola III, *Software Radio Architecture. Object-Oriented Approaches to Wireless Systems Engineering*. Canada: John Eiley & Sons, Inc, 2000.

[4] M. E. Angoletta. (2003, Feb. 20). From Analog to Digital Domain. [Online]. Available: http://humanresources.web.cern.ch/ humanresources/external/training/special/ DISP2003/DISP-2003_L01A_20Feb 03.pdf (20 Feb 2003).

[5] Srikanteswara, S., Reed, J. H., Athanas, P., Boyle, R., "A Soft Radio Architecture for Reconfigurable Platforms", IEEE Communications Magazines. February 2000.

[6] Z. Chen., "Performance analysis of channel estimation and adaptive equalization in slow fading channel," [On-line], Available at http://www.wu.ece.ufl.edu/projects/channelEstimation/full_files.zip.

[7] S. Haykin, *Communication System*, 4th ed. New York: John Wiley & Sons, Inc, pp. 362-365.

[8] H. Harada, R. Prasad, *Simulation and Software Radio for Mobile Communications*. Boston: Artech House, 2002, pp. 75-76.