A PLATFORM FOR DEVELOPING AND TESTING A POWER QUALITY METER

Ms.Theingi Zin

A Thesis Submitted in Partial Fulfillment of the Requirements
for the Degree of Master of Engineering Program in Electrical Engineering
Department of Electrical Engineering
Faculty of Engineering
Chulalongkorn University
Academic Year 2011
Copyright of Chulalongkorn University

แพลตฟอร์มสาหรับพัฒนาและทดสอบมิเตอร์คุณภาพไฟฟ้า

นางสาวเต็งกิ ซีน

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชาวิศวกรรมไฟฟ้า ภาควิชาวิศวกรรมไฟฟ้า
คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2554

Thesis Title	A PLATFORM FOR DEVELOPING AND TESTING A POWER QUALITY METER
By	Ms.Theingi Zin
Field of Study	Electrical Engineering
Thesis Advisor	Associate Professor Ekachai Leelarasmee, Ph.D.

Accepted by the Faculty of Engineering, Chulalongkorn University in Partial Fulfillment of the Requirements for the Master's Degree

………………………………… Dean of the Faculty of Engineering
(Associate Professor Boonsom Lerdhirunwong, Ph.D.)

THESIS COMMITTEE

………………………………… Chairman
(Assistant Professor Wanchalerm Pora, Ph.D.)

………………………………… Thesis Advisor
(Associate Professor Ekachai Leelarasmee, Ph.D.)

………………………………… External Examiner
(Nuntachai Poobuapheum, Ph.D.)

นางสาวเต็งกิ ซีน: แพลตฟอร์มสำหรับพัฒนาและทดสอบมิเตอร์คุณภาพไฟฟ้า (A Platform for Developing and Testing a Power Quality Meter) อ. ที่ปรึกษาวิทยานิพนธ์หลัก: รศ. ดร. เอกชัย ลีลารัศมี, 65 หน้า.

การเติบโตอย่างรวดเร็วของการใช้อุปกรณ์ที่รบกวนคุณภาพไฟฟ้าทำให้เกิดปัญหากับเครื่องใช้ไฟฟ้าตามบ้านเรือน และภาคอุตสาหกรรม นักวิจัย และนักออกแบบต่างพยายามพัฒนากรรมวิธีและขั้นตอนต่างๆ เพื่อแก้ปัญหาคุณภาพไฟฟ้าเหล่านี้ วิทยานิพนธ์นี้จะกล่าวถึงแพลตฟอร์ม 2 แบบ สำหรับประยุกต์เพื่อพัฒนามิเตอร์วัดคุณภาพไฟฟ้าซึ่งได้แก่ แพลตฟอร์มคุณภาพไฟฟ้าและแพลตฟอร์มสัญญาณรูปร่างใดๆ แพลตฟอร์มคุณภาพไฟฟ้าจะใช้แผง TMS320C5515 สำหรับประเมินตัวประมวลผลสัญญาณ แผงนี้จะต่อกับชิพ ADS8556 ที่มีช่อง 2 ช่อง เพื่อรับสัญญาณอนาล็อกแบบแรงดันและกระแสแล้วแปลงเป็นสัญญาณเชิงเลขความละเอียด 16 บิต ที่อัตราสุ่ม 20 KSamples/sec ผู้พัฒนาจะสามารถเขียนโปรแกรมภาษาซีบนคอมพิวเตอร์ส่วนบุคคลเพื่อสังเคราะห์ขั้นตอนวิธีต่างๆ ได้แก่ การแปลง FFT, การคำนวณค่าต่างๆ เช่น ค่าเฉลี่ยรากกำลังสองของแรงดันและกระแส ตัวประกอบกำลัง กำลังจริง กำลังปรากฏ และอื่นๆ จากนั้นจึงนำโปรแกรมไปลงแผงประมวลผลสัญญาณต่อไป

แพลตฟอร์มสัญญาณรูปร่างใดๆ ใช้แผงมอดูลจัดเก็บข้อมูล USB6216 ของบริษัท National Instrument มันสามารถสร้างสัญญาณต่างๆ ได้แก่ รูปคลื่นซายน์ แรงดันตก แรงดันชั่วขณะ และอื่นๆ ได้โดยวิธีป้อนโปรแกรมจากคอมพิวเตอร์ให้กับแผง USB6216 นี้ โครงงานนี้จึงสามารถตรวจวัดค่าตัวแปรต่างๆ ในสายส่ง เช่น พลังงานจริง พลังงานปรากฏ ค่าเฉลี่ยรากกำลังสองของแรงดันและกระแส ตัวประกอบกำลัง ความถี่ ปริมาณฮาร์มอนิกส์ในแรงดันและกระแส ตลอดจนสิ่งผิดปกติต่างๆ เช่น แรงดันตก แรงดันเกิน และไฟฟ้าดับ ยิ่งกว่านั้นแพลตฟอร์มทั้งสองแผงนี้ยังใช้ทดลองขั้นตอนวิธีอื่นๆ ที่เกี่ยวข้องกับคุณภาพไฟฟ้าได้อีกด้วย

ภาควิชา : วิศวกรรมไฟฟ้า ลายมือชื่อนิสิต ....

สาขาวิชา: วิศวกรรมไฟฟ้า ลายมือชื่อ อ.ที่ปรึกษาวิทยานิพนธ์หลัก ....

ปีการศึกษา 2554

5370594621 : MAJOR ELECTRICAL ENGINEERING

KEYWORDS :POWER QUALITY/ FFT ALGORITHM/ ARBITRARY WAVEFORM

THEINGI ZIN: A PLATFORM FOR DEVELOPING AND TESTING
A POWER QUALITY METER. ADVISOR:ASSOC.PROF.EKACHAI
LEELARASMEE, PhD 65 pp.

According to the rapid growth in the use of equipment that generates PQ disturbances, PQ problems will be faced with in not only domestic appliances but also utilities as well as industries. Many researchers or designers are trying to become more developed in solving these power quality problems by using many methods or algorithms. In this thesis, two major platforms will be applied for developing power quality meter. They are PQ platform and arbitrary platform. The PQ platform is based on DSP (Digital Signal Processing) Evaluation Board TMS320C5515. This board is connected with ADS8556chip having two analog input signals of voltage and current for ADC channels, 16 bit resolution and sampling rate of 20KSamples/sec. The designer can write a C program on a personal computer for implementing various algorithms such as calculating fast Fourier transform (FFT) , root mean square (RMS) values of voltage and current, power factor, real power, apparent power of sample voltage and current,etc and download the program to the DSP board for execution.

The arbitrary platform is based on Data Acquisition Module National Instrument USB6216. Various waveforms such as pure sinusoidal wave, voltage sag and transients, and other distorted waveforms can be programmed on the computer and download to the DAQ board in real time. This project can monitor power line measurement parameters such as active and reactive energy, RMS values of voltage and current, power factor, line frequency, harmonics contents of voltage and current, and abnormalities e.g.sag, swell, flicker, spike and blackout. Moreover the new methods or algorithms may be implemented on the platforms to improve the quality of power.

Department : Electrical Engineering Student's Signature..................................
Field of Study: Electrical Engineering Advisor's Signature.................................
Academic Year: ………2011…………….

Acknowledgements

The work contained in this thesis represents the accumulation of two years of my work in Thailand. Firstly , I would like to express my deep gratitude to my advisor Dr. Ekachai Leelarasmee. It was a great privilege and honor to work and study under his guidance. I would like to thank him for his supports, friendship, empathy and great vision throughout the course and during the thesis period.

In addition, I would like to thank the thesis committee members for their valuable comments and recommendations.

My true gratitude is given to my scholarship donor, AUN/SEED-Net for their financial support. This research and studying at Chulalongkorn University would not be completed without this fellowship.

And I would also like to thank my teacher Associate Professor Dr.Hla Myo Tun from Mandalay Technological University, Myanmar for his valuable guidance and supports.

I am extremely grateful to my parents, U Ye Nyunt and Daw Khin Mar Kyi, for their simultaneous supports, encouragements and educating and preparing me for my future life since I was a kid.

And I would also like to thank all of my friends from my lab for their supports during my studying in Thailand. This work is dedicated to them.

CONTENTS

Page

Chapter

Chapter **Page**

List of Tables

List of Figures

CHAPTER I

INTRODUCTION

Power Quality is a term used for describing whether the power line waveform is good or not deviates from the ideal single-frequency sine wave of constant amplitude and frequency. Nowadays many electronic devices may be developed day by day. The more increased in electronic equipments, the more power disturbances are faced with in large industries as well as utilities. The measure of power quality depends upon the needs of the equipment that is being supplied. The increases in using power electronic devices (such as inverters) and installing renewable energy power plant (such as wind and solar) have significantly deteriorated the power quality (PQ) [1] of a power distribution line because they usually inject harmonics, high switching frequencies and varying power output.

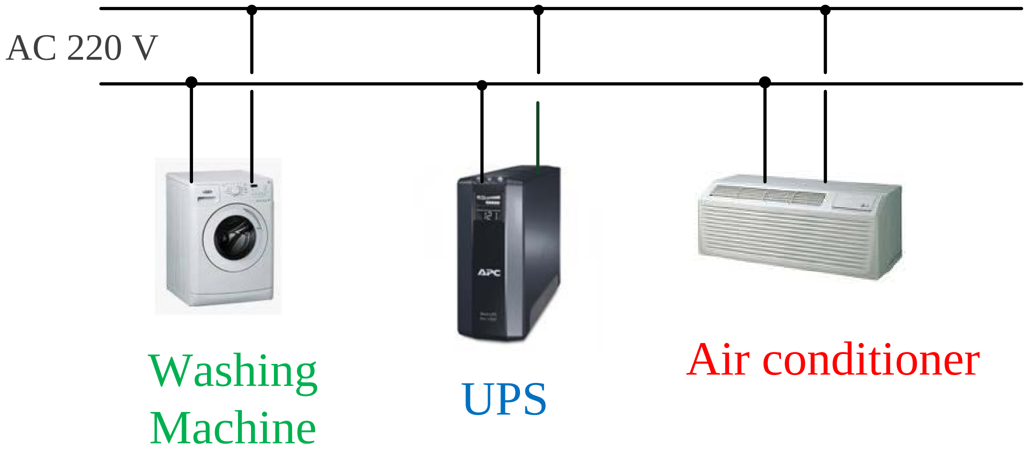


Figure.1-1. Power quality problem due to domestic appliance

PQ problems affect both consumers and electric utilities because many electronic devices are very sensitive to power system disturbances. Moreover most of modern equipments have become the major sources of the degradation of PQ. For example, many distortions may be produced by power converters and variable speed drives in the supply current and voltage waveform. These distortions may propagate all over the electrical network. This situation arises in every place including consumer houses as in Figure.1-1 where domestic equipments such as washing machine, uninterruptible power supply (UPS), air conditioning, etc. are used.

Another causes of quality of power supply depend on a number of external factors such as lighting, large load changes in industrial premises and grounding or short circuits caused by animals, branches, vehicular impact and human accidents involving electric lines.

1.1. Objectives

To develop two platforms used for

- testing various algorithms implemented in a Power Quality meter
- generating various conditions for testing a Power Quality meter

1.2. Scope of the Research

This dissertation focuses on the two platforms to implement and test a power quality meter applied in utilities as well as industries.

1.3. Expected Outcome

- Can be easily chosen which is better for customers in comparison with the performance of different PQ meters from different industries.

1.4. Research Procedure

- Review relevant literature about FFT algorithms, especially based on DSP processor
- Programming for simulation
- Propose a FFT algorithm
- Simulation to confirm the proposed idea

1.5. Outline of Thesis

This thesis is organized as follows: The power quality disturbances and monitoring are reviewed in the Chapter 2. In Chapter 3, the literature review of FFT algorithms and the manual calculation of RMS value is described. Methodology is explained in the Chapter 4. The hardware description of the platform is given in Chapter 5. The verifications and testing are given in Chapter 6, followed by the conclusions in the Chapter 7.

CHAPTER II

POWER QUALITY DISTURBANCES AND MONITORING

2.1. Typical Power Quality Problems

According to the increased in the power electronic equipments, many designers or manufacturers face with power quality problems in utilities as well as industries nowadays. Power quality problems affect many electronic equipments. There are several common parameters that characterize power quality problems in power line system. They are

(1) Surges and spikes
(2) Harmonics (current& voltage)
(3) RMS Voltage Fluctuations
(4) Transient voltages
(5) Blackouts
(6) Electrical Noise
(7) Sags (under powering) and Swell (over powering)
(8) Outage

Some of power quality problems are described in details in the next section.

2.1.1 Over voltage/ Under voltage

Occasionally over or under voltages may be supplied due to abnormal operating conditions or events. They last for more than a few seconds. They are caused by circuit overloads, poor voltage regulation, and intentional reductions by the utility. Long term over or under voltage may affect sensitive electronic equipment. Fig.2-1. shows waveform of over voltage/ under voltage.

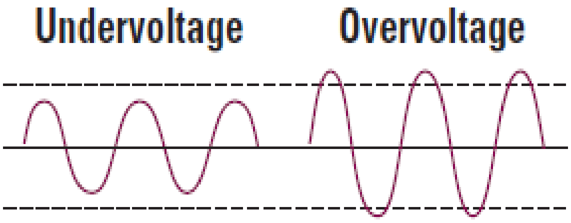


Fig.2-1.Sample circuit waveform of Over voltage/ Under voltage

2.1.2 Voltage Sags (Under powering) and Voltage swell (Over Powering)

Voltage sag is a momentary decrease in the voltage outside the normal tolerance. When an electrical fault occurs either within our facility or on our system, the voltage drop depends on the distance between the equipment and the location of the fault. If the distance of the equipment is close to the fault, the voltage drop that will be seen at the equipment terminals will be great. The duration of the fault is determined by the speed of the circuit protection device (fuse or circuit breaker) that detects the fault. If a voltage sag occurs in power distribution line, sensitive electronic equipment may be affected. Voltage sags are often caused when heavy loads are started, by lighting and by power system faults.

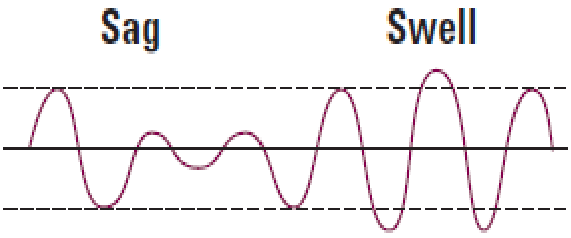


Fig.2-2. Sample waveform of Voltage Sag and Voltage Swell

A voltage swell is a momentary increase in voltage. Swells may be caused by removing a large load from or by adding capacitance to a circuit. Swells may also occur when a fault on one phase results in a temporary voltage rise on the unfaulted phases. Swells may result in voltages that exceed electronic device ratings. Voltage swells are often caused by sudden load decreases or turn-off of heavy equipment. Repetitive swells may cause electronic device damage. The waveform representation of voltage sag and voltage swell can be illustrated in Fig.2-2.

2.1.3 Electrical Noise and Harmonic Distortion

Electrical noise is a distortion of the normal sine wave power and a high frequency interference that can be induced on electrical system by those conductors. It can be caused by radar and radio transmitter, fluorescent lights, power electronics control circuits, arcing utility and industrial equipment, loads with solid-state rectifiers and switching power supplies typically used in computer systems. Electrical noise disturbs microprocessor-based equipment, such as microcomputers and programmable controllers.

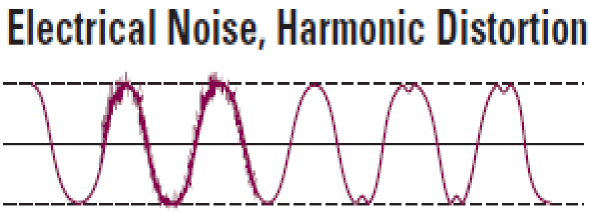


Fig.2-3. Sample waveform of Electrical Noise and Harmonic Distortion

When the 60-cycle voltage waveform is altered by higher frequencies, harmonic distortion occurs. The amount of the harmonic distortion is directly related to the harmonic content. The magnitude of the harmonic distortion depends on the amount of computer- based equipment, communication equipment and other electronic equipment in use. A large amount of this equipment in use can generate levels of harmonic distortion that can interfere with the operation of other equipment within a facility. Harmonic distortion causes motor loads, such as compressors, pumps and disk drives to overheat. Fig.2-3 illustrates the sample waveform of electrical noise and harmonic distortion.

2.1.4 Spikes (Impulses)

Spikes (impulses) are very short in duration voltage increases. They occur on the electrical transmission and distribution system as a result of normal operation of fault clearing devices, capacitor switching, or even normal switching of loads by the utility or its customers. Since they can damage some electronic components, some manufacturers build impulse protection into their products. The protection typically consists of a device that safely diverts the incoming impulse into the grounding system before it enters the equipment. Spikes can destroy electronic loads and breakdown transformer or motor insulation. Fig.2-4. shows the illustration of voltage spikes waveform.

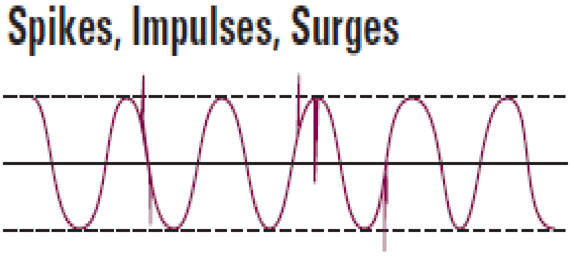


Fig.2-4. Illustration of voltage spikes waveform

2.1.5 *Outage*

An outage is a complete loss of power that may last from several milliseconds to several hours and may be caused by power system faults, accident involving power lines, transformer failures and generator failures. Some sensitive equipment may be disrupted by outages as short as 15 milliseconds. All equipment may be affected by outages. The sample waveform of outage can be shown in Fig.2-5.

