BIT RESOLUTION IMPROVEMENT FOR CONTINUOUS DATA ACQUISITION OF ELECTRICAL WAVEFORMS IN MULTIPHASE ENERGY MEASUREMENT SYSTEMS

A Thesis

by

GANG LI

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

December 2010

Major Subject: Mechanical Engineering

BIT RESOLUTION IMPROVEMENT FOR CONTINUOUS DATA ACQUISITION OF ELECTRICAL WAVEFORMS IN MULTIPHASE ENERGY MEASUREMENT SYSTEMS

A Thesis

by

GANG LI

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Approved by:

Chair of Committee,	Alexander Parlos
Committee Members,	Won-Jong Kim
	Edgar Sanchez-Sinencio
Head of Department,	Dennis O'Neal

December 2010

Major Subject: Mechanical Engineering

ABSTRACT

Bit Resolution Improvement for Continuous Data Acquisition of Electrical Waveforms in Multiphase Energy Measurement Systems. (December 2010)

Gang Li, B.S., Tsinghua University

Chair of Advisory Committee: Dr. Alexander G. Parlos

A data acquisition platform has been previously developed that enables extraction of information about an electrical machine's health and energy conversion efficiency by monitoring its electrical signals. It is desired to extend the functionality of this platform to enable multi-phase signal acquisition with higher bit resolution while providing continuous waveform sampling that does not present any gaps in information acquisition.

The previously designed platform samples signals at 8 kHz and can achieve about 15 bits resolution following down sampling to 1920 Hz. To get higher bit resolution without altering the hardware, more oversampling is required. Thus, the hardware is set to sample at 64 kHz which is the maximum sampling frequency the 6-channel simultaneous sampling analog to digital converter (ADC) can provide.

Sampling at higher frequencies results in larger raw data sizes while the sampling time window remains same, and this increases the data transfer time reducing the information available for analysis in a given period of time. To address this issue, in this thesis, several proposed approaches are explored and a final hybrid solution is used. The final solution achieves about 1.3 bits improvement in signal resolution compared to the original firmware, while also performing continuous waveform acquisition if the end-to-end network delay is within expected ranges.

To My Parents

ACKNOWLEDGEMENTS

I would like to express my appreciation to my committee chair and advisor Dr. Alexander G. Parlos. Without his guidance, technical support and advice, this work would not have been possible. I would also like to thank other members of my committee: Dr. Won-Jong Kim and Dr. Edgar Sanchez-Sinencio, for their comments and advice.

Thanks also go to Mr. Jianxi Fu, who provided me a lot of thoughts and assistance in my research. I also want to extend my gratitude to my friends Mr. Pang-Chun Chu, Mr. Yong Li and Mr. Tengxi Wang at Texas A&M University for their contributions.

I also wish to thank Dr. Sujie Guo in P.R. China for her continuous encouragement and patience.

Finally, I would like to convey a sincere thanks to my parents for their continued support in all that I do.

TABLE OF CONTENTS

CHAPTER Page

Page

LIST OF FIGURES

FIGURE Page

FIGURE Page

LIST OF TABLES

TABLE Page

CHAPTER I

INTRODUCTION

A. Motivation

In industry, unplanned shutdown will generally result in large unpredictable financial loss compared to planned maintenance service. To predict the equipment failure for avoiding unplanned shutdown, algorithms has been developed to have machine's health monitored and indicate the trend of the machine’s deterioration [1]. It is possible to acquire much of the required equipment condition information from the electrical signal waveforms of the equipments. Compared to using accelerometers and any other sensory devices, electrical information acquisition is normally less expensive because only the low voltage side of the power cables which supplies the equipment is needed to be accessed.

If the electrical information is properly collected, energy consumption and equipment energy efficiency can also be estimated based on the actual voltage and current waveforms and equipment rated values [2]. This can provide further analysis on equipment condition and suggestions for equipment operation and service.

In order to enable such algorithms to be executed in near real-time, a data acquisition device with both hardware and software functionality has developed, this device simultaneously receives the electrical waveforms of five or six electrical signal channels simultaneously (three currents plus two or three voltages from a 3-phase system) and transfers the information to a central location, which is generally a PC. All electrical

This thesis follows the style of *IEEE Transactions on Automatic Control.*

waveforms are analyzed at the central location to obtain power consumption, efficiency estimates, and access the equipment condition. In this thesis, the device that collects and transfers data is defined as an “endpoint”.

B. Problem Definition

To meet various analysis requirements, e.g. frequency and time resolution etc., the time length of a data set is set to 30 seconds, which means that during the 30 seconds, the waveform samples are continuous. Also according to the Nyquist criterion, to accurately digitize a signal, one must sample at a frequency of at least twice the highest frequency of interest in the signal. For analyzing the condition of equipment and estimating their efficiency, the resampling frequency was set to 1920Hz, so only signals with frequencies up to 960Hz were being investigated, and the analog to digital converter (ADC) was setup to work at its lowest sampling frequency of 8 kHz.

So originally the system was working with an approximate 4 times (8 kHz divided by 1920 Hz) oversampling rate. Based on this configuration, the signal-to-noise ratio (SNR) achieved from the power spectral density is around 98db, according to the relationship between SNR and bit resolution, about 15 bits resolution was achieved.

Although energy efficiency estimation is not very sensitive to signal bit resolution, higher signal bit resolution could enhance equipment condition assessment. One method to increase bit resolution is oversampling, and to get 1 additional bit resolution improvement, one must increase the oversampling rate by 4 times [3].

If the ADC is set to work at 64 kHz, increased bit resolution could be achieved because the overall oversampling rate will be increased from 4 to 32 times (64 kHz divided by 1920 Hz), or about 1.5 additional bits compared to 8 kHz sampling rate.