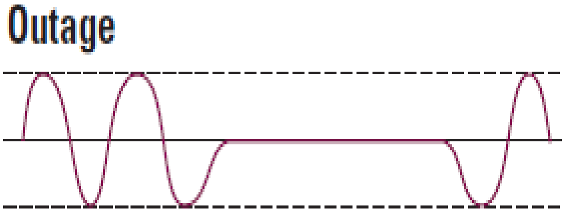


Fig.2-5. Sample waveform of outage

2.2. Power Quality Monitoring

It is therefore important to monitor PQ in real time because this can help the utilities as well as industries in order to detect and correct power quality problem quickly. The equipment for this purpose is called a PQ meter. It can measure several parameters in real time situation. These are described in the following.

(1) Active and reactive energy/ power usages

(2) RMS values of voltage/ current and its power factor

(3) Line frequency

(4) Harmonic contents of voltage/current

(5) Abnormalities (sag, swell, flicker, spike, blackout)

CHAPTER III

LITERATURE REVIEW

There are many algorithms to implement a power quality meter in real time situation. To implement the various algorithms, such as calculating FFT (fast Fourier transform) and RMS values on a personal computer, the manual calculation of FFT and RMS will be needed to compute. Therefore it will be explained how to evaluate these values to obtain the desired frequency spectra.

3.1. Discrete Fourier Transform (DFT)

The discrete Fourier transform (DFT) is the conversion of time- domain signal into equivalent frequency- domain signal. The DFT produces a set of coefficients, which are sampled values of the frequency spectrum at regular intervals, just like the discrete Fourier series. In this algorithm, the number of samples obtained depends on the number of samples in the time sequence. The general formula of the discrete Fourier transform is shown in equation (3.1).

$$X(k) = \sum_{n=0}^{N-1} x(n) e^{-j2\pi kn/N} \quad , \quad k = 0,1,\ldots,N-1 \qquad (3.1)$$

where n is the discrete- time index and N is the frame length or number of sample points. The output sequence $X(k)$ is the sampled value of the continuous frequency spectrum of $x(n)$. In this equation, W_N^{kn} can be substituted instead of $e^{-j2\pi kn/N}$ .Therefore the sequence $X(k)$ can be rewritten as:

$$X(k) = \sum_{n=0}^{N-1} x(n) W_N^{kn} \quad , \; k = 0,1,\ldots,N-1 \qquad (3.2)$$

the parameter W_N^{kn} is called the twiddle factors or constants of the DFT. It is also an important value in calculating the Fourier coefficients.

3.2. Fast Fourier Transform (FFT)

The fast Fourier transform (FFT) [2] is an efficient algorithm that is used for converting a discrete time-domain signal into an equivalent discrete frequency-domain signal based on the discrete Fourier transform (DFT). Nowadays many designers utilize this algorithm to implement the various electronic equipments because of its much more rapid computation. It is a faster version of the discrete Fourier transform. The computation of the sequence $X(k)$ is dependent on the value of number of samples N and the number of periods of the signal . In theoretically, for computing N samples of $X(k)$ for $k = 0,1,...,N-1$, N^2 complex multiplications and $\left(N^2 - N\right)$ complex additions are needed. The number of complex multiplication can be calculated by $N/2*\log_2 N$ and the number of complex addition can be calculated by the formula of $N*\log_2 N$. Therefore the number of complex multiplication and addition may be changed depending on the value of N.

The basic idea of the FFT is to separate the DFT equation into two parts such as even part and odd part as in given equation (3.3).

$$X(k) = \sum_{n=0}^{N/2-1} x(2n)W_N^{2nk} + W^k \sum_{n=0}^{N/2-1} x(2n+1)W_N^{2nk}, \quad k = 0,1,...,N-1 \tag{3.3}$$

$$= X_1(k) + W_N^k X_2(k)$$

where $X_1(k) = \sum_{n=0}^{N/2-1} x(2n)W_N^{2nk}$ and $X_2(k) = \sum_{n-0}^{N/2-1} x(2n+1)W_N^{2nk}$. The first part is the DFT of the even sequence and the second part $X_2(k)$ is the DFT of the odd sequence. The complex factor W_N^k is known as the twiddle factor. These subsequences can be subdivided into even and odd sequences until only 2-point DFTs are left. So each $N/2$-point DFT is obtained by combining two $N/4$-point DFTs, each of that is obtained by combining two $N/8$-point DFTs, etc. There are two classes of FFT algorithms: decimation-in-time (DIT) and decimation-in-frequency (DIF).

3.2.1. Decimation-in-Time (DIT)

The decimation-in-time (DIT) [2] is the process that decomposes the input sequence into smaller subsequences until the simplest computation remains. In the decimation-in-time algorithm, the N -samples sequence is first divided into two sequences: the even numbered sequence and the odd numbered sequence. The DFT can be divided into two DFTs of length $N/2$. So the equation can be rewritten as in the following:

$$X(k) = \sum_{n=0}^{N-1} x(n) W_N^{kn} \tag{3.4}$$

$$= \sum_{m=0}^{(N/2)-1} x(2m) W_N^{2mk} + \sum_{m=0}^{(N/2)-1} x(2m+1) W_N^{(2m+1)k}$$

The first part is the even part of the sequence and the second part is the odd part of the sequence. In the above equation, the twiddle factor W_N^{2nk} is equal to :

$$W_N^{2mk} = e^{-j\frac{2\pi}{N}2mk} = e^{-j\frac{2\pi}{N/2}mk} = W_{N/2}^{mk} \tag{3.5}$$

The reformulated equation of the output sequence $X(k)$ can be described as follow.

$$X(k) = \sum_{m=0}^{(N/2)-1} x_1(m) W_{N/2}^{mk} + W_N^k \sum_{m=0}^{(N/2)-1} x_2(m) W_{N/2}^{mk} \tag{3.6}$$

where $x_1(m) = x(2m)$ and $x_2(m) = (2m+1)$. According to the above formula, each of the summation terms is reduced to a $N/2$ two point DFT. The two point DFT is called a butterfly, and it is the simplest computational kernel of Radix-2 FFT algorithms. Each butterfly consists of a single complex multiplication by a twiddle factor W_N^k , one addition and one subtraction. The butterfly representation for the decimation-in-time FFT algorithm can be illustrated in Figure.3-1.

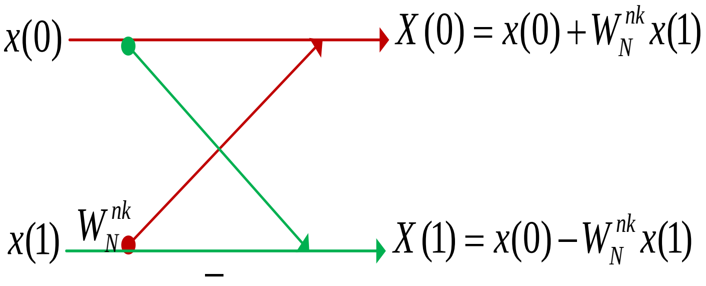


Figure.3-1. Flow graph for a butterfly computation

In decimation-in-time algorithm, the output sequence is in sequential order, while the input sequence has the unusual order. Before the calculation, the order of the input sequence is actually arranged as if each index was written in binary form and then the order of binary digits was reversed. Each of the time sample indices in decimal is needed to convert to its binary representation. The binary bit streams are then reversed. Converting the reversed binary numbers to decimal values gives the reordered time indices. If the input is in natural order, the output will be in bit-reversed order. Most modern DSP chips such as the TMS32055xx provide the bit-reversal addressing mode to support this bit-reversal process. Therefore the input sequence can be stored with the bit-reversed addresses computed by the hardware. Figure.3-2 shows the flow graph which illustrates the 8 points DFT.

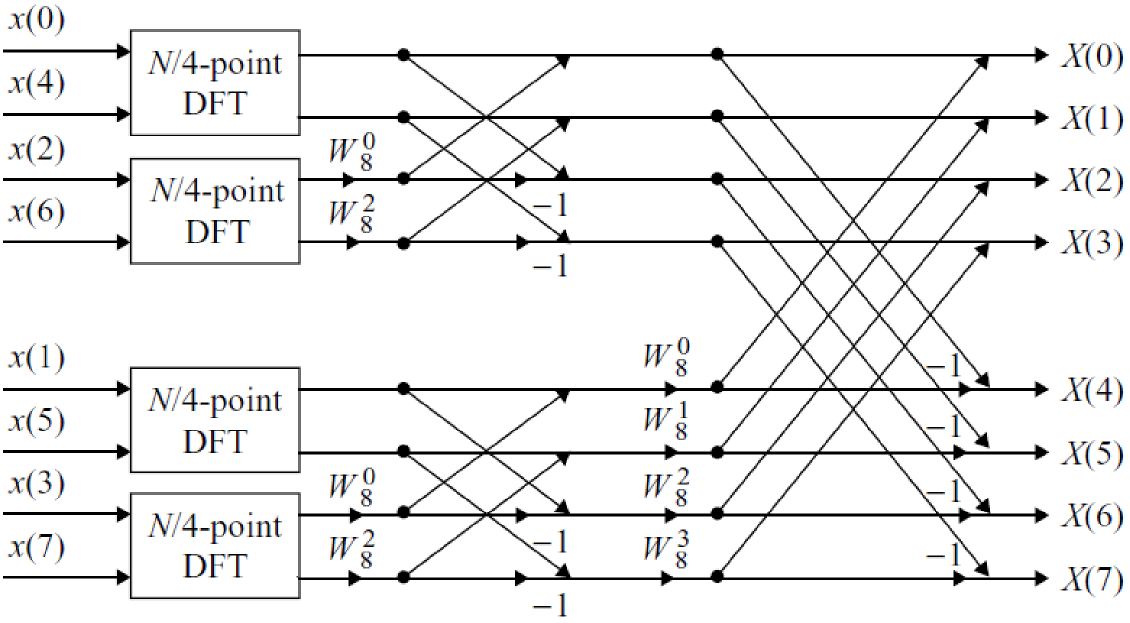


Figure.3-2. Flow graph illustrating of N-point DFT, N=8

3.2.2. Decimation-in-Frequency (DIF)

Another algorithm of fast Fourier transform is the decimation-in-frequency (DIF) [2]. The development of the decimation-in-frequency FFT algorithm is similar to the decimation-in-time algorithm. In this algorithm, it consists of two steps. The first step is to divide the data sequence into two halves, each of $N/2$ samples. Then the output sequence $X(k)$ can be expressed as the sum of two components as described in equation (3.7):

$$X(k) = \sum_{n=0}^{(N/2)-1} x(n)W_N^{nk} + \sum_{n=N/2}^{N-1} x(n)W_N^{nk} \qquad (3.7)$$

$$= \sum_{n=0}^{(N/2)-1} x(n)W_N^{nk} + W_N^{(N/2)k} \sum_{n=0}^{(N/2)-1} x\left(n+\frac{N}{2}\right)W_N^{nk}$$

The next step is to separate the frequency terms $X(k)$ into even and odd samples of k .Therefore the output samples can be described as the following equations:

$$X(2k)=\sum_{n=0}^{(N/2)-1}\left[x(n)+x\left(n+\frac{N}{2}\right)\right]W_{N/2}^{kn} \tag{3.8}$$

$$X(2k+1)=\sum_{n=0}^{(N/2)-1}\left[x(n)-x\left(n+\frac{N}{2}\right)\right]W_{N}^{k}W_{N/2}^{kn} \tag{3.9}$$

where $k,n=0,1,2,...,N-1$. The process of decomposition continues until the last stage consists of 2-point DFTs. The decomposition and symmetry relationships are reversed from the decimation-in-time algorithm. In this algorithm, the bit reversal occurs at the output instead of the input in decimation-in-time algorithm. Figure.3-3. illustrates the decomposition of an N-point DFT into two $N/2$ DFTs. The output sample is equal for both algorithms. The important differences between FFT algorithms concern how data are passed to and from the subroutines.

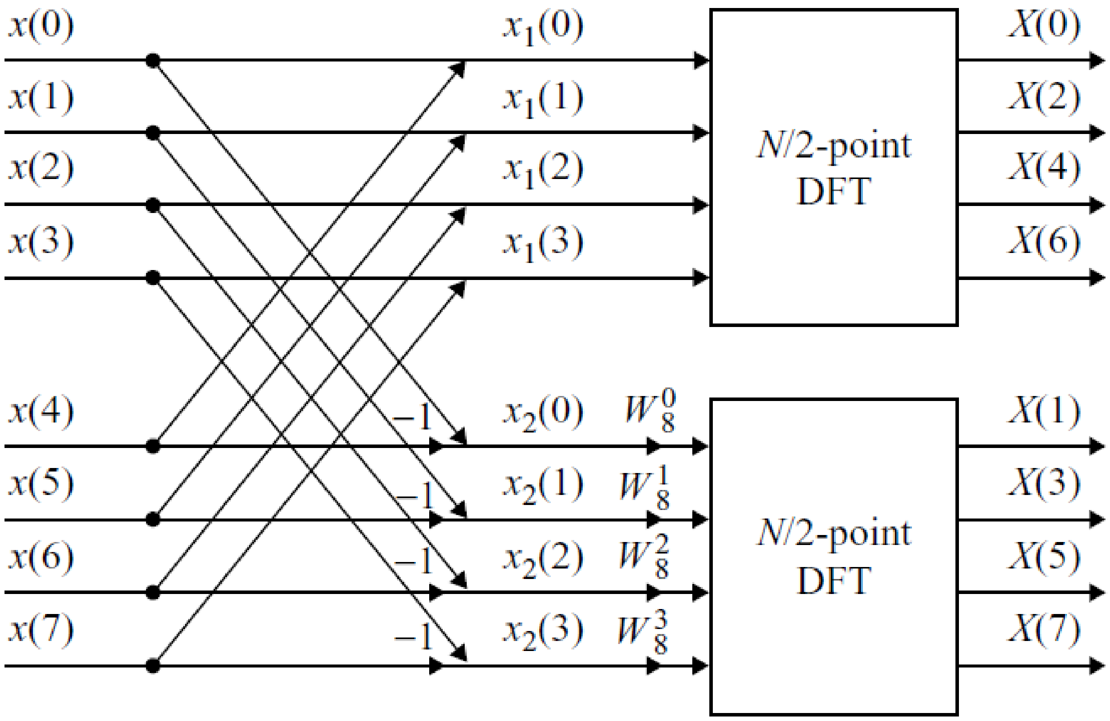


Figure.3-3. Decomposition of an N-point DFT into two $N/2$ DFTs

3.3. Calculation of Root Mean Square (RMS) Value

Root mean square (RMS) [3] is a fundamental measurement of the magnitude of an ac signal. It is the square root of the arithmetic mean of the square of the original values. Its operation can be used in various fields, including electrical engineering. It can be applied not only in continuous signal but also in discrete signal. The rms value of a continuous signal $V(t)$ is mathematically defined as

$$V_{RMS} = \sqrt{\frac{1}{T} \times \int_{0}^{T} V^{2}(t)dt} \qquad (3.10)$$

where $V(t)$ is the continuous input signal, T is the period and V_{RMS} is the rms value of the voltage. The rms value of the current can be calculated with the same formula for continuous signal. In this formula the integration sign may be used.

For discrete time signal, the rms calculation involves squaring the signal, taking the average, and obtaining the square root. It can be described in the following formula

$$V_{RMS} = \sqrt{\frac{1}{N} \times \sum_{i=1}^{N} V^{2}(i)} \qquad (3.11)$$

where $V(i)$ is the input signal, N is the number of points and V_{RMS} is the rms value of the voltage. In contrast with the continuous signal, the summation sign may be used in discrete time signal. The rms value of the current can be calculated with the same formula. Even though the calculation is a bit more complicated for power and current, it is easy to determine the value of the voltage.

CHAPTER IV

METHODOLOGY

In this chapter, the operation of the hardware devices will be explained how to implement and test power quality meter. The internal block diagram of the proposed PQ meter is shown in Fig.4-1. The voltage and current sensors scale the incoming AC voltage/ currents using a resistive divider and current transformer. An analog-to-digital converter (ADC) converts sampled V/I signals into streams of digital data. The digital signal processor (DSP) performs digital calculation (such as fast Fourier transform) to obtain the desired PQ parameters. The liquid crystal display (LCD) shows the PQ parameters, time and status.

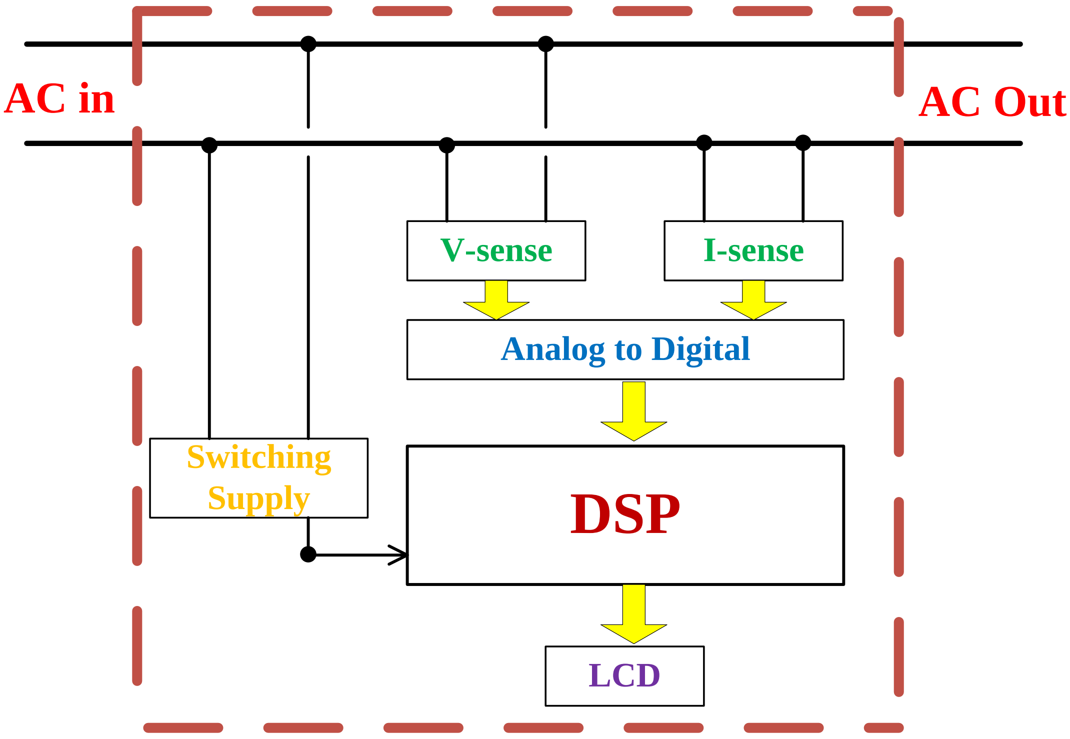


Figure.4-1.The internal blocks of a PQ meter

4.1. A DSP Based Platform for PQ Measurement

In this project, the DSP development board TMS320C5515 [5] as shown in Fig.4-2. is selected as the main platform in order to save effort time as opposed to conventional microcontroller. This board consists of several components that are suitable for implementing the PQ meter. They are

a) A low power 16-bit fixed point DSP running at 100MHz. It has an internal 128KB ROM and 192KB RAM.
b) An FFT accelerator for FFT calculation built-in the DSP chip.
c) 128MB RAM, 16MB flash and 256KB EEPROM on board memory.
d) 128 x 128 map color LCD display
e) USB 2.0, RS232, SPI (Serial Peripheral Interface)
f) JTAG for program downloading and debugging
g) MMC/ SD connector for additional external storage
h) Expansion connectors for Blue Tooth interface
i) 10 User push button switches

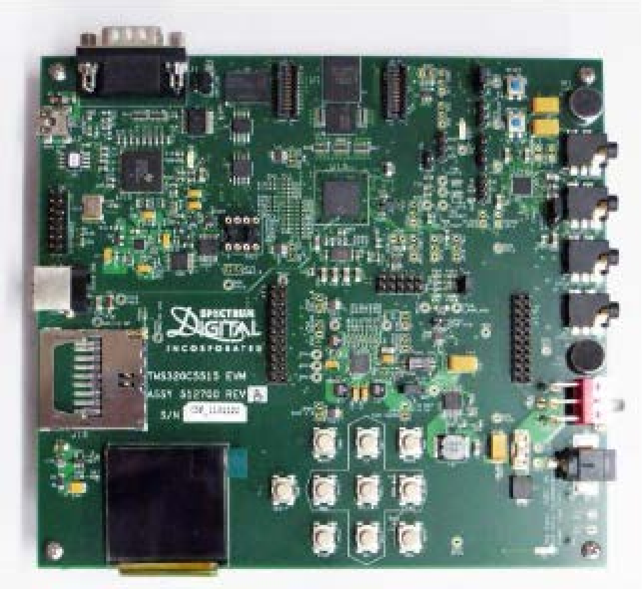


Figure.4-2. Picture of DSP board (TMS320C5515)

In this work, the proposed method is to develop FFT (fast Fourier transform) algorithm for implementing on the TMS320C5515 DSP to obtain the desired PQ parameters. It is one of the most commonly used techniques for digital signal processing because of its computational efficiency. The TMS320C5505/15 (C5505/15) DSP [7] includes an FFT hardware accelerator (HWAFFT) that is tightly coupled with the CPU (central processing unit), allowing high FFT processing performance at very low power.

4.2. A Platform for PQ Testing

As it is hard to find various conditions for testing a PQ meter in the real situation, an instrumental setup or test bench is build up in the laboratory. The test bench utilizes a signal generator for generating waveforms that simulate voltage and current for testing it under various situations such as harmonics and spike. The test bench needs an arbitrary waveform generator (AWG) with large memory for storing sampled data. When the amount of memory is large, we choose an arbitrary waveform generator for testing a PQ meter in real situation. A high speed data acquisition (DAQ) module NI USB 6216 [6] as shown in Fig.4-3 should be able to use with high speed USB connection with a personal computer in order to develop a low cost with high flexible AWG for this test.

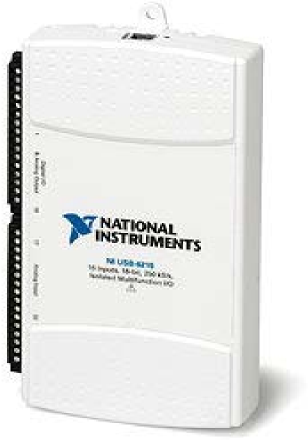


Figure.4-3. Picture of National Instrument USB 6216

The NI USB-6216 [9] is a bus-powered USB M Series multifunction data acquisition (DAQ) module optimized for superior accuracy at fast sampling rates. It offers 16 analog inputs, 400KS/s sampling rate, two analog outputs, 32 digital I/O lines, four programmable input ranges (± 0.2 to ± 10 V) per channel, digital triggering, and two counter/timers. The USB-6216 is designed specifically for mobile or space-constrained applications. This product does not require external power supply because it is only a hardware USB device to connect with a personal computer. It can also be quickly acquired, analyzed and presented data without programming because every M Series DAQ device includes a copy of NI LabVIEW Signal Express LE data-logging software.

4.3. ADC Board for Developing a Power Quality Meter

The operation of ADC board as shown in Fig.4-4. is to perform analog to digital conversion (ADC) of voltage and current waveforms obtained from the voltage and current sensors using transformers. It employs ADS 8556 chip [8] which has 6 analog inputs (of which two channels are used), a maximum sampling rate of 250 kHz and 16 bit resolution. Digitized data are multiplexed and serially read through the SPI (Serial Peripheral Interface) port of the DSP. The full sampling rate is used to capture transient waveform as short as 4uS. However, the DSP is programmed to down sample it to 10KSamples/Sec before applying 1000 points (5 cycles) FFT algorithm to yield up to 30th harmonic. The ADC board can digitized automatically the analog input signals of voltages and currents.

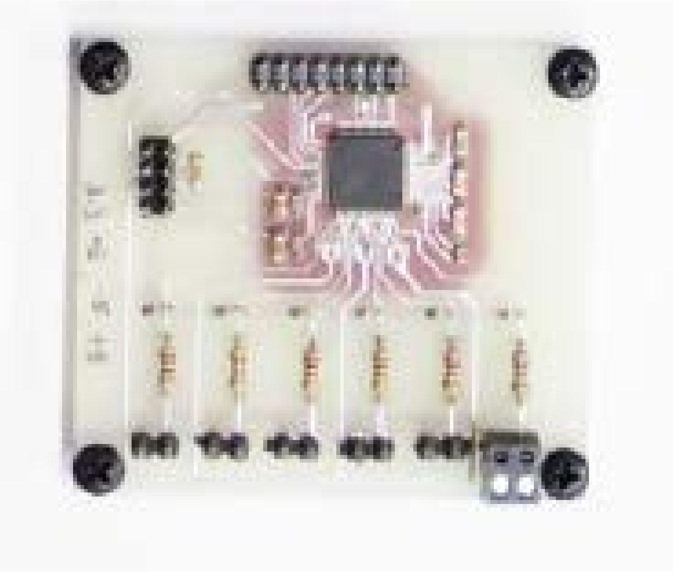


Figure.4-4 Picture of ADS 8556 chip

4.4 Power Quality Monitoring System of the Whole Process

The power quality monitoring system is built around the Texas Instruments TMS320C5515 [4] DSP evaluation board that is interface to a computer for presentation and data logging purposes. The proposed system of the whole project is depicted in Fig.4-5. This project presents a system that can monitor power-line voltage and current waveforms. In implementing a power quality meter in this project, the analog input signal of voltage and current (AC220V, 50Hz) form the voltage and current sensors using a single phase transformer is applied to the analog-to-digital converter (ADC) board. It converts the incoming analog input signal to discrete output signals. In this portion, the heart of the process is the Texas Instrument DSP TMS320C5515 evaluation board. It is used as a PQ platform to implement a power quality meter. The designer can write a C program on a personal computer to implement various algorithms such as calculating FFT and RMS values and download the program to the DSP board. Moreover the methods and algorithms applied by other researchers can be improved in this section.

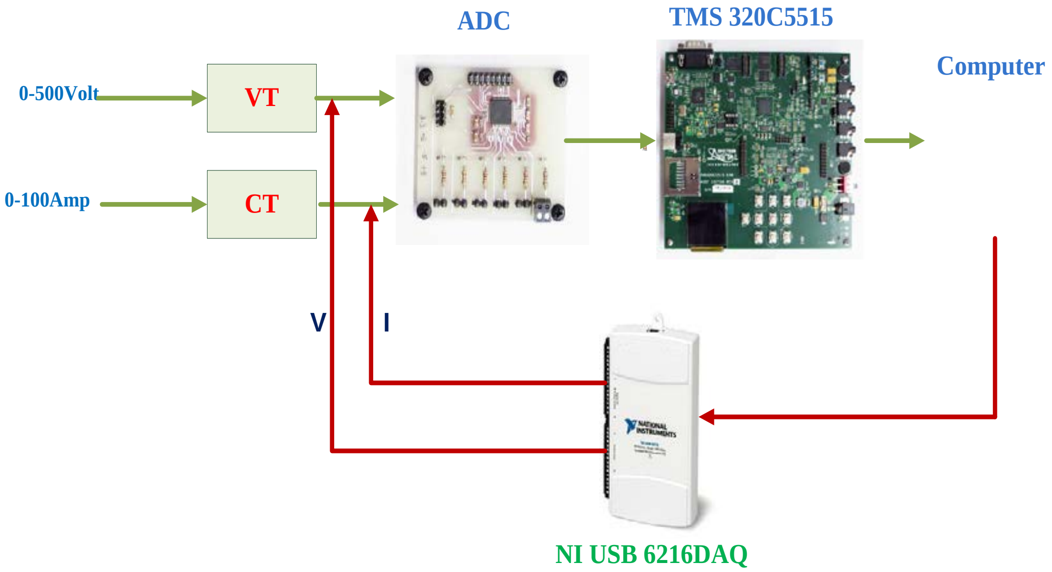


Figure.4-5. Block Diagram of the whole process

The arbitrary platform is used to test a PQ parameters based on NI DAQ module connected with a personal computer. In this portion, waveforms of arbitrary shapes such as pure sinusoidal of any frequency , harmonic of any order, voltage sag and transient can be programmed on the computer and download to the DAQ board in real time for testing a power quality meter. The resulted output power signal can be compared with the implemented value. By using these two conventional platforms, the development of power quality meter can be easily constructed and tested.

CHAPTER V

HARDWARE DESCRIPTION

In this chapter, the hardware description of the TMS320C5515 EVM used as the main platform in this project will be explained. It is a standalone development platform that enables users to evaluate and develop applications. It is widely used in many applications such as portable voice/audio devices, noise cancellation headphones, software-defined radio, musical instruments, medical monitoring devices, wireless microphones, biometrics, industrial instruments, telephony, and audio cards. Figure 5-1 shows the picture of TMS320C5515 EVM [5].

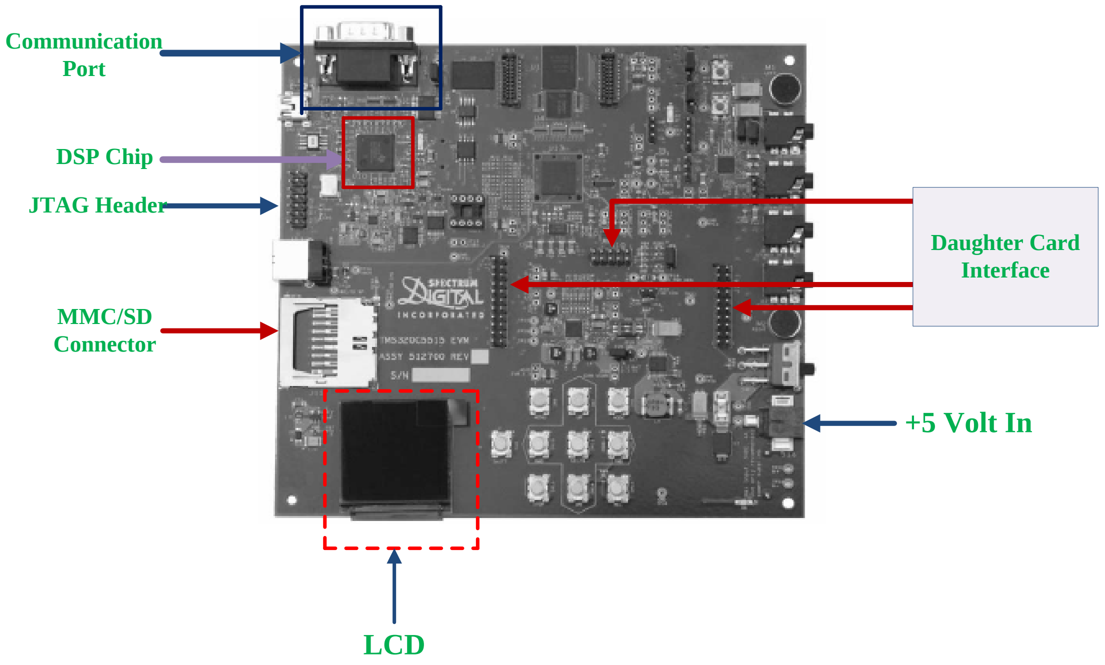


Figure. 5-1. Picture of DSP Evaluation Board

There are several key features in DSP Evaluation Board. The main components are 128x128 bit mapped color LCD display to show the PQ parameter, time and status, Joint Test Action Group (JTAG) for program downloading and debugging, MMC/SD (Multimedia Card or Secure Digital Card Connector) for additional external storage, On-Chip RAM composed of 64K Bytes of Dual-Access

RAM (DARAM) and 256K Bytes of Single-Access RAM(SARAM) , and 128K Bytes of On-Chip ROM to store data, fixed point digital signal processor for low power application with high performance and USB port for communicating with personal computer.

5.1. Functional Block Diagram of DSP Processor

The processor is a member of TI's TMS320C5000^{TM} fixed-point Digital Signal Processor (DSP) [5] product family and is designed for low-power application. The functional block diagram of DSP Processor can be illustrated in Figure 5-2. It consists of the following primary components:

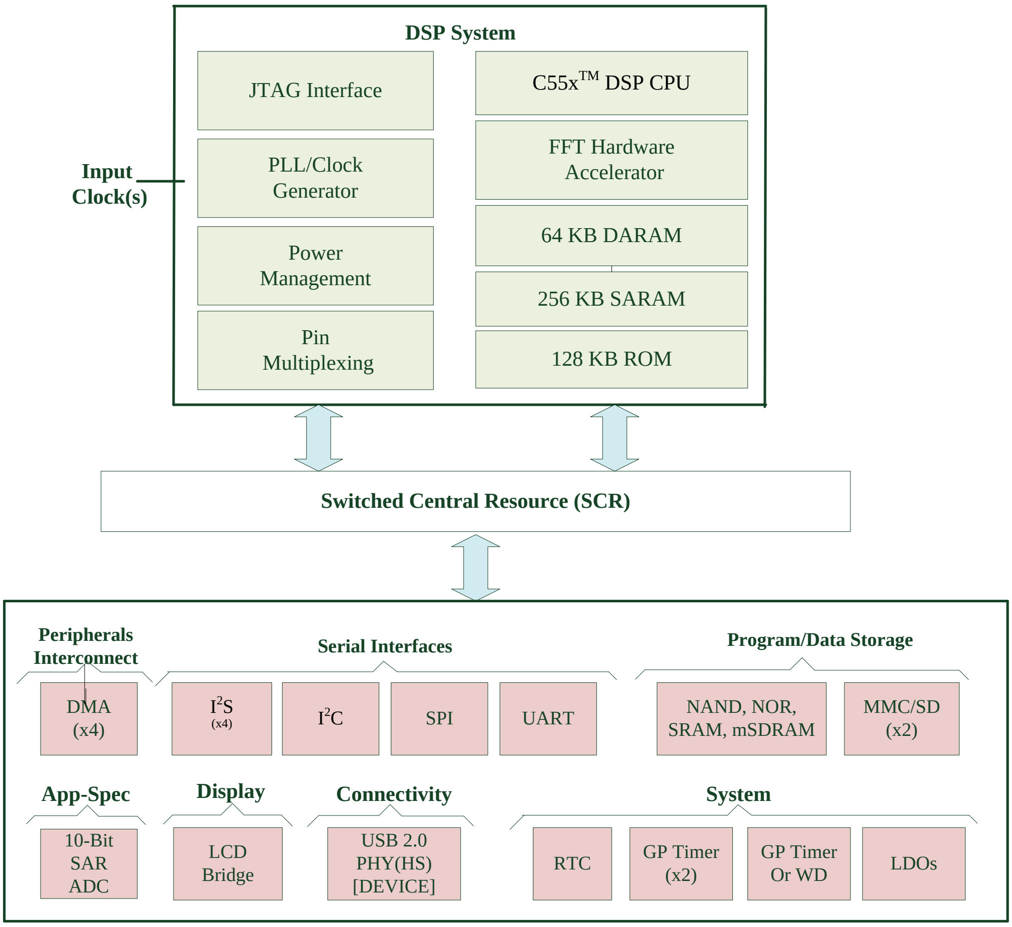


Figure.5-2.Functional Block Diagram

1) Dual MAC (multiply-accumulate), C55x CPU
2) On-Chip memory: 320KB RAM (64KB DARAM, 256 KB SARAM) , 128KB ROM
3) HWAFFT that supports 8- to 1024-point (powers of 2) real and complex-valued FFTs
4) Power management module
5) A set of I/O peripherals that includes I^2C, I^2S, SPI, UART, Timers, EMIF, MMC/SD, GPIO, 10-bit SAR, LCD Controller, USB 2.0, DMA controller
6) Three on-chip LDO Regulators
7) SDRAM/ mSDRAM support

5.1.1. C55x DSP CPU

The CPU[5] is the important part of the processor. It supports an internal bus structure that consists of one program bus, one 32-bit data read bus and two 16-bit data read buses, two 16-bit data write buses, and additional buses dedicated to peripheral and DMA activity. These buses provide the ability to perform up to four 16-bit data reads and two 16-bit data writes in a single cycle.