Since the ADC driver is already properly configured in the original endpoint software, setting the ADC to sample at higher frequency can be simply achieved by changing the configuration register in the ADC driver. Sampling at higher frequency, however, will result in larger data sizes when sampling duration remains the same; the data size is determined in Equation (1).

$$Data\ Size = 16\,[bit\,/\,(sample \cdot channel)] \times f_s\,[samples\,/\,\sec] \times T\,[\sec] \times 6\,[channels] \quad (1)$$

When sampled at 8 kHz, 30 seconds of six channels of 16 bit resolution results in data files of 2.88 MB. Whereas when sampled at 64 kHz, with the other variables being the same, the data size will be 23.04 MB, the larger files require more time to be transferred to the PC.

To quantify the problem, the “Data Rate” (DR) is defined as the amount of data sets received by the analysis software per hour; also the “Data Available Time” (DAT) is defined as the overall time for the software to get one data set. This time should be the sum of the sampling time (30 seconds) and transfer time from the endpoint to the analysis system. Since the length of each data set is 30 seconds, the maximum DR value without overlap between consecutive data sets is 120 sets per hour. Lower DR will require longer analysis time for equipment condition and increasing the delay of any event detection and efficiency estimation. The availability of the DR becomes a critical design requirement in the overall system design.

The problem to be addressed is to find a method to improve the bit resolution of the electrical signals subject to the maximum sampling rate limitation of the current ADC while enabling the analysis of continuous 30 seconds waveforms by a remote analysis software.

C. Literature Review

When the endpoint is initially designed, the Analog Devices AD73360 ADC was selected for three reasons [4]. The first reason is that there is a minimum channel requirement for monitoring three-phase electrical setup. In the delta configuration, two voltages and three currents should be monitored. While in another possible Y configuration, three voltages and three currents need to be measured at the same time. This indicates that the selected ADC needs to have six channels at least, which the AD73360 is capable of doing. Also in order to reduce the complexity of the digital signal processing, simultaneous sampling feature among six channels is advantageous because if sequential sampling is utilized, a phase shift introduced by time difference would be existed between different channels and needs to be calibrated. This calibration can be avoided by using six separate ADCs or a device like the AD73360. But six separate ADCs will introduce many complications in the communication with the selected processor and in the printed circuit board layout, also synchronizing six separate ADCs is required because phase shift will still be introduced if synchronization is not properly done. The AD73360 contains six A/D channels which can do sampling at the same time in a single chip and have all six channels internally synchronized, this feature greatly reduces the complexity in the whole system design.

The second reason is sampling rate requirement, in this electrical signal waveform sampling application, since a minimum 1920 Hz sampling rate is required. The AD73360 with multiple sampling rates of 8/16/32/64 kHz is more than adequate for this system. The third requirement of the ADC was its resolution, for equipment condition monitoring, the signal information required for the assessment of equipment condition is contained in the harmonics of the signal, so significant bit resolution is necessary. This resolution has been determined previous to be at least 14 bits. The AD73360 has a minimum 12 bit SNR. While this minimum is below what is required, it is possible to meet the requirement of at least 14 bits of effective resolution with sufficient signal conditioning in the following procedure [5].

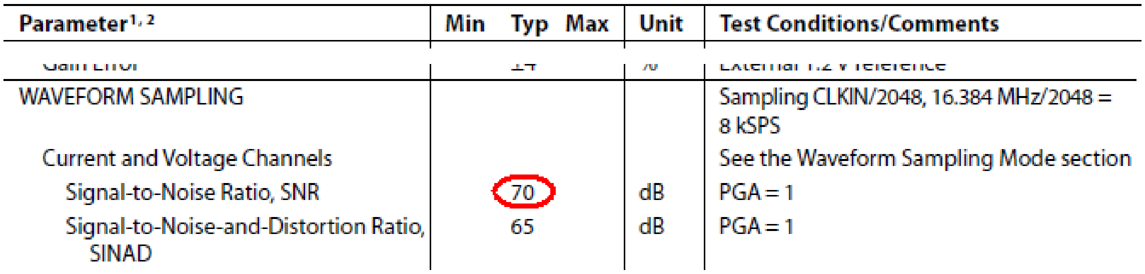
For the digital signal processor (DSP) selection, the Analog Device Blackfin BF537 fixed point DSP was chosen for two reasons [4]. The first reason is the communication with AD73360. BF537 has a full-duplex, synchronous serial interface which is easy to be configured to link with AD73360. The second reason is the communication with PC (central location). Since BF537 has a 10/100 Ethernet MAC, it will greatly simplify the system development process because Ethernet between endpoint and PC can be established by only adding an Ethernet physical layer interface to the DSP [6].

Analog Device recently released several new chips which include Polyphase Multifunction Energy Metering IC ADE7878 and simultaneous sampling bipolar 16bit ADC AD7656. The ADE7878 is highly accuracy, 3-phase electrical energy measurement IC with serial interfaces and three flexible pulse outputs. The ADE7878 device can perform total (fundamental and harmonic) active, reactive and apparent energy measurement and RMS calculations, as well as fundamental-only active and reactive energy measurement and RMS calculations. ADE7878 also features a waveform sampling mode which can provide waveform samples of the current and voltage waveform, the active, reactive, and apparent power outputs and all these samples are stored every 125 microseconds (8 kHz rate) into 24-bit signed registers that can be accessed through various serial ports of the ADE7878 [7].

The AD7656 is a 250 kSPS, 16-bit, 6-Channel, simultaneous sampling, bipolar, low power consumption, successive approximation ADC, which contains all components for use in a multichannel simultaneous power line monitoring system; it uses a high speed serial interface for writing to the configuration register and receiving converted digital results [8].