The C55x CPU provides two multiply-accumulate (MAC) units, each capable of 17-bit x 17-bit multiplication and the addition of 32-bit in a single cycle. A central 40-bit arithmetic/logic unit (ALU) is supported by an additional 16-bit ALU. The ALUs may be used under instruction set control, providing the ability to optimize parallel activity and power consumption. These resources are managed in the Address Unit (AU) and Data Unit (DU) of the C55x CPU.

The C55x CPU supports a variable byte width instruction set for improved code density. The Instruction Unit (IU) performs 32-bit program fetches from internal and external memory, stores them in a 128-byte Instruction Buffer Queue, and queues instructions for the Program Unit (PU). The Program Unit decodes the instructions, directs tasks to the Address Unit (AU) and Data Unit (DU) resources, and manages the fully protected pipeline.

5.1.2. FFT Hardware Accelerator

The FFT Hardware Accelerator [7] is a tightly-coupled, software-controlled coprocessor designed to perform FFT and inverse FFT (IFFT) computations on complex data vectors ranging in length from 8 to 1024 points (powers of 2). It is physically located outside of the DSP core but has access to the core's full memory read bandwidth, access to the core's internal registers and accumulators, and access to its address generation units. The HWAFFT cannot access the data write busses or memory mapped registers (MMRs). It can also generate internal twiddle factor for optimal use of memory bandwidth and is more efficient for programming.

The core of the HWAFFT consists of a single Radix-2 Butterfly implemented in hardware. It can support Single-stage and Double-stage modes to compute one or two stages in one pass. The double-stage mode is significant to speed-up for large FFT lengths. The single-stage mode is provided when the final stage is necessary to compute alone because the number of required stages is odd at a lower acceleration rate. It implements a Radix-2 DIT structure that returns the FFT or IFFT result in bit-reversed order. It is 4 to 6 times energy efficient and 2.2 to 3.8 times faster than FFT computations on the CPU.

5.1.3. Memory Allocation

The device peripheral set includes 320K Bytes Zero-Wait State On-Chip RAM that is composed of 64K Bytes of Dual-access RAM (DARAM) [10] with 8 Blocks of 4K x 16-Bit and 256K Bytes of Single-Access RAM (SARAM) with 32 Blocks of 4K x 16-Bit. It also consists of 128K Bytes of Zero Wait-State On-Chip ROM with 4 Blocks of 16K x 16-Bit. It includes an external memory interface (EMIF) that provides glueless access to asynchronous memories like EPROM, NOR, NAND, and SRAM, as well as to high-speed, high-density memories such as synchronous DRAM (SDRAM) and mobile SDRAM (mSDRAM).

5.1.4 Peripheral Interfacing

5.1.4.1. Direct Memory Access (DMA)

The TMS320C5515 Digital Signal Processor includes four Direct Memory Access (DMA) controller with four DMA channels each for a total of 16 DMA channels. This controller is used to move data among internal memory, external memory, and peripherals without intervention from the CPU and in the background of CPU operation. Each DMA controller can perform one 32-bit data transfer per cycle, in parallel and independent of the CPU activity.

5.1.4.2. Successive Approximation Register (SAR)

The successive approximation register (SAR) in the device is a 10-bit analog-to-digital converter (ADC) using a switched capacitor architecture that converts an analog input signal to a digital value at a maximum rate of 64 kilo samples per second (ksps) for use by the DSP. This SAR module provides six channels that are connected to four general-purpose analog pins that can be used as general- purpose outputs.

5.1.4.3. General-Purpose Timers (GP)

In addition the device also includes three 32-bit General-Purpose Timers. Each timer can be used as a general-purpose (GP) timer. Timer2 also contains a 16-bit Watchdog (WD) which shares the same clock gating bit as the GP but works independently. General purpose timers are typically used to provide interrupts to the CPU to schedule periodic tasks or a delayed task. The general-purpose (GP) timers are 32-bit prescalar that divides the CPU system clock and uses this scaled value as a reference clock. These timers can be used to generate periodic interrupts. Watchdog timers are used to reset the CPU in the event of a deadlocked state, such as a non-existing code loop. In contrast with 32-bit timers, watchdog timer is a 32-bit timer composed of a 16-bit counter with a 16-bit prescalar that divides the CPU system clock and uses this scaled value as a reference clock.

5.1.4.4. Real Time Clock (RTC)

The device includes a real time clock (RTC) that provides a time reference to an application executing on the DSP. The RTC has its own crystal input, clock domain, and core and I/O power supplies. The separate clock domain allows the RTC to run while the rest of the device is clock gated. The RTC has the capability to wake-up the rest of the device through an alarm interrupt, periodic interrupt, or external WAKEUP signal.

5.1.4.5. Serial Peripheral Interface (SPI)

The serial peripheral interface (SPI) in this device is a high-speed synchronous serial input/output port that allows a serial bit stream of programmed length (1 to 32 bits) to be shifted into and out of the device at a programmed bit-transfer rate. The SPI supports multi-chip operation of up to four SPI slave devices. The SPI can only operate in master mode only. The SPI is normally used for communication between the DSP and external peripherals.

5.1.4.6. Inter-IC Sound (I2S) Bus

The Inter-IC Sound (I2S) Bus is used as an interface for full-duplex serial ports such as those found in audio or voice-band analog to digital converters (ADC) to acquire audio signals or digital to analog converters (DAC) to drive speakers and headphones. It allows serial transfer of full duplex streaming data, usually streaming audio, between DSP and an external I2S peripheral such as an audio codec. The DSP includes four independent I2S modules.

5.1.4.7. Inter-Integrated Circuit (I2C)

The inter-integrated circuit (I2C) peripheral in the device provides an interface between the DSP and other devices that are compliant with the I2C-bus specification and connected to an I2C bus. External components that are attached to this two-wire serial bus can transmit and receive data that is up to eight bits wide both to and from the DSP through the I2C peripheral.

5.1.4.8. Universal Asynchronous Receiver/Transmitter (UART)

The universal asynchronous receiver/transmitter (UART) peripheral performs serial-to-parallel conversions on data received from a peripheral device and parallel-to-serial conversion on data received from the CPU. The CPU can read the UART status at any time. The UART includes control capability and a processor interrupt system that can be tailored to minimize software management of the communications link. The UART can be operated with or without the FIFOs.

5.1.4.9. Multimedia Card/Secure Digital Card (MMC/SD)

The multimedia card (MMC)/secure digital (SD) card is used in a number of applications to provide removable data storage. The MMC/SD card controller provides an interface to MMC, SD and SDHC external cards. The CPU and DMA controller can read/write the data in the card by accessing the registers in the MMC/SD controller.

5.1.4.10. Universal Serial Bus (USB)

The universal serial bus (USB) controller provides a low-cost connectivity solution for consumer portable devices by providing a mechanism for data transfer between USB devices up to 480 Mbps. With the USB controller, the DSP can be used to create a high-speed USB slave device that is complaint with the Universal Serial Bus Specification version 2.0. The USB controller can be allowed to be forced into full-speed mode and high-speed mode may be used for debug purposes.

5.1.4.11. Liquid Crystal Display Controller (LCDC)

The liquid crystal display controller (LCDC) supports asynchronous (memory-mapped) LCD interfaces. The LCD controller consists of two independent controllers, the Raster Controller and the LCD Interface Display Driver (LIDD) controller. Each controller operates independently from the other and only one of them is active at any given time.

5.2. On-Chip Single-Access RAM (SARAM)

SARAM [5] stands for Single- Access Random Access Memory. It is located at the byte address range 010000h - 04FFFFh and is composed of 32 blocks of 4 K words each as shown in Table 5-1. It can perform one access per cycle (one read or one write). In this memory, two dataBuffers namely dataBuffer1 and dataBuffer2 may be accessed. If the dataBuffer1 is full, the data from buffer1 can be filled up to sample the voltage in channel 0 of dataBuffer1 and the current in channel 1 of dataBuffer1. If the condition is not correct, the data from buffer2 can be stored to sample the voltage in channel 0 of buffer2 and the current in channel 1 of buffer2 in the same way. Therefore the sampled data can be stored in the corresponding block number alternatively. And then the active power, reactive power, power factor and energy of the sampled voltage and current can be calculated and stored in the memory block. SARAM can be accessed by the internal program, data, or DMA buses. SARAM is also accessed by the USB and LCD DMA buses.

.

Table 5-1. SARAM Blocks

CPU BYTE ADDRESS RANGE	**MEMORY BLOCK**	**APPLICATION OF MEMORY**
010000h - 011FFFh	SARAM 0	VOLTAGE-SAMPLES
012000h - 013FFFh	SARAM 1	CURRENT-SAMPLES
014000h - 015FFFh	SARAM 2	
016000h - 017FFFh	SARAM 3	
018000h - 019FFFh	SARAM 4	
01A000h - 01BFFFh	SARAM 5	
01C000h - 01DFFFh	SARAM 6	
01E000h - 01FFFFh	SARAM 7	
020000h - 021FFFh	SARAM 8	VOLTAGE-SAMPLES
022000h - 023FFFh	SARAM 9	CURRENT-SAMPLES
024000h - 025FFFh	SARAM 10	
026000h - 027FFFh	SARAM 11	
028000h - 029FFFh	SARAM 12	
02A000h - 02BFFFh	SARAM 13	
02C000h - 02DFFFh	SARAM 14	
02E000h - 02FFFFh	SARAM 15	
030000h - 031FFFh	SARAM 16	ENERGY-DATA
032000h - 033FFFh	SARAM 17	
034000h - 035FFFh	SARAM 18	
036000h - 037FFFh	SARAM 19	
038000h - 039FFFh	SARAM 20	
03A000h - 03BFFFh	SARAM 21	
03C000h - 03DFFFh	SARAM 22	
03E000h - 03FFFFh	SARAM 23	
040000h - 041FFFh	SARAM 24	
042000h - 043FFFh	SARAM 25	
044000h - 045FFFh	SARAM 26	
046000h - 047FFFh	SARAM 27	
048000h - 049FFFh	SARAM 28	
04A000h - 04BFFFh	SARAM 29	
04C000h - 04DFFFh	SARAM 30	
04E000h - 04FFFFh	SARAM 31	

5.3. On-Chip Dual - Access RAM (DARAM)

DARAM [5] stands for Dual Access Random Access Memory. Its CPU byte address range is between 000000h and 00FFFFh and it consists of eight blocks 4K words each in contrast with SARAM as shown in Table 5-2. Each DARAM block can perform two accesses per cycle (two reads, two writes, or a read and a write). In this memory, the data can be stored to the given block and the fast Fourier transform of the sampled voltage can be calculated in the same block during the same cycle. Similarly the fast Fourier transform of the current can also be stored and calculated in the block at the corresponding cycle. The DARAM can be accessed by the internal program, data, or DMA buses just like the SARAM.

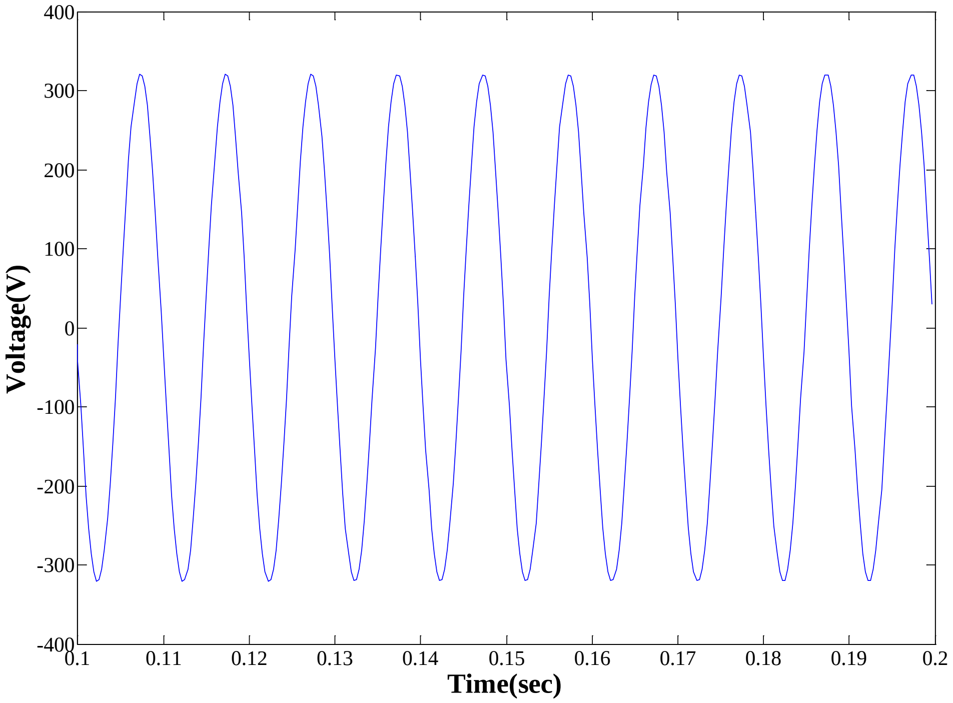
Table 5-2. DARAM Blocks

CPU BYTE ADDRESS RANGE	**MEMORY BLOCK**	**APPLICATION OF MEMORY**
000000h - 001FFFh	DARAM 0	FFT OF VOLTAGE
002000h - 003FFFh	DARAM 1	FFT OF CURRENT
004000h - 005FFFh	DARAM 2	
006000h - 007FFFh	DARAM 3	
008000h - 009FFFh	DARAM 4	
00A000h - 00BFFFh	DARAM 5	
00C000h - 00DFFFh	DARAM 6	
00E000h - 00FFFFh	DARAM 7	

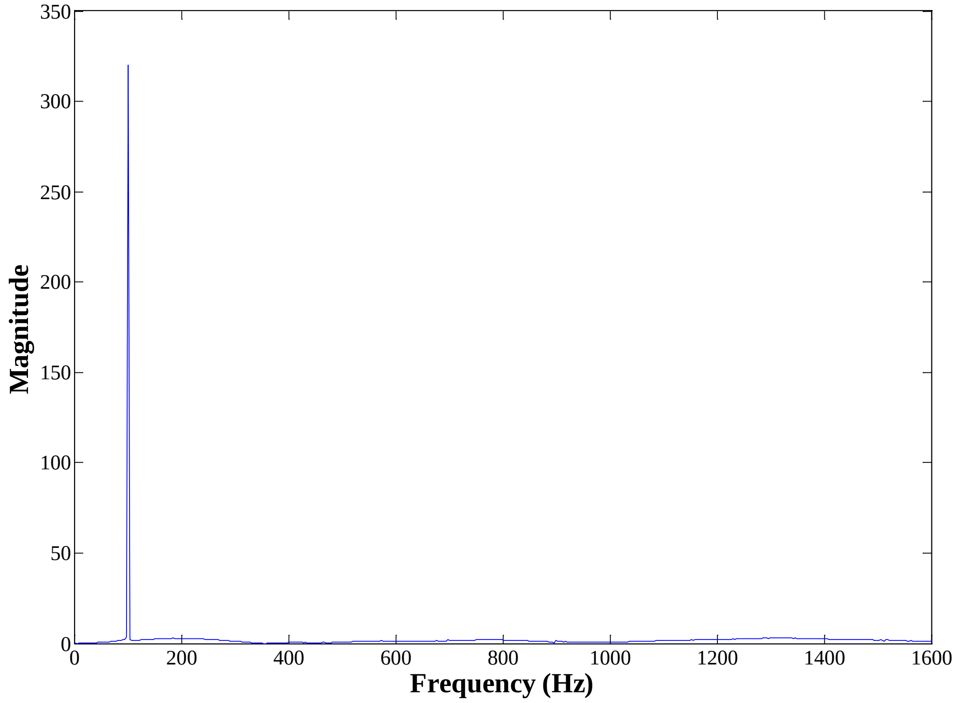
CHAPTER VI

VERIFICATIONS AND TESTING

Since the resultant data from the DSP board can't be transferred to the personal computer all the time, the GUI-panel may be used to request the data. In this chapter, the simulation results of the various waveforms of the voltage and current are shown with the following examples. In the first testing, the sinusoidal 100Hz frequency of output voltage and the sinusoidal 50Hz frequency with 3rd and 5th harmonics of output current are generated using data acquisition board. The amplitude of the voltage and current from the data acquisition board can be considered as 2V. The scaling factor for 2V is approximately equal to 0.049. Therefore the amplitude of the generated signal is 311V in both waveforms. The simulated voltage and current waveforms and their corresponding FFT magnitudes are illustrated in Fig.6-1.



(a) Simple sinusoidal 100Hz frequency of voltage waveform

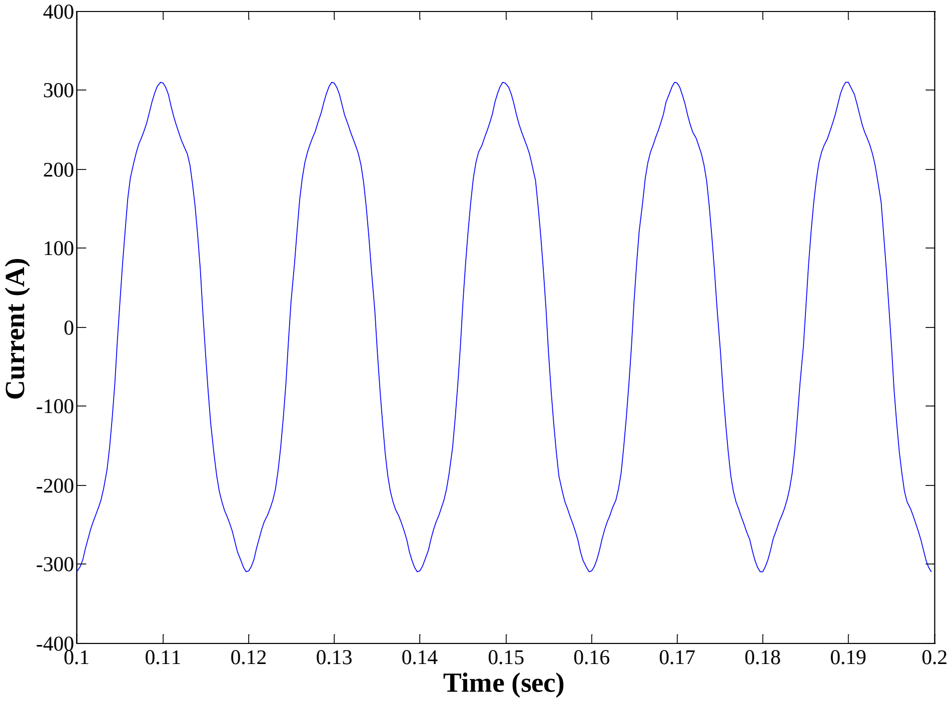


(b) FFT magnitude of voltage in Fig (a)

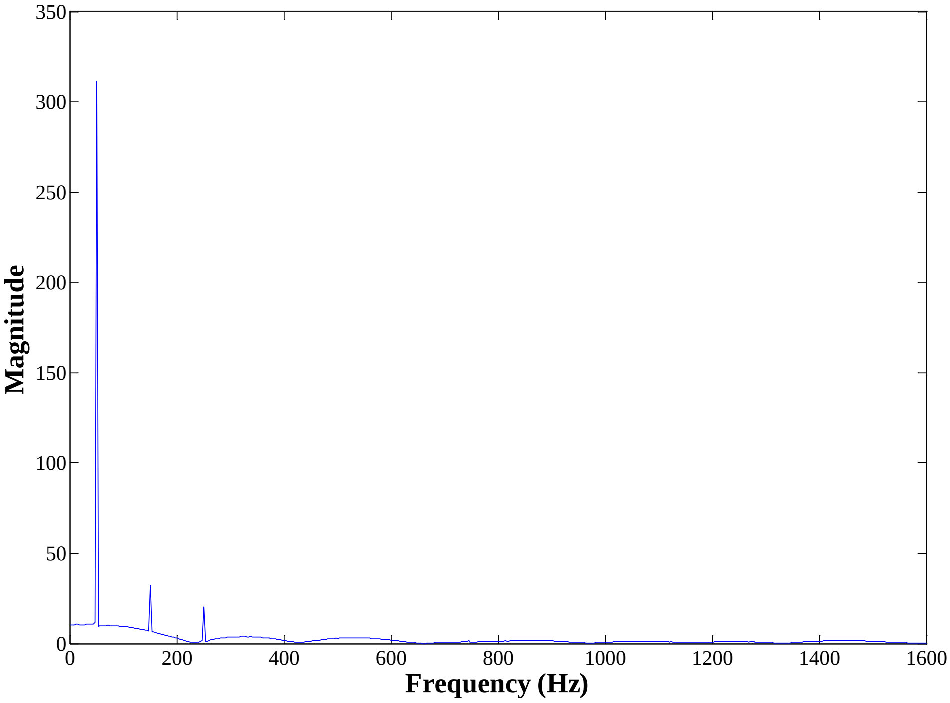
The equation for generating the analog current signal is described in equation (6.1)

$$a \times \sin(2\pi f \times t) + a/10 \times \sin(2\pi f \times 3 \times t) + a/15 \times \sin(2\pi f \times 5 \times t) \qquad (6.1)$$

In the above equation, the fundamental frequency is 50Hz, the amplitude of the 3rd harmonics is 1/10 times of the fundamental frequency and the amplitude of the 5th harmonics is 1/15 times of the fundamental frequency. Therefore the amplitude of the FFT spectrum at the frequency of 150Hz is approximately equal to 31.1V and at the frequency of 250Hz is approximately equal to 20.7V. It can be illustrated in Fig.6-1 (d).



(c) Sinusoidal frequency 50Hz with 3rd and 5th harmonics of current waveform



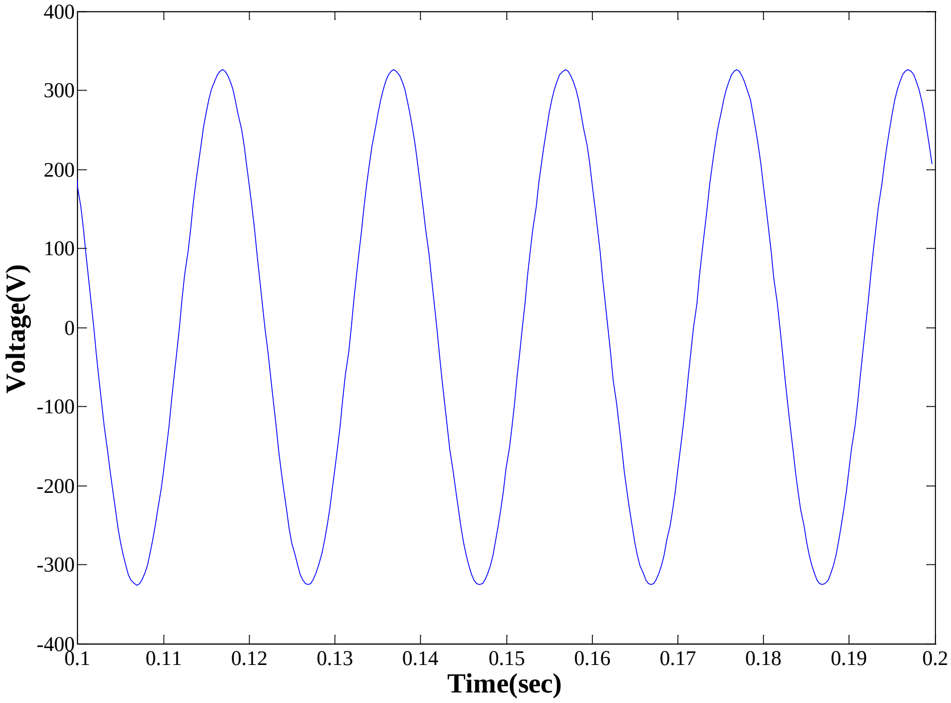
(d) FFT magnitude of current in Fig (c)

Figure.6-1.Measured several waveforms from experiment

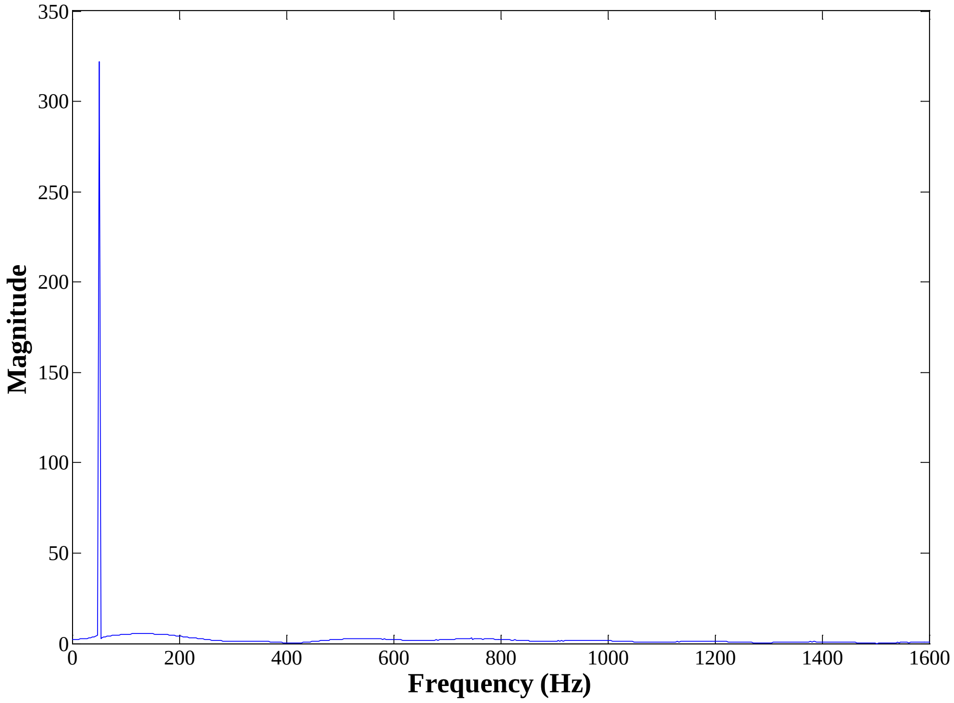
In the next testing, the pure sinusoidal waveform 50Hz frequency and the square waveform of 25Hz frequency from 10ms to 20ms can be considered as the analog output voltage and current. The analog output current pulse can be generated by the following equation:

$$step(t-0.01)-step(t-0.02) \tag{6.2}$$

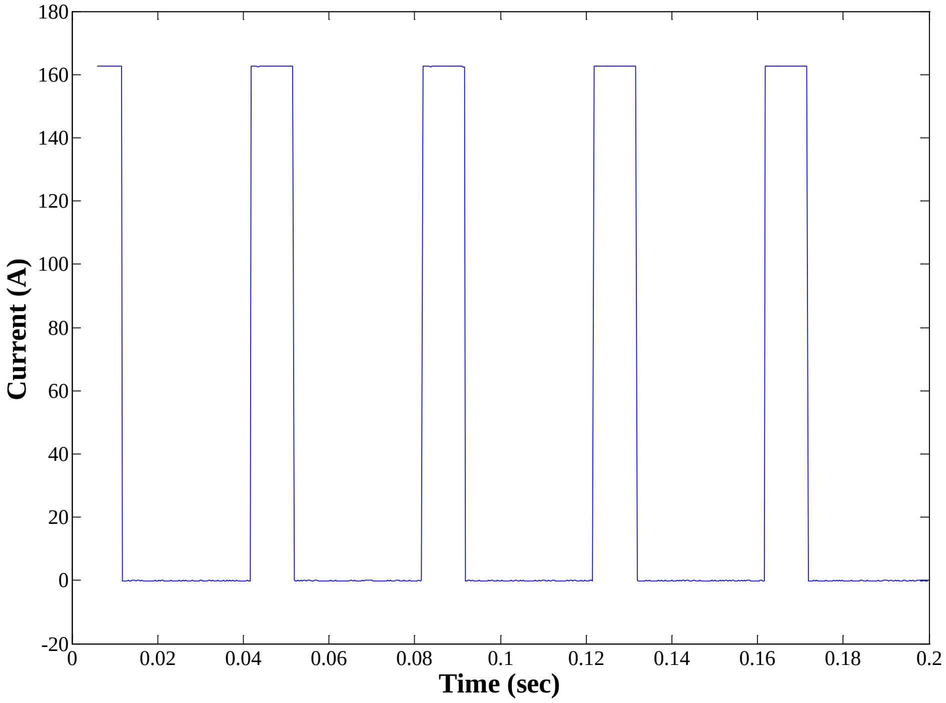
which means that the amplitude of current signal increases at t=10ms and decreases at t=20ms. According to the above testing, the simulated values of voltage and current waveforms and the corresponding FFT magnitude of voltage and current are illustrated in Figure.6-2. Note that the FFT of the current in Fig.6-2(d) follows the $\frac{\sin x}{x}$ shape which agrees with theory.



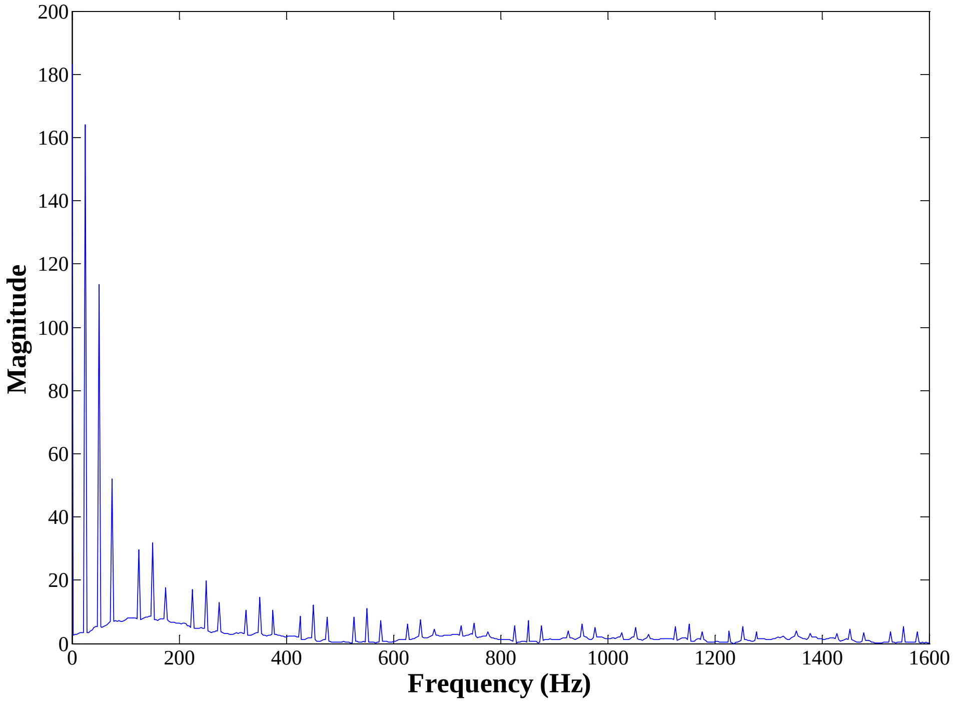
(a) Pure sinusoidal waveform of 50Hz frequency of analog output voltage



(b) FFT magnitude of voltage in Fig. (a)



(c) Square waveform from 10ms to 20ms of output current with 25Hz frequency



(d) FFT magnitude of current in Fig (c)

Figure.6-2.Measured waveforms from testing

In the next experiment, the sinusoidal waveform of 50Hz frequency with 3rd harmonics of analog output current is generated in Fig.6-3 (a) with its FFT output shown in Fig.6-3(b). The equation for generating the analog output current with 3rd harmonics is

$$a \times \sin(2\pi f \times t) + a/2 \times \sin(2\pi f \times 3 \times t) \tag{6.3}$$

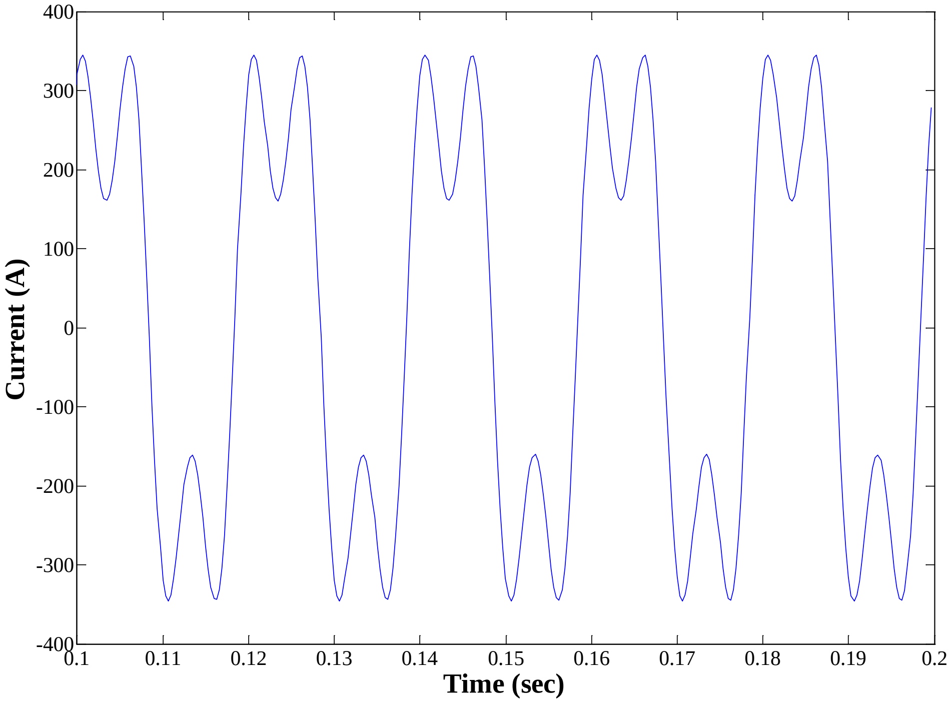
where the amplitude of the 3rd harmonics is half of the fundamental frequency. So the amplitude of FFT spectrum of the 3rd harmonics is approximately equal to half of the amplitude of current at the frequency of 50 Hz. The simulated values of voltage and current and their respective FFT magnitudes can be produced in the following Figure.6-3.