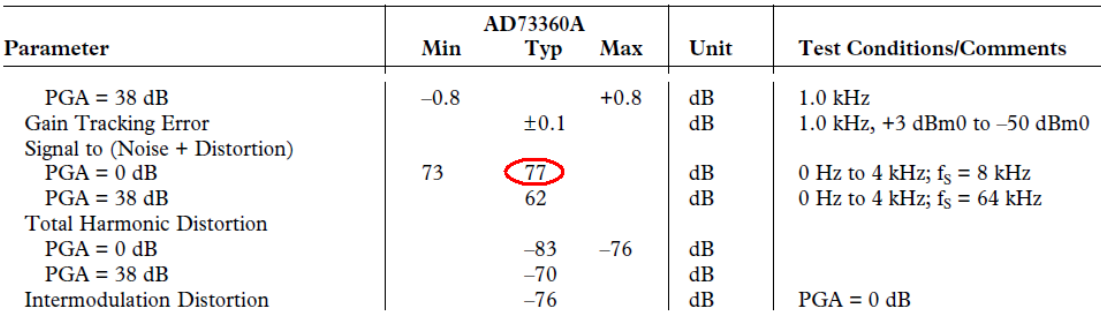
Compared to AD73360, ADE7878 features energy measurement functions which can simplify the software design. The only concern is the SNR; from the data sheets, the waveform sampling SNR provided by ADE7878 (Table 1) is lower than AD73360 (Table 2). The AD7656 features higher SNR (Table 3) and higher sampling rate, and thus it has better performance than AD73360.

Table 1. ADE7878 waveform sampling SNR



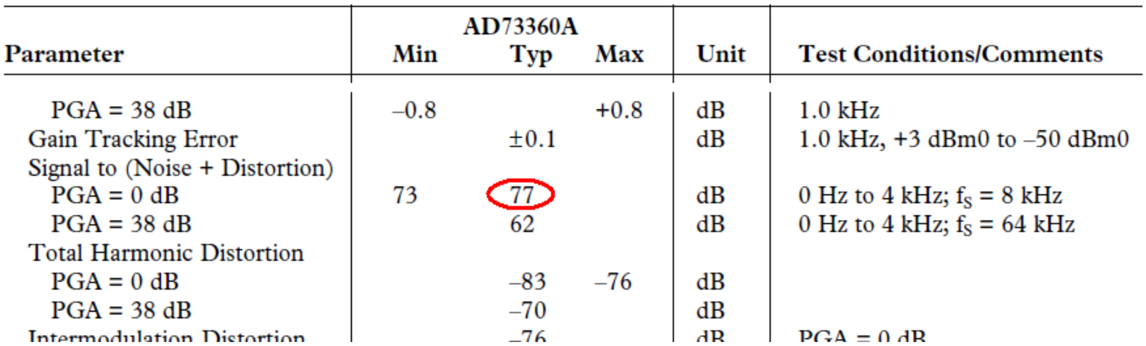
Parameter[1, 2]	Min	Typ	Max	Unit	Test Conditions/Comments
Gain Error		14		%	External 1.2 V reference
WAVEFORM SAMPLING					Sampling CLKIN/2048, 16.384 MHz/2048 = 8 kSPS
Current and Voltage Channels					See the Waveform Sampling Mode section
Signal-to-Noise Ratio, SNR		70		dB	PGA = 1
Signal-to-Noise-and-Distortion Ratio, SINAD		65		dB	PGA = 1

Table 2. AD73360 SNR



Parameter	AD73360A Min	Typ	Max	Unit	Test Conditions/Comments
PGA = 38 dB	–0.8		+0.8	dB	1.0 kHz
Gain Tracking Error		±0.1		dB	1.0 kHz, +3 dBm0 to –50 dBm0
Signal to (Noise + Distortion)					
PGA = 0 dB	73	77		dB	0 Hz to 4 kHz; f_S = 8 kHz
PGA = 38 dB		62		dB	0 Hz to 4 kHz; f_S = 64 kHz
Total Harmonic Distortion					
PGA = 0 dB		–83	–76	dB	
PGA = 38 dB		–70		dB	
Intermodulation Distortion		–76		dB	PGA = 0 dB

Table 3. AD7656 SNR



Parameter	AD73360A Min	Typ	Max	Unit	Test Conditions/Comments
PGA = 38 dB	–0.8		+0.8	dB	1.0 kHz
Gain Tracking Error		±0.1		dB	1.0 kHz, +3 dBm0 to –50 dBm0
Signal to (Noise + Distortion)					
PGA = 0 dB	73	77		dB	0 Hz to 4 kHz; f_S = 8 kHz
PGA = 38 dB		62		dB	0 Hz to 4 kHz; f_S = 64 kHz
Total Harmonic Distortion					
PGA = 0 dB		–83	–76	dB	
PGA = 38 dB		–70		dB	
Intermodulation Distortion		–76		dB	PGA = 0 dB

According to the principle of a Delta-Sigma converter [9] and oversampling theory, bit resolution can be improved by oversampling. When an analog signal is digitized by ADC, the signal is quantized into discrete levels. This will introduce some error which is often referred to be quantization error. Spreading ADC's quantization error or quantization noise over a wider bandwidth by oversampling technique can simply reduce these errors and noises. A normal averaging will only even out signal fluctuations, while decimation will improve the resolution. In a 4-times-oversampled signal, four adjacent data points are averaged to produce one new data point, this result is the same as if the ADC is sampling at one fourth of the rate, but it has the effect of averaging the quantization noise, which improves SNR, thus also increase the effective number of bits and reduce the quantization error. To lower measurement errors, faster sampling rate is required and the ADC must be more accurate than the measurement uncertainties inherent in the system. The availabilities of low cost memories and faster ADC make the advantages of oversampling become cost effective and desirable. In [10], the author had successfully got 3 bits more resolution from an 8-bit ADC by oversampling 64 times. Tian-You Yu, et al, developed a resolution enhancement technique by using range oversampling in radar application [11]. Also Atmel Corporation introduces a method to enhance the bit resolution by oversampling on its AVR controller with a 10-bit ADC [3].

Deflate is a lossless data compression algorithm that uses a combination of the LZ77 algorithm [12] and Huffman coding [13]. The first stage of the compression is duplicate string elimination, within compressed blocks, if a duplicate series of bytes is spotted (a repeated string), then a back-reference is inserted, linking to the previous location of that identical string instead. The second compression stage is bit reduction by replacing commonly-used symbols with shorter representations and less commonly used symbols with longer representations. One encoder implementation of Deflate algorithm is zlib [14], which is a standard reference implementation widely used, owing to public availability of the source code and a license allowing inclusion into other software.

There are also many other compression methods which have been investigated, Omer Nezih Gerek, et al, developed a data compression method for power quality event data by using 2D representation [15].

Since the current system is designed with a fixed point DSP, the principles of fixed point DSP have also been investigated, Boris Lerner, in his text, compares the differences between a fixed point and a float point DSP in different applications like military radar, mobile television, professional-audio-effects processor, and automatic echo canceller in a hands-free portable device [16]. Bob Pendleton introduced an algorithm to perform fixed point computation. In fixed point multiplication, the maximum number of digits in the product is the sum of the number of digits in the multiplier and the multiplicand; the number of digits of fraction in the product is the sum of the number of digits of fraction in the multiplier and the number of digits of fraction in the multiplicand; an integer is a fixed point number with zero digits of fraction [17]. Eduardo Pinheiro, et al., implemented a power line notch filter used to improve the signal-to-noise ratio in biomedical signals [18].

Many of the methods and algorithm listed above have been designed for specific applications. None of them can be directly applied to the problem at hand and address it effectively.

D. Proposed Approach

To improve bit resolution, one could replace the current ADC AD73360 with another one which has more bit resolutions. Another method to improve the bit resolution is oversampling. The AD73360 can increase the sampling rate from 8 kHz to 64 kHz, while the AD7656 has a higher sampling rate at 250 kHz. The ADE7878 is an energy measurement chip with waveform sampling function, but in its waveform sampling mode, both the bit resolution and the sampling rate are lower than AD73360. Since the ADE7878 and the AD7656 are recently released items, the decision of hardware upgrade requires further tests and verification. In this thesis, methods of improving bit resolution by upgrading the hardware are not considered.

Based on the current hardware setup, the sampling rate of the AD73360 is set to 64 kHz rather than 8 kHz. Generally, there is a relationship between DR and DAT; DR will be decreased as DAT increases, based on original endpoint software logic, the relationship can be described in Equation (2).

$$DR = \frac{3600\,[sec/\,hour]}{DAT} = \frac{3600\,[sec/\,hour]}{average(\text{Sampling Time (30 sec)} + \text{Transfer Time})} \tag{2}$$

So as the sampling rate is increased from 8 kHz to 64 kHz, the raw data size will be increased, this will result in longer transfer time. Therefore the DAT will be longer and the DR will be lower. To address this issue, following two methods are considered:

(1) The first method is compressing the raw data on the endpoint. If the data size can be reduced, the transfer time can be reduced as well. To achieve this, a lossless compression on the raw data on the endpoint is considered, and the compressed raw data are decompressed by the analysis software after transmission.

(2) The second method is pre-processing the raw data on the end point. Because of oversampling, the raw data are downsampled during signal processing by the remote analysis software. If this downsampling procedure is moved from the remote location locally within the endpoint, the transfer time can also be reduced since as downsampled data will have a much smaller overall size.

In this thesis, transfer speed and software threading logic optimization are also discussed to maximize the DR. Increasing the transfer speed and configuring data sampling and transferring threads running in parallel in the endpoint software can also reduce DAT.

All of the approaches above are based on the observation that in the current endpoint software, the DSP utilization is very low. As a result, the performance of the data acquisition system could be increased if more tasks are transferred to the endpoint.

E. Contributions

Based on the current hardware setup, the main anticipated contribution of this research is to provide nine db (one and half bit resolutions) improvement in SNR, and achieve time domain continuity between consecutive data sets (maximum DR).

F. Thesis Organization

This thesis starts with the overview of multi-phase energy measurement system and introduces the experimental setups. Following the verification of bit resolution improvement by oversampling, the proposed approaches are explored. Next, the software optimizations are performed to maximize the performance. The feature of continuous waveform sampling with bit resolution improvement is developed and verified finally.

CHAPTER II

MULTI-PHASE ENERGY MEASUREMENT SYSTEM AND EXPERIMENTAL SETUP

A. System Overview

1. Hardware

The previously designed data acquisition system is composed of several blocks, as shown below in Figure 1.