A 50Hz sinusoidal wave with 80% under voltage at 100ms of analog output voltage can be tested. The analog output voltage signal has a period of 100ms can be generated with the following equation:

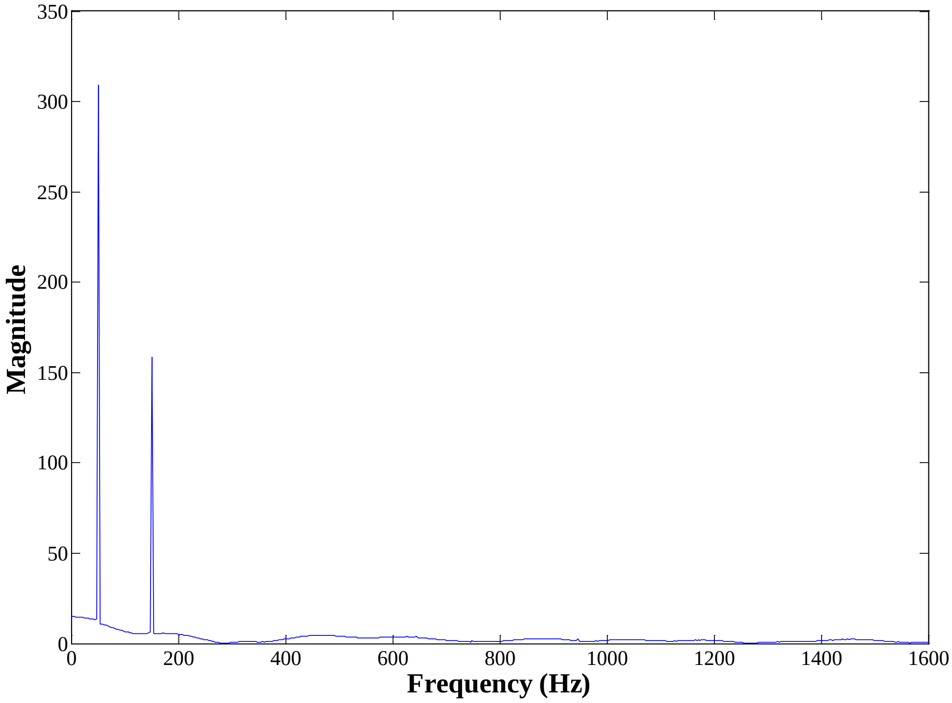
$$a \times \sin(2\pi f \times t) - 0.8 \times a \times \sin(2\pi f \times t) \times step(t - 0.1) \tag{6.4}$$

where f is the fundamental frequency and its value is 50Hz. In this case when the time is at 100ms, the amplitude is 0.8 less than the value of the sinusoidal waveform.

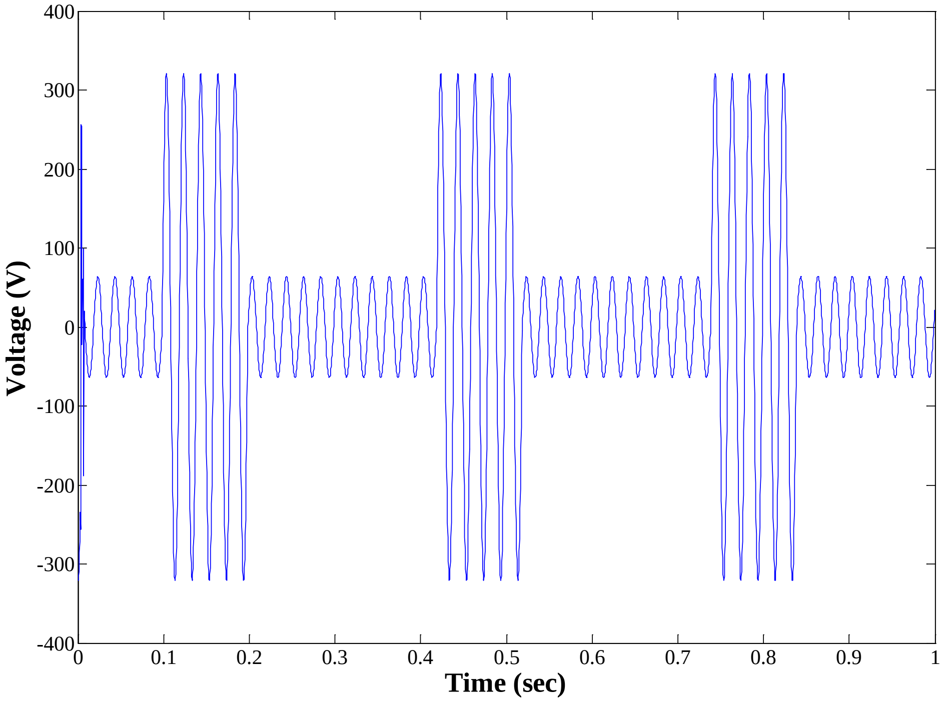
Therefore the amplitude of the voltage waveform is decreased until about 60 V at t=100ms.



(a) Current waveform of sinusoidal wave 50Hz frequency with 3rd harmonics



(b) FFT magnitude of current in Fig (a)



(c) Voltage waveform of sinusoidal wave with 80% under voltage at 100ms

Figure.6-3. Measured waveforms from testing

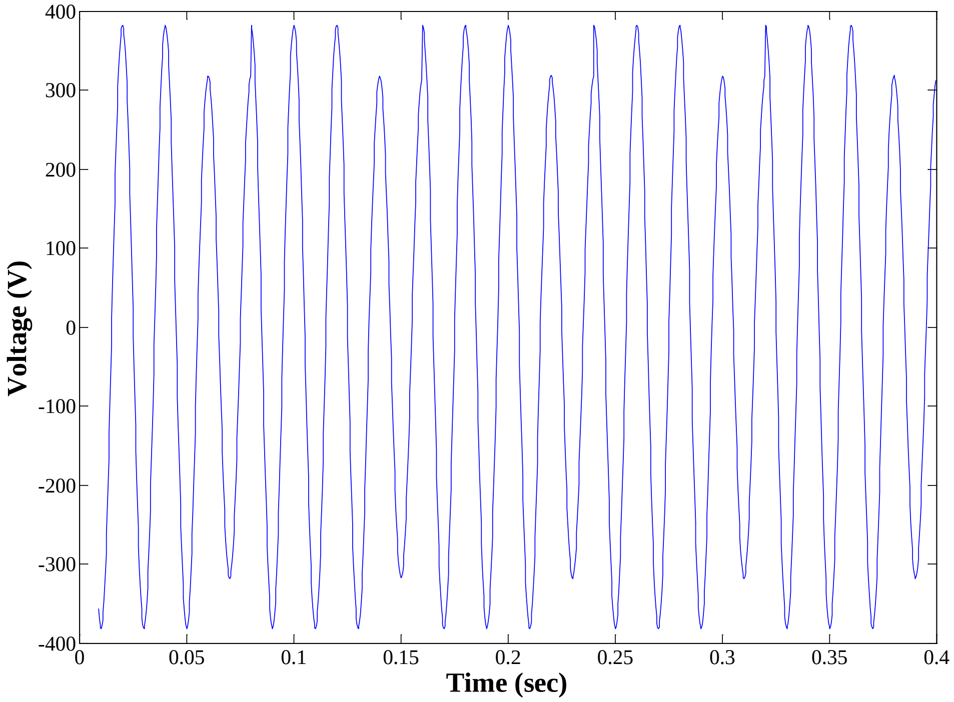
In the next experiment, the sinusoidal 100Hz frequency with 20% overvoltage at 25ms of analog output voltage and the sinusoidal 50Hz frequency decrease in current at 10ms of analog output current can be tested. The analog output voltage can be calculated as the following formula:

$$a \times \sin(2\pi f \times t) + 0.2 \times a \times \sin(2\pi f \times t) \times step(t - 0.025) \tag{6.5}$$

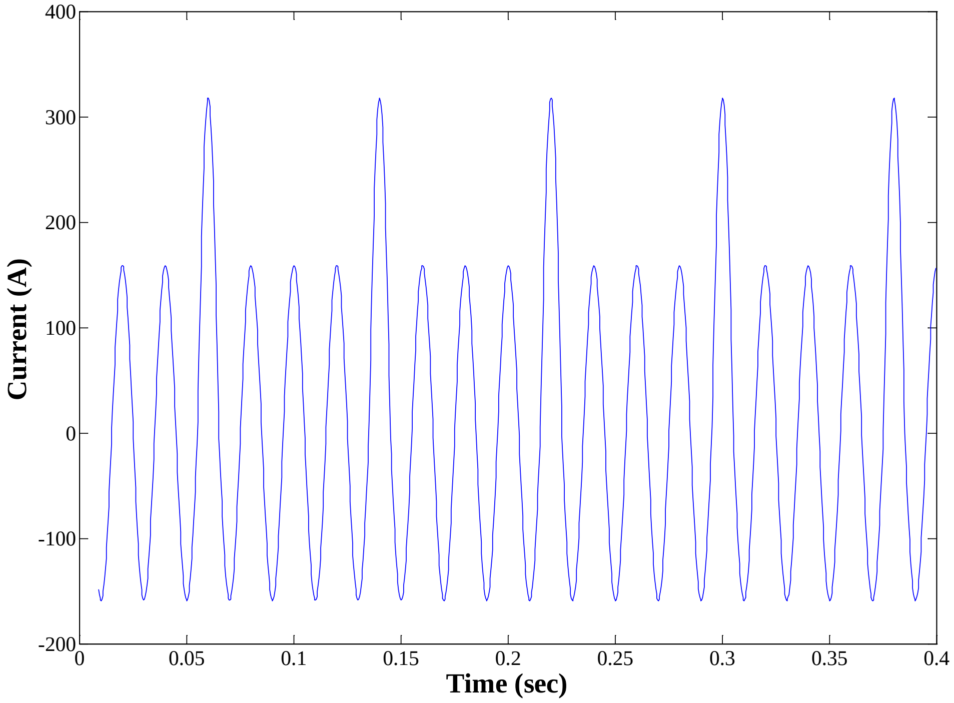
where the amplitude of the voltage waveform is increased in 20% of the value of the pure sinusoidal waveform at t=25ms. The sinusoidal wave 50Hz frequency decrease in current at 10ms can be evaluated with the equation:

$$(a \times step(t) - a/2 \times step(t - 0.01)) \times \sin(2\pi f \times t) \tag{6.6}$$

where the amplitude of the current waveform is decreased in 1/2 times of the reference input value when t is equal to 10ms. The fundamental frequency used in this equation is 50Hz. The simulated results of voltage and current by using Matlab programming can be shown in Figure.6-4.



(a) Sinusoidal waveform of 100Hz frequency with 20% overvoltage at 25ms



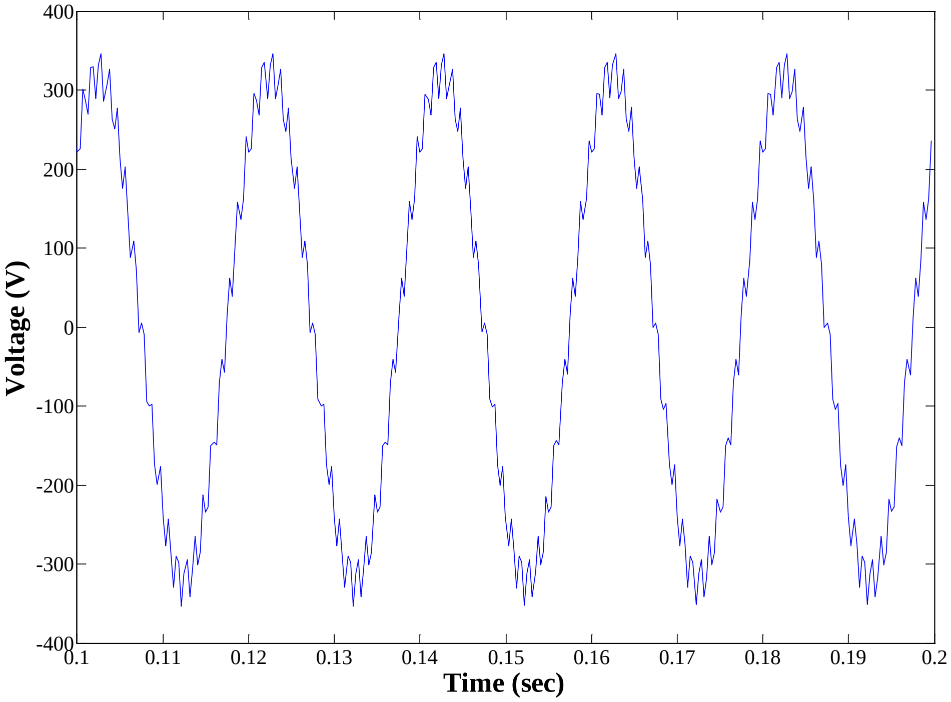
(b) Sinusoidal wave 50Hz frequency decrease in current at 10ms

Figure.6-4. Measured waveforms from testing

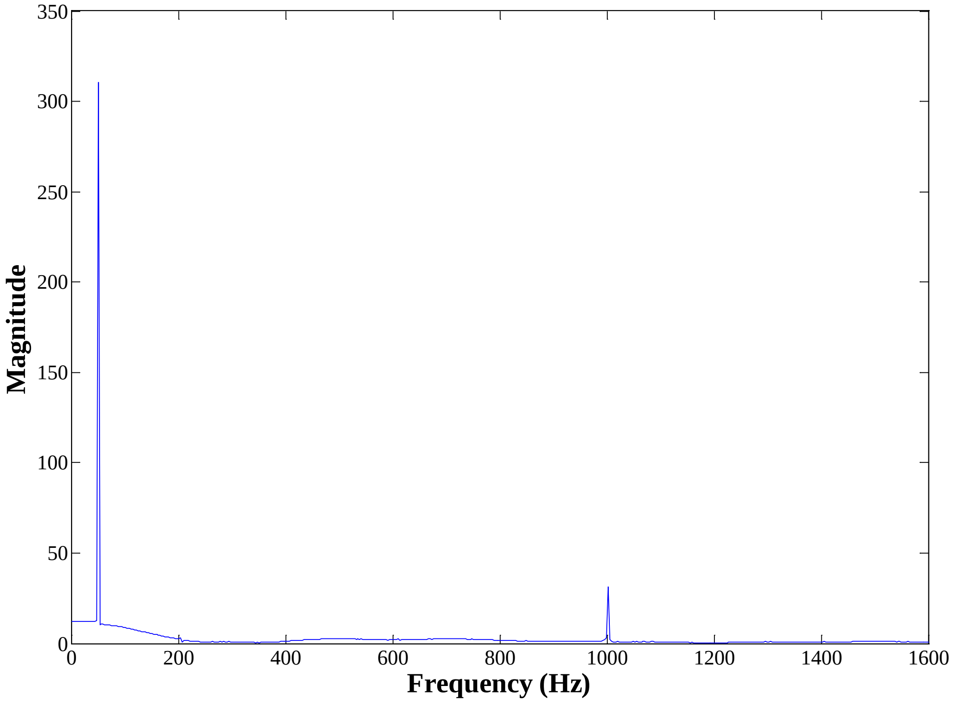
In accordance with the following Figure.6-5, not only the measured waveforms of voltage and current but also the FFT magnitudes can be produced. The 50Hz sinusoidal waveform with 20th harmonics of analog output voltage and the sawtooth wave of 100Hz frequency of analog output current can be generated. The 50Hz sinusoidal waveform with 20th harmonics of analog output voltage can be generated by the following equation:

$$a \times \sin(2\pi f \times t) + a/10 \times \sin(2\pi f \times 20 \times t) \qquad (6.7)$$

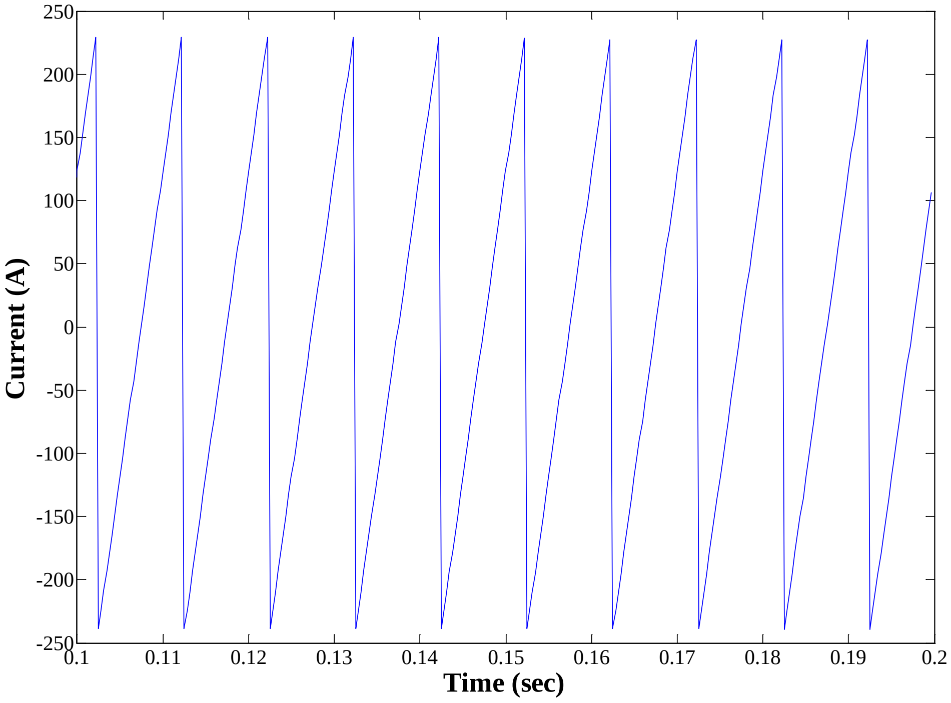
where the fundamental frequency is 50Hz. The amplitude of the FFT spectrum is approximately equivalent to the value of 31.1 Volt at the frequency of 1000Hz. This can be shown in Figure.6-4 (b).