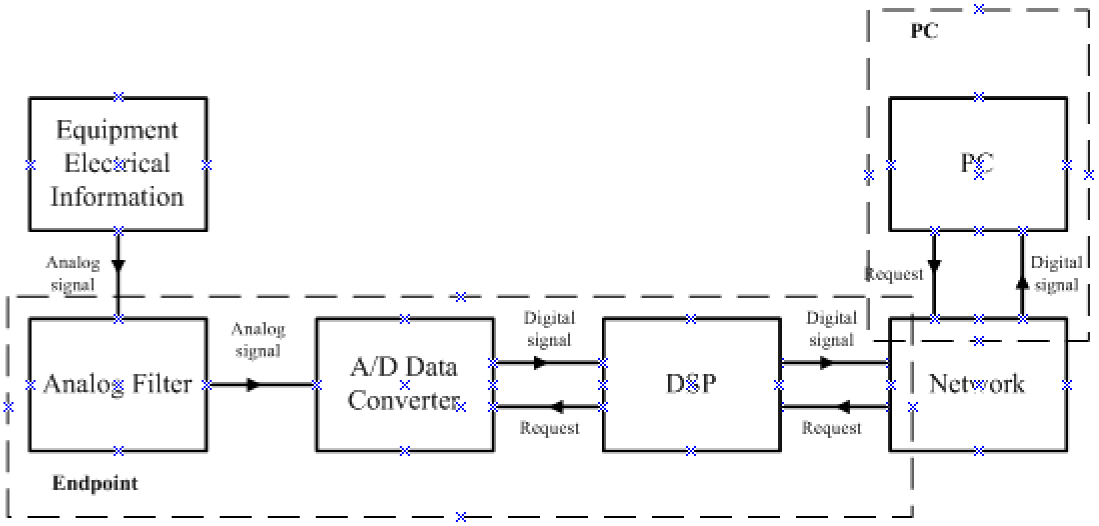


Figure 1. Overall system mechanism

The raw power signals are passed through analog filters and then sampled by an analog to digital converter which passes the data on to a digital signal processor. This processor then needs to transmit the data to a central PC. This central PC will then receive, store, and process the data, yielding the needed information regarding the power equipment's health.

Each of the endpoints is composed of several modules. For DSP module, Analog Device fixed point DSP BF537 which has a clock speed of 600 MHz and a 10/100 Ethernet MAC has been chosen. And Analog Device AD73360 which is a six channels 16 bit A/D converter with a minimum 12 bit SNR and 8K/16K/32K/64K sampling frequency is selected in the A/D Data Converter module. For the Network part, the endpoint can be configured in either wired 10/100 Ethernet connection or wireless connection by 10/100 Ethernet through an 802.11b/g wireless bridge (Quatech WLNG-ET-DP101 [19]). Each of the three voltage and three current channels contain properly designed analog front end which can avoid the aliasing when the analog signal is being sampled to digital, also the currents are sensed by using external shunted current transformers (CT). Figure 2 and Figure 3 show the real endpoint hardware.

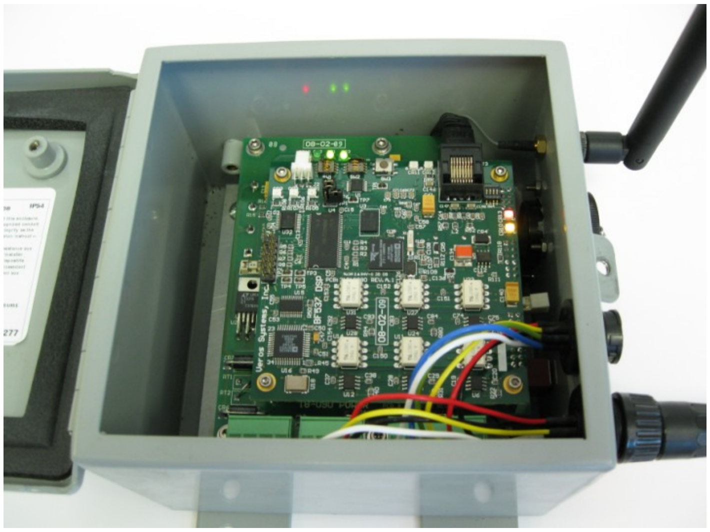


Figure 2. Endpoint with an enclosure

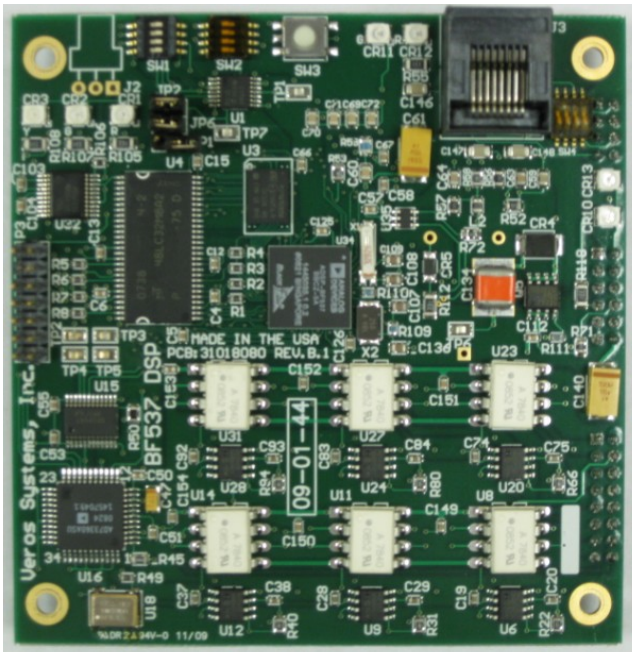


Figure 3. Endpoint PCB with Blackfin-BF537 and AD73360

2. Software

a. Endpoint Software

The endpoint software is built around the Analog Devices VDK kernel. This is a multi-threaded kernel that provides the necessary support for the Ethernet interface. This support comes in the form of driver software and a TCP/IP stack.

The endpoint software is programmed and compiled in VisualDSP++4.0 environment and loaded to the endpoint through an emulator.

b. PC Software

The PC software is comprised of several modules. The first module handles the polling of the endpoints for data. A second module performs the actual processing of the data to determine the condition of the equipment being monitored. A third module that resides either in the processing server or another PC makes the equipment condition available through a web interface.