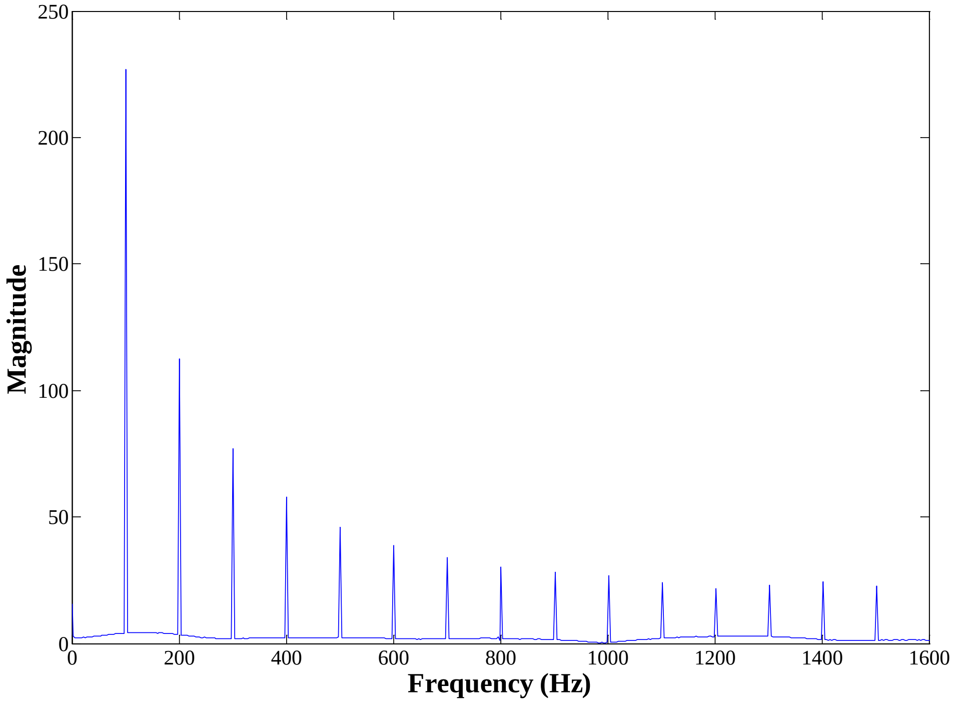
(a) Sinusoidal waveform 50Hz frequency with 20th harmonics



(b) FFT magnitude of voltage in Fig (a)



(c) Sawtooth waveform with 100Hz frequency



(d) FFT magnitude of current in Fig (c)

Figure.6-5. Measured waveforms from experiment

CHAPTER VII

CONCLUSIONS

This dissertation focuses on two platforms for developing and testing a power quality meter. Many designers or manufacturers face with power quality problems because many power electronics equipments are very sensitive to power system disturbances. Therefore some power quality disturbances faced with in utilities as well as industries and how to monitor these problems are given in Chapter 2. There are many algorithms for implementing a power quality meter on a personal computer. In order to implement the various algorithms, such as calculating fast Fourier transform (FFT) and RMS values of sample voltage and current , the manual calculation of FFT and RMS will be needed to compute. Therefore the literature review of FFT and RMS value calculation are discussed in Chapter 3. In Chapter 4, the internal block diagram of the power quality meter that is the important part in this dissertation is described. In this chapter, the two major platforms such as PQ platform used for PQ measurement with its some specific specifications and arbitrary platform used for testing with its general specifications are explained in details. In real situation, it is difficult to find various conditions for testing a power quality meter. So arbitrary waveform generator will be needed with large memory for storing sample data.

The analog input signals of voltage and current generated from a personal computer using digital to analog converter called DAQ board may be sent to the analog-to-digital converter. And then the digitized value of voltage and current may be sent to the digital signal processor for performing digital calculations to obtain the desired PQ parameters. The designer can write a program on a personal computer to implement various algorithms such calculating FFT , RMS, power factor, and any other values of voltage and current and download the program on the DSP board for execution via a black USB. The hardware description of the board is given in Chapter 5. In this portion the important part of the dsp board such as dsp chip, FFT hardware accelerator built in dsp chip, memory allocation in the block and interfaced peripherals is described. The generated analog output signals of voltage and current from the data acquisition board are compared with the data produced from the DSP board by using the GUI panel in MATLAB programming in Chapter 6, based on simulation results.

REFERENCES

[1] IEEE std 1159-1995, IEEE Recommended Practice for Monitoring Electric Power Quality, 1995.

[2] Rulph Chassaing. Digital Signal Processing: Laboratory Experiments Using C and TMS320C31 DSK , 605 Third Avenue, New York: John Wiley & Sons, Inc, 1999

[3] Root Mean Square Value of the voltage and current . [Online]. Available from: http://www.analog.com/static/importedfies/application_notes/AN578_a.pdf

[4] Rahim bin Abdullah,A and Zuri bin Sha' ameri,A, Real-Time Power Quality Monitoring System Based on TMS320CV5416 DSP Processor , International Conference on Power Electronics and Drives Systems, Vol.2, pp.1668-1672, 2005

[5] TMS320C5515 Fixed-Point Digital Signal Processor [Online].2012.Available from: http://www.ti.com/lit/ds/sprs645e.pdf [2012, January]

[6] NI USB 6216Bus-Powered M Series Multifunction DAQ [Online]. Available from: http://sine.ni.com

[7] FFT Implementation on the TMS320VC5505, TMS320C5505, and TMS320C5515 DSPs [Online].2010. Available from: http://www.ti.com/lit/an/sprabb6z.pdf [2010,June]

[8] 16,14,12-Bit, Six-Channel, Simultaneous Sampling ANALOG-TO-DIGITAL CONVERTERS [Online].2012.Available from: www.focus.ti.com [2012, January]

[9] NI USB-6216,16-Bit,400kS/s Isolated M Series MOI DAQ, Bus-Powered [Online]. Available from: http://sine.ni.com/nips/cds/view/p/lang/en/nid/207099#resources

APPENDICES

APPENDIX A
MATLAB Programming for Calculating the FFT value

```
 % Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @PQ_GUI_PANEL_OpeningFcn, ...
                   'gui_OutputFcn',  @PQ_GUI_PANEL_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end
 if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT
 %% --- Executes just before PQ_GUI_PANEL is made visible.
function PQ_GUI_PANEL_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to PQ_GUI_PANEL (see VARARGIN)
clc;
global SerialObj sine_data cosine_data;
 % Choose default command line output for PQ_GUI_PANEL
handles.output = hObject;
% Update handles structure
```

```
guidata(hObject, handles);
 % UIWAIT makes PQ_GUI_PANEL wait for user response (see UIRESUME)
% uiwait(handles.figure1);
 % Define parameter for serial object
COMPORT = 'COM4';
% Baudrate = 9600;
% Baudrate = 57600;
Baudrate = 115200;
DataBits = 8;
None = 'none';
StopBits = 1;
% ByteOrder = 'bigEndian';
ByteOrder = 'littleEndian';
InputBufferSize = 40000;
OutputBufferSize = 1000;
Timeout = 4;
 disp(['Create SerialObj ',COMPORT]);
SerialObj = serial(COMPORT);
set(SerialObj,'baudrate',Baudrate,...
            'dataBits',DataBits,...
            'flowControl',None,...
            'parity',None,...
            'stopBits',StopBits,...
            'byteOrder',ByteOrder,...
            'inputBufferSize',InputBufferSize,...
            'outputBufferSize',OutputBufferSize,...
            'Timeout',Timeout);
set(handles.Comport_tb, 'String', COMPORT);
set(handles.ComStatus, 'String', get(SerialObj,'Status'));
disp(['COMPORT Status = ',get(SerialObj,'status')]);
disp('Opening PQ_GUI_PANEL');
%% --- Executes during object creation, after setting all properties.
```

```
function Graph_menu_CreateFcn(hObject, eventdata, handles)
% hObject    handle to Graph_menu (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called
global SelectNum;
% Hint: popupmenu controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
   set(hObject,'BackgroundColor','white');
end
set(hObject,'String',strvcat('DATA Voltage',...
                   'DATA Current',...
                   'FFT_Voltage_Real',...
                   'FFT_Voltage_Imag',...
                   'FFT_Current_Real',...
                   'FFT_Current_Imag'));
SelectNum = get(hObject,'Value');
disp('Graph Menu Create');
 %% --- Executes during object creation, after setting all properties.
function Graph1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to Graph1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called
global LevelZoom;
% Hint: place code in OpeningFcn to populate Graph1
 LevelZoom = 1;
 %% --- Executes during object creation, after setting all properties.
function Filename_CreateFcn(hObject, eventdata, handles)
% hObject    handle to Filename (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called
% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
 %% --- Outputs from this function are returned to the command line.
function varargout = PQ_GUI_PANEL_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
 % Get default command line output from handles structure
varargout{1} = handles.output;
disp(datestr(now));
 %% --- Executes on selection change in Graph_menu.
function Graph_menu_Callback(hObject, eventdata, handles)
% hObject    handle to Graph_menu (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global SelectNum;
% Hints: contents = cellstr(get(hObject,'String')) returns Graph_menu contents as cell
array
%        contents{get(hObject,'Value')} returns selected item from Graph_menu
MenuSelect = cellstr(get(hObject,'String'));
SelectNum = get(hObject,'Value');
set(handles.SystemMsg,'String', MenuSelect{SelectNum});
% disp(['Graph Menu Changed to ' MenuSelect{SelectNum}]);
 %% --- Executes on button press in View_pb.
function View_pb_Callback(hObject, eventdata, handles)
% hObject    handle to View_pb (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLA
% handles    structure with handles and user data (see GUIDATA)
global read_data3 read_data4 fft_V fft_I LevelZoom;
 % change sampling rate here
sampling_rate = 3200;
time = (1:3200)/sampling_rate;
 % change frequency domain
sample = 512;
freq = (1:sample);
freq = (freq -1 )*sampling_rate/2/(sample-1);
axes(handles.Graph1);
num = get(handles.Graph_menu,'Value');
% disp(num);
switch num
    case 1
        plot(read_data3);
%         set(handles.Graph1,'YLim',[-30000 30000],'XLim',[0 3200]);
        set(handles.Graph1,'YLim',[-30000 30000],'XLim',[400 800]);
        LevelZoom = 4;
    case 2
        plot(read_data4);
%         set(handles.Graph1,'YLim',[-30000 30000],'XLim',[0 3200]);
        set(handles.Graph1,'YLim',[-30000 30000],'XLim',[400 800]);
        LevelZoom = 4;
    case 3
        plot(freq,fft_V.real);
%         set(handles.Graph1,'XLim',[0 1024]);
        set(handles.Graph1,'XLim',[0 sampling_rate/2]);
    case 4
        plot(freq,fft_I.imag);
%         set(handles.Graph1,'XLim',[0 1024]);
        set(handles.Graph1,'XLim',[0 sampling_rate/2]);
    case 5
```

```
        plot(freq,fft_V.real);
%         set(handles.Graph1,'XLim',[0 1024]);
        set(handles.Graph1,'XLim',[0 sampling_rate/2]);
    case 6
        plot(freq,fft_I.imag);
%         set(handles.Graph1,'XLim',[0 1024]);
        set(handles.Graph1,'XLim',[0 sampling_rate/2]);
end
 set(handles.SystemMsg,'String','Push View Button');
% disp('Push View Button');
 %% --- Executes on button press in Comport_tb.
function Comport_tb_Callback(hObject, eventdata, handles)
% hObject    handle to Comport_tb (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global SerialObj;
% Hint: get(hObject,'Value') returns toggle state of Comport_tb
 ComStatus = get(hObject,'Value');
if(ComStatus)
    fopen(SerialObj);
    if(strcmp(get(SerialObj,'Status'),'open'))
        disp('COMPORT CONNECTED');
        set(handles.SystemMsg,'String','COMPORT CONNECT SUCCESSFULL');
%         fprintf(SerialObj,'%s','p');
    else
        set(handles.SystemMsg,'String','CANNOT CONNECT TO COMPORT');
        set(hObject,'Value', 0);
    end
else
    fclose(SerialObj);
    if(strcmp(get(SerialObj,'Status'),'closed'))
        disp('COMPORT DISCONNECTED');
        set(handles.SystemMsg,'String','DISCONNECT TO COMPORT');
    end
end
set(handles.ComStatus,'String',get(SerialObj,'Status'));
% disp(['Toggle Comport ',num2str(ComStatus)]);
 %% --- Executes on button press in Update_pb.
function Update_pb_Callback(hObject, eventdata, handles)
% hObject    handle to Update_pb (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global SerialObj read_data1 read_data2 read_data3 read_data4 fft_I fft_V;
% cmd = 'P';
data_1_sec = 3200;
data_len = 1024*2;
 % disp('Push Update Button');
set(handles.SystemMsg,'String','Push Update Button');
while(get(SerialObj,'BytesAvailable') < 1)
    fwrite(SerialObj,13);
    pause(3);
end
 % header = fscanf(SerialObj,'%s',6);
% set(handles.SystemMsg,'String',header);
% fprintf(SerialObj,'%s',cmd);
 % read FFT_I from DSP Board
[read_data1,count] = fread(SerialObj,data_len,'int16');
disp(['fft_result_I = ' num2str(count)]);
 % read FFT_V from DSP Board
[read_data2,count] = fread(SerialObj,data_len,'int16');
disp(['fft_result_V = ' num2str(count)]);
 % read Data Voltage from DSP Board
[read_data3,count] = fread(SerialObj,data_1_sec,'int16');
disp(['adc_data_V = ' num2str(count)]);
```

```
% read Data Current from DSP Board
[read_data4,count] = fread(SerialObj,data_1_sec,'int16');
disp(['adc_data_I = ' num2str(count)]);
 % read Calculate Data from DSP Board
Data1 = fread(SerialObj,3,'uint32');
tmp = fread(SerialObj,8,'uint32');
% result = ['Volt = ' int2str(Data(1)); 'Current = ' int2str(Data(2))];
% set(handles.DataMsg,'String',['DATA RESULT'; int2str(Data); int2str(Data2)]);
byte = get(SerialObj,'BytesAvailable');
disp(['Over_data = ' num2str(byte)]);
if(byte~=0)
   fread(SerialObj,byte,'char');
   set(handles.SystemMsg,'String',['Read Data Left ' num2str(byte)]);
else
%     fft_V = struct('real',read_data2(2:2:2048),'imag',read_data2(1:2:2048));
%     fft_I = struct('real',read_data1(2:2:2048),'imag',read_data1(1:2:2048));
   fft_V = struct('real',read_data2(2:2:1024),'imag',read_data2(1:2:1024));
   fft_I = struct('real',read_data1(2:2:1024),'imag',read_data1(1:2:1024));
   ActPower = tmp(2)*2^32 + tmp(1);
   AprPower = tmp(4)*2^32 + tmp(3);
   EGY_Sum = tmp(6)*2^32 + tmp(5);
   EGY = tmp(8)*2^32 + tmp(7);
%     Data2 = [tmp(1)+tmp(2)*65536; tmp(3)+tmp(4)*65536; tmp(5)+tmp(6)*65536;
tmp(7)+tmp(8)*65536];
%     disp(int2str(Data1));
%     disp(int2str(Data2));
   Sdata = struct('Volt_RMS',Data1(1),...
            'Current_RMS',Data1(2),...
            'PowerFactor',Data1(3),...
            'ActPower',ActPower,...
            'AprPower',AprPower,...
            'EGY_Sum',EGY_Sum,...
```

```
          'EGY',EGY);
  set(handles.SystemMsg,'String','Read Data Success.');
  Result = strvcat(int2str(Sdata.Volt_RMS),...
           int2str(Sdata.Current_RMS),...
           int2str(Sdata.PowerFactor),...
           int2str(Sdata.ActPower),...
           int2str(Sdata.AprPower),...
           int2str(Sdata.EGY_Sum),...
           int2str(Sdata.EGY));
%    disp(Result);
  set(handles.DataMsg,'String',Result);
end
function Filename_Callback(hObject, eventdata, handles)
% hObject    handle to Filename (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global FileName index;
% Hints: get(hObject,'String') returns contents of Filename as text
%        str2double(get(hObject,'String')) returns contents of Filename as a double
 FileName=get(hObject,'String');
index = 0;
set(hObject,'BackgroundColor','green');
 %% --- Executes on button press in Save_pb.
function Save_pb_Callback(hObject, eventdata, handles)
% hObject    handle to Save_pb (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global FileName index read_data1 read_data2 read_data3 read_data4;
 FileID=[FileName int2str(index) '.txt'];
FileObj=fopen(FileID,'a+');
fprintf(FileObj,'%d\n',read_data1);
fprintf(FileObj,[datestr(now) '\n']);
```

```
fclose(FileObj);
index = index+1;
FileID=[FileName int2str(index) '.txt'];
FileObj=fopen(FileID,'a+');
fprintf(FileObj,'%d\n',read_data2);
fprintf(FileObj,[datestr(now) '\n']);
fclose(FileObj);
index = index+1;
FileID=[FileName int2str(index) '.txt'];
FileObj=fopen(FileID,'a+');
fprintf(FileObj,'%d\n',read_data3);
fprintf(FileObj,[datestr(now) '\n']);
fclose(FileObj);
index = index+1;
FileID=[FileName int2str(index) '.txt'];
FileObj=fopen(FileID,'a+');
fprintf(FileObj,'%d\n',read_data4);
fprintf(FileObj,[datestr(now) '\n']);
fclose(FileObj);
% index = index+1;
% FileID=[FileName int2str(index) '.txt'];
% FileObj=fopen(FileID,'a+');
% fprintf(FileObj,'%d\n',read_data5);
% fprintf(FileObj,[datestr(now) '\n']);
% fclose(FileObj);
Status = ['File ' FileName ' Save'];
set(handles.SystemMsg,'String',Status);
 %% --- Executes on button press in Zoom_pb.
function Zoom_pb_Callback(hObject, eventdata, handles)
% hObject    handle to Zoom_pb (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
```

```
global LevelZoom XLim SelectNum;
 axes(handles.Graph1);
 if(SelectNum<3)
   switch LevelZoom
      case 1
         Left = 0;
         Right = 1600;
         LevelZoom = 2;
      case 2
         Left = 0;
         Right = 800;
         LevelZoom = 3;
      case 3
         Left = 0;
         Right = 400;
         LevelZoom = 4;
      case 4
         Left = 0;
         Right = 200;
         LevelZoom = 5;
      case 5
         Left = 0;
         Right = 100;
         LevelZoom = 6;
      case 6
         Left = 0;
         Right = 3200;
         LevelZoom = 1;
   end
   XLim = [Left, Right];
   set(handles.Graph1,'XLim',[Left Right]);
end
```

```
%% --- Executes on button press in Next_pb.
function Next_pb_Callback(hObject, eventdata, handles)
% hObject    handle to Next_pb (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global LevelZoom SelectNum XLim;
 axes(handles.Graph1);
 Left    = XLim(1);
Right   = XLim(2);
Width   = Right-Left;
 if(LevelZoom ~= 1) && (SelectNum<3)
    if(Right<3200)
        Left = Left+Width;
        Right = Right+Width;
    else
        Left = 0;
        Right = Width;
    end
    XLim = [Left Right];
    set(handles.Graph1,'XLim',XLim);
end
%% --- Executes when user attempts to close figure1.
function figure1_CloseRequestFcn(hObject, eventdata, handles)
% hObject    handle to figure1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global SerialObj;
 % Hint: delete(hObject) closes the figure
delete(hObject);
clear all;
clc;
 %% --- Executes during object deletion, before destroying properties.
```

```
function figure1_DeleteFcn(hObject, eventdata, handles)
% hObject    handle to figure1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
```

APPENDIX B

List of Publication

The following publication is written during the course of this research work.

1. Theingi Zin and Ekachai Leelarasmee, “ Platforms for Developing Power Quality Meter and Arbitrary Waveform Generator”, Proceedings of the 2011 International Symposium on Electrical & Electronics Engineering (ISEE 2011, Ho Chi Minh City, Vietnam), November 2011.

Platforms for Developing Power Quality Meter and Arbitrary Waveform Generator

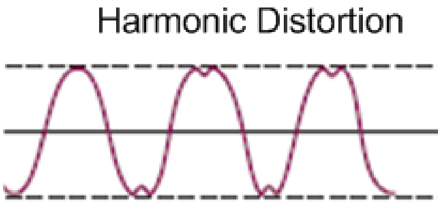
Theingi Zin and Ekachai Leelarasmee
Chulalongkorn University
Electrical Engineering Department
Bangkok 10330, Thailand
Tel./Fax. +662 218 6488
Email: Ekachai.l@chula.ac.th

Abstract— A hardware platform suitable for developing a two phase power quality (PQ) meter is described. It is based on a DSP development board having two 16 bit 20KSample/Sec ADC channels and a USB 2.0 port. The designer can write a C program on a personal computer to implement various algorithms, such as calculating FFT and RMS values, and download the program to the DSP board for execution. To test whether the developed PQ meter works correctly, another platform or test bench for generating arbitrary waveforms is described. It consists of a data acquisition (DAQ) board that is connected with a personal computer. Waveforms of arbitrary shapes such as pure sinusoidal of any frequency, harmonics of any order, voltage sag and transient can be programmed on the computer and download to the DAQ board in real time. Using these flexible platforms, the development of a PQ meter with various features can be easily constructed and tested.

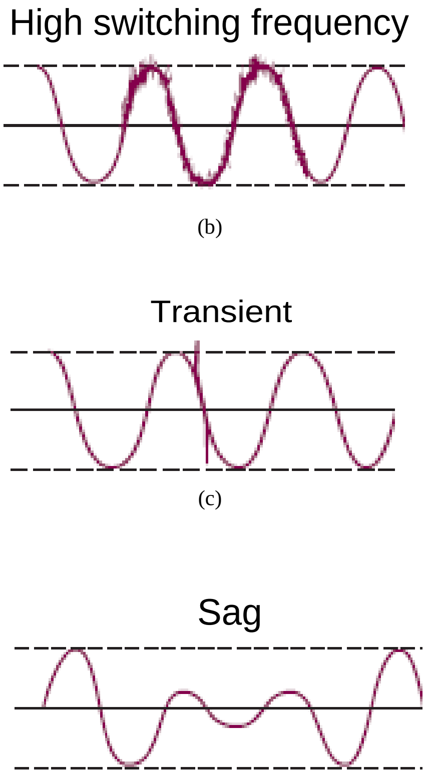
.

I. Introduction

The increases in using power electronic devices (such as inverters) and installing varying renewable energy power plants (such as wind or solar) have significantly deteriorated the power quality (PQ) [1] of a power distribution line because they usually inject harmonics and produce high switching frequencies. Other examples of PQ problems are transient spike [2] due to lightning and high power switching supply, high power switching power supply, voltage sag/swell [3] due to large motor operations. The waveforms associated with these PQ problems are shown in Fig. 1 (a) to (d).



(a)



(b)

(c)

(d)

Fig .1 Examples of AC waveforms with PQ problems

Poor power quality can cause malfunctions in sensitive devices or equipments such as those found in high tech industries. Moreover most of modern equipments have become the major sources of the degradation of PQ. For example, power converters and variable speed drives produce distortions in the supply current and voltage waveforms. This situation arises in every place including consumer houses as shown in Fig. 2 where consumer equipments such as washing machine, uninterruptible power supply (UPS), air conditioning, etc. are used.

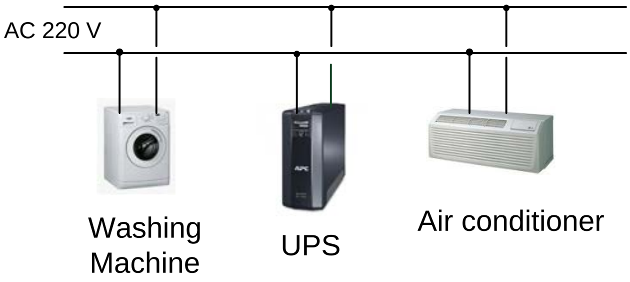


Fig. 2 Power quality problem due to domestic appliances

It is therefore important to monitor PQ in real time because this can help the utilities as well as industries to detect and correct power quality problem quickly. The equipment for this purpose is called a PQ meter. Parameters measured by a typical PQ meter are

(a) Active and reactive energy / power usages
(b) RMS values of voltage / current and its power factor
(c) Line frequency
(d) Harmonic contents of voltage / current
(e) Abnormalities, e.g. sag, swell, flicker, spike, blackout

Although there commercial PQ meters [4] available, they are mostly designed for industrial uses at a relatively cost for household applications. Hence it is too costly to install this PQ meters on a large scale. As part of an on-going research project in smart grid [5], the embedded system laboratory of Chulalongkorn University plans to design and construct a prototype smart meter [6] with PQ measurement capability. Two platforms of the project will be described in this paper, one for measuring the PQ parameters (Section II) and the other for generating the test waveforms (Section III).

II. A DSP BASED PLATFORM FOR PQ MEASUREMENT

Fig. 3 shows the connection and internal blocks of the PQ meter being developed. As the most computational demand of the meter is in the calculation of the harmonic contents in the voltage and current waveforms using Fast Fourier Transform (FFT) [7], a powerful digital signal processor (DSP) is selected as opposed to conventional microcontrollers (MCU).

To save effort time, the DSP development board TMS320C5515 [8], shown as the top board of Fig. 4, is selected as the main platform. This DSP development board consists of several components that are suitable to implement the PQ meter. They are

a) A low power 16-bit fixed point DSP running at 100MHz. It has an internal 128KB ROM and 192KB RAM.

b) An FFT accelerator for FFT calculation built-in the DSP chip.

c) 128MB RAM, 16MB flash and 256KB EEPROM on board memory

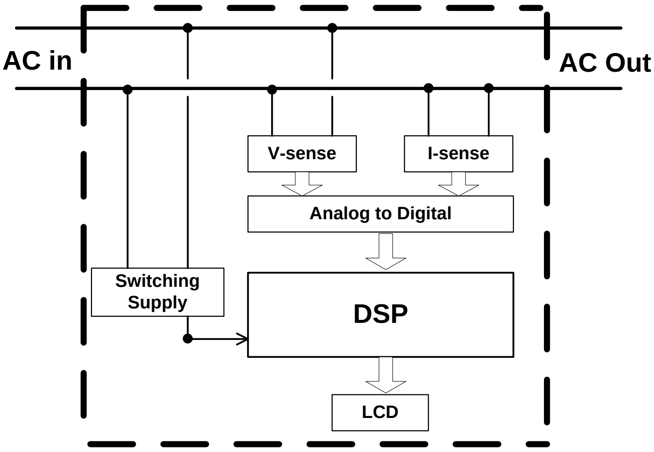


Fig. 3 The internal blocks of a PQ meter

d) 128 x 128 map color LCD display
e) USB 2.0, RS232, SPI (Serial Peripheral Interface)
f) JTAG for program downloading and debugging
g) MMC/SD connector for additional external storage

Fig. 4 The DSP and ADC boards for developing a PQ meter

The bottom board of Fig. 4 performs analog to digital conversion (ADC) of voltage and current waveforms obtained from the voltage and current sensors using transformers. It employs ADC 8556 chip [9] which has 6 analog inputs (of which two channels are used), a maximum sampling rate of 250 kHz and 16 bit resolution. Digitized data are multiplexed and serially read through the SPI port of the DSP. The full sampling rate is used to capture transient waveform as short as 4uS. However, the DSP is programmed to down sample it to 10KSamples/Sec before applying 1000 points (5 cycles) FFT algorithm to yield up to 30th harmonic components.

III. A Platform for Arbitrary Waveform Generator

Since it is difficult to find various conditions for testing a PQ meter in the real situation, an instrumental setup or test bench as shown in Fig. 5 is built up in the laboratory.

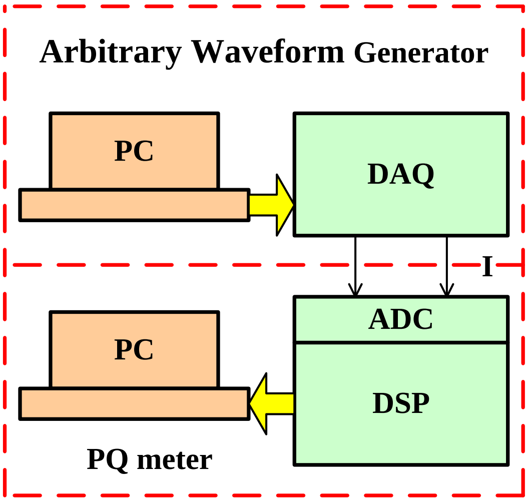


Fig. 5 Test bench for testing a PQ meter

The test bench utilizes a signal generator for generating waveforms that simulate voltage and current needed to test a PQ meter under various situations such as harmonics and spike. Such a waveform must be easily programmable and may have a period much longer than 20mS to cover abnormally like sag and flicker [10]. Hence the test bench needs an arbitrary waveform generator (AWG) with large memory for storing sampled data. In this work, the AWG should be able to generate a spike with as short duration as 4uS and a waveform with period as long as 100S to cover sag and flicker conditions. With 16 bits (2 Bytes) per sample, the total memory depth or the AWG must be 2x100S/0.4uS or 500MB. Such a large amount of memory is hard to find in a conventional AWG [11] which is also expensive. Therefore a low cost with high flexible AWG is developed for this PQ test bench. It uses a high speed data acquisition (DAQ) module NI USB 6341 [12] with high speed USB connection with a personal computer as shown in Fig. 6 This DAQ module has the following specifications

a) 16 bit resolution/sample

b) Two analog outputs

c) +/- 10Volt maximum

d) 900 KSample/Second

e) USB 2.0 with high speed (450Mbit/Sec) data transfer

f) Bus powered (USB) with no need for extra DC supply

Fig. 6 A DAQ based arbitrary waveform generator

Using LABVIEW [13] graphical programming, waveforms of arbitrary shape can be created using digitized data or basic waveforms operations and stored in the memory of the personal computer. Using 2GB or more, an arbitrary waveform of 1000 sec duration with as small as 1uS time step can be synthesized. These data are then transferred to USB 6341 to generate the waveform with repetition. Let t_i ; i =0, 1, 2,… denote the time point, the following examples show how various waveforms can be generated

a) $V_1 \sin(100\pi t_i)$ for a 50Hz sinusoidal waveform .

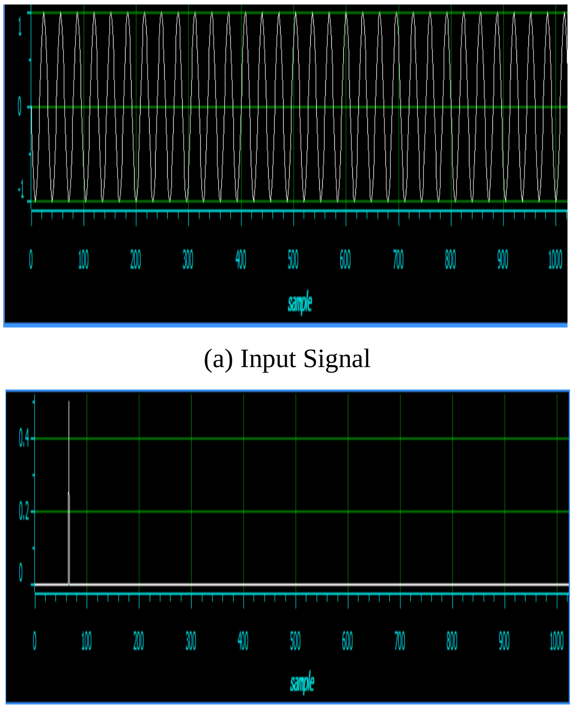
b) $V_1 \sin(100\pi t_i)$ + $V_2 \sin(200\pi t_i)$ for a 50Hz sinusoidal waveform of amplitude V_1 with 2nd harmonic of amplitude V_2

c) $\left(\frac{1+2t_i^2}{1+t_i^2}\right)\sin\left(100\pi t_i\right)$ for a 50 Hz sinusoidal waveform with amplitude of 2Volt but has a 1V sag at $t=0$

d) $\sin(100\pi t_i)+\delta(t-T)$ for a 50Hz sinusoidal waveform with an impulse at time T.

IV EXPERIMENTAL RESULTS AND CONCLUSION

The design and construction of a PQ meter requires knowledge in several areas. They include embedded system technology, algorithm and software development. A DSP platform is required to construct a prototype meter and another DAQ platform is used to set up waveforms for testing it. Measured PQ dara can be stored in the memory within the board which is large enough to hold for a long time. However when this memory is full, additional memory can be inserted on the board. Future work includes adding 2.4 GHz RF Zigbee [14] or Power line communication (PLC) [15] modules for remote reading. Some experimental results of power line waveforms are depicted based on DSP processor (TMS320C5515 EVM) as shown in Fig.7.



(a) Input Signal

(b) Frequency Spectrum

Fig.7 Sample Result

REFERENCES

[1] IEEE std 1159-1995 , IEEE Recommended Practice for Monitoring Electric Power Quality, 1995.

[2] Patel H.K., "Voltage transient spikes suppression in flyback converter using dissipative voltage snubbers", 3rd conference on Industrial Electronics and Applications, 3-5 June 2008.

[3] Hardi, S.; Daut, I.; "Sensitivity of low voltage consumer equipment to voltage sags", 4th International Power Engineering and Optimization Conference (PEOCO), 23-24 June 2010.

[4] Fluke 1750 Three-Phase Power Quality Recorder, www.fluke.com

[5] Louie, H.; Burns, M.; Lima, C.; "An introduction and user's guide to the IEEE Smart Grid Web Portal", IEEE PES Innovative Smart Grid Technologies Conference Europe (ISGT Europe), 11-13 October 2010.

[6] Depuru, S.S.S.R.; Lingfeng Wang; Devabhaktuni, V.; Gudi, N.; "Smart meters for power grid — Challenges, issues, advantages and status", IEEE/PES Power Systems Conference and Exposition (PSCE), 20-23 March 2011.

[7] Brenner, N.; Rader, C. (1976). "A New Principle for Fast Fourier Transformation". IEEE Acoustics, Speech & Signal Processing 24 (3): 264–266.

[8] TMS320C5515 Fixed-Point Digital Signal Processor, www.focus.ti.com.

[9] ADC 8556, 16-Bit, Six-Channel, Simultaneous Sampling ANALOG-TO-DIGITAL CONVERTERS, www.focus.ti.com.

[10] Yang, X.X.; Kratz, M.; "Power System Flicker Analysis by RMS Voltage Values and Numeric Flicker Meter Emulation", IEEE Transactions on Power Delivery, July 2009.

[11] 33220A Function / Arbitrary Waveform Generator, www.agilent.com.

[12] NI USB-6341 X Series High Speed Data Acquisition, http://sine.ni.com.

[13] "What is NI LABVIEW", http://www.ni.com/labview.

[14] www.zigbee.org.

[15] Hui Xiao; Xuebin Wu; "Design of Medium Voltage Power Line Spread Spectrum Carrier Communication System Based on Embedded System", 5th International Conference on Wireless Communications, Networking and Mobile Computing, (WiCom '09) 24-26 September 2009.

VITAE

Ms.Theingi Zin was born on August 4, 1985 in Kawlin, Myanmar. She was awarded a Bachelor degree of Electronic Engineering at the Department of Electronic Engineering, Mandalay Technological University in 2007. She has worked as an Assistant Lecturer in Technological University (Magway) since 2007.She has been a graduate student in the Master’s Degree Program in Electrical Engineering at the Department of Electrical Engineering, Chulalongkorn University since 2010.