In this thesis, only the endpoint software, the first module and part of the second module of the PC software are concerned. The previously designed endpoint software and PC software described above are mentioned as “original software” in the rest of this thesis.

B. Experimental Setups

Two test beds are established to provide analog electrical power from which the endpoint can sample. The first one is a fan motor powered by 3-phase 208V 60Hz AC, an open-delta 208V to 120V potential transformer and three 5A to 333mV shunted current transformers are also utilized to provide proper input to the endpoint; the second one is pure resistor voltage dividing circuit powered by single-phase 120V 60Hz AC, three voltage channels are directly connected to the 120V and three current channels are connected to the output of the voltage divider.

All tests related to signal quality like bit resolution and compression have been done on the fan motor test bed, since the fan motor test bed can provide 3 phases signal which is similar to the reality; other tests like transfer speed, software threads configuration and software stability are done on the pure resistor test bed because the actual data content is not important.

Figure 4 and Figure 5 show the actual test beds of the fan motor and the pure resistors. The circuit diagram in Figure 6 shows the principle of the setups.

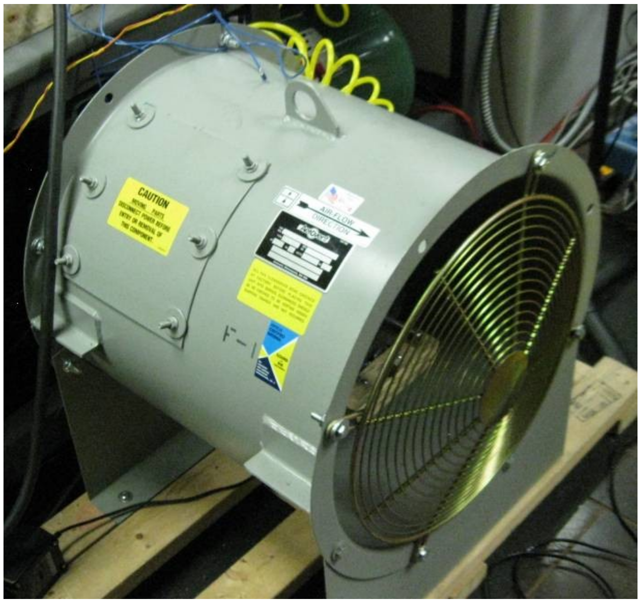


Figure 4. Fan motor test bed

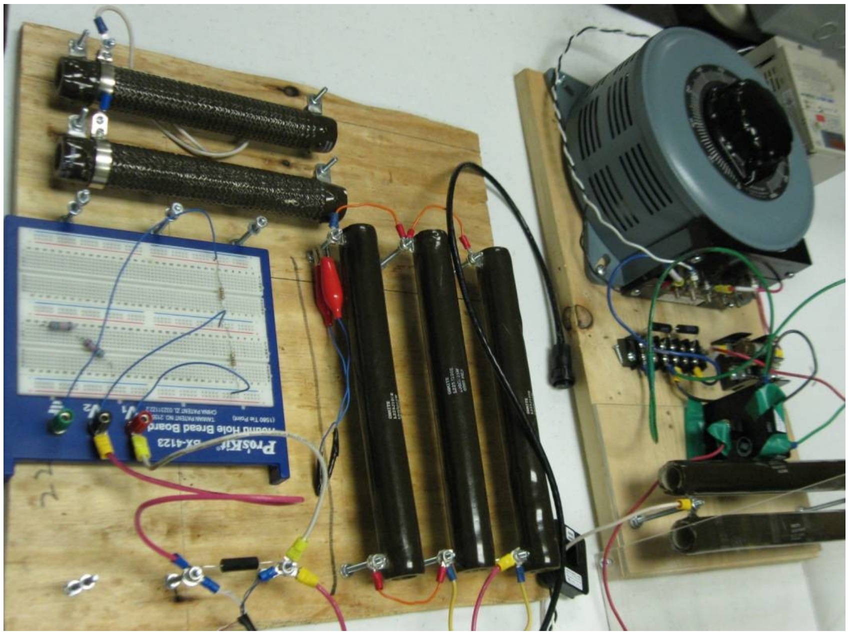


Figure 5. Resistor test bed

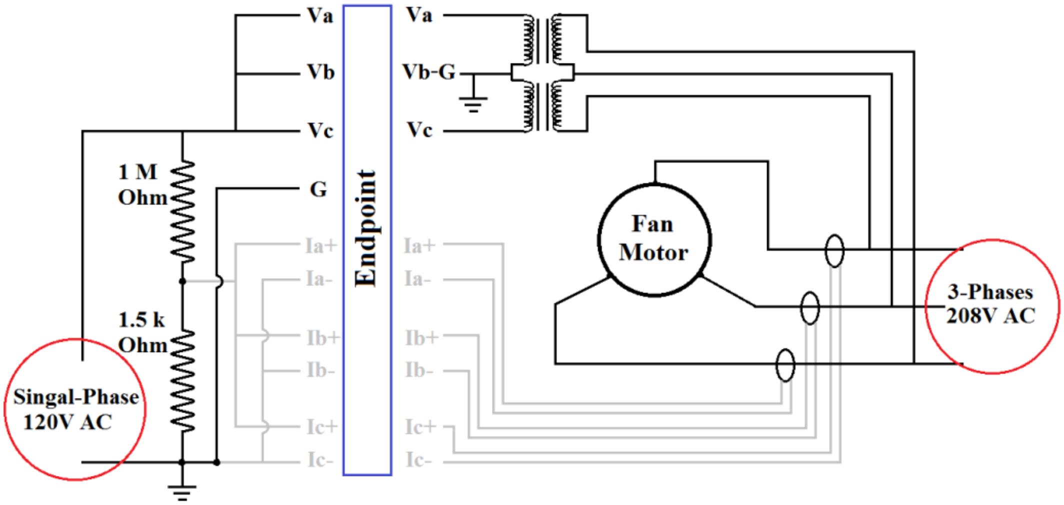


Figure 6. Circuit diagram

C. Chapter Summary

In this chapter, the previously designed data acquisition and energy measurement system is introduced. Both hardware and original software are reviewed. The experimental setups which provide electrical signal to the endpoint in this research are described finally.

CHAPTER III

PROPOSED SOLUTIONS

A. Oversampling for Bit Resolution Improvement

To get 1 bit resolution improvement, 4 times oversampling is required, so theoretical speaking, 1.5 bit resolution should be achieved when there is an 8 times oversampling from 8 kHz to 64 kHz. Before trying the proposed approaches in chapter I to address the problem, bit resolution improvement by oversampling needs to be verified. In this section, both 8 kHz and 64 kHz raw data are collected on endpoint, transferred to PC and analyzed on PC, the power density spectrum based on 8 kHz and 64 kHz raw data are compared to verify the SNR improvement.

1. Raw Data Acquisition

The A/D converter can not sample at 8 kHz and 64 kHz at the same time, since the power quality will change as time varies and electrical noise will be different in different endpoints, the best way to get the comparable data sets is to set one single endpoint working in the following sequence shown in Figure 7.

Initially, the A/D converter is set to 64 kHz; one 64 kHz raw data will be collected and transferred to PC; then the A/D converter will be set to 8 kHz to get a set of 8 kHz raw data; this procedure is repeated for 30 times.

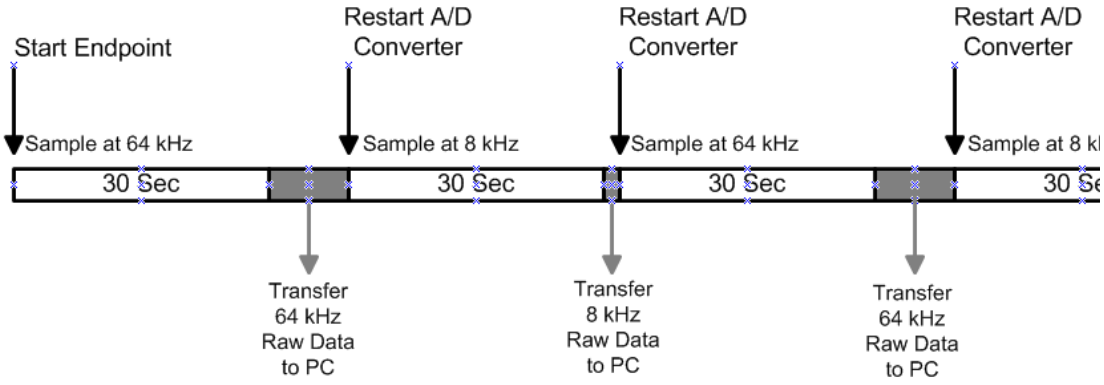


Figure 7. Timeline swapping 8000 and 64000

2. Bit Resolution Comparison

In order to get significant comparison, all data sets should go through the same data process procedure to get SNR; Figure 8 shows the standard down sample procedure in the original PC software. 8 kHz raw data from endpoint can go through this procedure directly, while 64 kHz raw data need to be pre-processed to 8 kHz data.

Two methods are developed to pre-process 64 kHz raw data, the first one is a standard down sampling method, from which the bit resolution improvement is expected to be achieved; the second one is a down sampling procedure without anti-aliasing low-pass filter, the bit resolution of the 8 kHz data from this procedure should be equal to the 8 kHz raw data directly sampled at endpoint.

After the SNR of every data set are computed from the power density spectrum. Two comparisons are made afterwards; the first comparison is between 8 kHz data down sampled from 64 kHz raw data and 8 kHz data "picked up" form 64 kHz raw data; the second comparison is between 8 kHz data down sampled from 64 kHz raw data and 8 kHz raw data from the endpoint. Figure 9 shows the mechanisms of the data processing procedure and comparison.

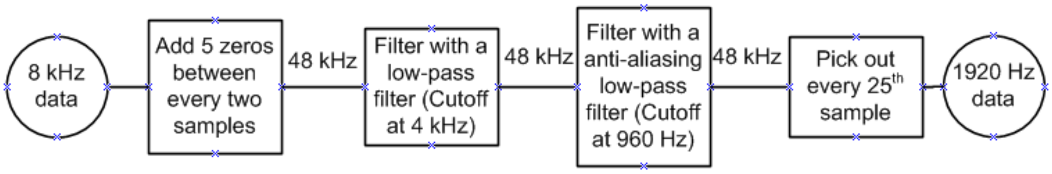

8 kHz data
Add 5 zeros between every two samples
48 kHz
Filter with a low-pass filter (Cutoff at 4 kHz)
48 kHz
Filter with a anti-aliasing low-pass filter (Cutoff at 960 Hz)
48 kHz
Pick out every 25th sample
1920 Hz data

Figure 8. 8000 to 1920 down sample

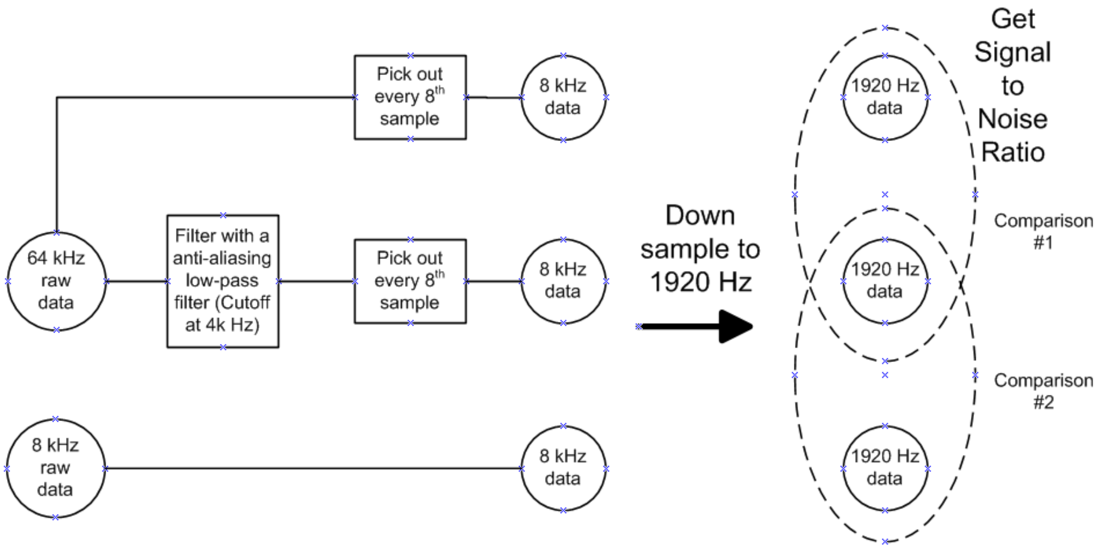


Figure 9. Comparisons #1 and #2

B. Oversampling and Compression

Compression on the endpoint is considered to reduce the transfer size, but in order to decompress the data and recover all the information on PC, a compression method with lossless feature is required. DEFLATE lossless data compression algorithm is considered in this approach.

1. Software Implement

One encoder implementation of Deflate algorithm is zlib, which is a standard reference implementation used in a huge amount of software, owing to public availability of the source code and a license allowing inclusion into other software. Source code of compression function can be obtained from zlib; it can be simply imported to endpoint software since it is written in C. Figure 10 shows the mechanism of the testing version of endpoint software with compression. If this compression method is acceptable, the actual version will not transfer raw data.

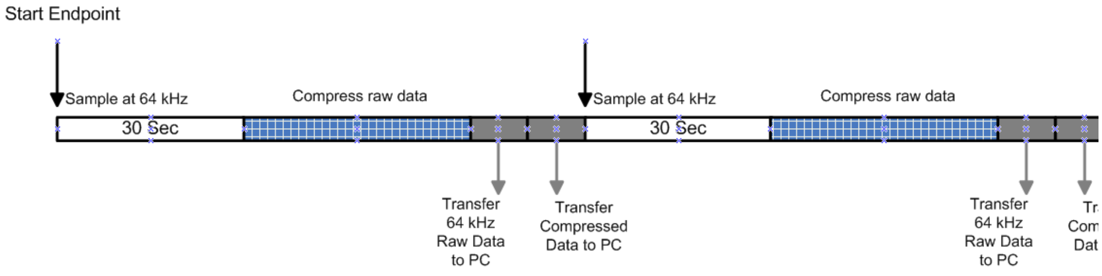


Figure 10. Timeline of compression

2. Performance

Raw data are compressed on the endpoint and transferred to PC; the compression time recorded on the endpoint is transferred to PC too to benchmark. To verify the compression performance, a copy of raw data without compression is also transferred to PC; this copy of data will be used to be compared to the decompressed raw data to make sure there is no loss during compression and decompression. This raw data copy will also be compressed on PC by a compression function which is same as the compression function on the endpoint, the compression ratios achieved by PC and by endpoint are compared to verify the endpoint software compression mechanism.

Compression ratio in this thesis is defined as Equation (3), so the higher the compression ratio is, the better the compression is. Based on the test results shown in Table 4 and Table 5, although the DEFLATE compression is confirmed to be lossless, the compression rate is too low to make significant reduction in data size, and also the time consumption for compressing on the endpoint is too long. Since the “DAT” in this case will be the sum of sampling time, compressing time and compressed data transfer time (Equation (4)), the Deflate compression algorithm requires a lot of compressing time but can only reduce the transfer time a little bit, so it is not an acceptable solution for increasing “DR”.

$$\mathit{Compression\ Ratio} = \frac{\mathit{Original\ Size}}{\mathit{Compressed\ Size}} \tag{3}$$

$$DR = \frac{3600\,[\mathrm{sec}/\,\mathit{hour}]}{\mathit{average}(\text{Sampling Time (30 sec)} + \text{Compressing Time} + \text{Transfer Time})} \tag{4}$$

Table 4. Real raw data compression

	Endpoint	PC
Original Size (Bytes)	23040000	23040000
Compressed Size (Bytes)	22275728	22275728
Execute Time (Seconds)	200.248	4.265
Decompressed Size (Bytes)	N/A	23040000
Compression Ratio	1.0343	1.0343

Table 5. All “1” data compression

	Endpoint	PC
Original Size (Bytes)	21000000	21000000
Compressed Size (Bytes)	20422	20422
Execute Time (Seconds)	47.393	0.859
Compression Ratio	1028.3	1028.3

C. Oversampling and Preprocessing

Since the benefit of the bit resolution improvement is achieved by oversampling and down sampling, it may be feasible to implement some down sampling function in the endpoint software; the goal of this is not to reduce the PC software burden but to reduce the transfer time since the down sampled data will have a much smaller size.

1. Down Sample Rate Selection

Integer down sample factor is the basic requirement for the endpoint down sampling function because the fractional down sampling will be composited of an integer up sampling step and an integer down sampling step, which will require more computation and memory.

The closest down sample rate to 1920 Hz with an integer down sample factor is 2 kHz. But since 1920 Hz down sampled rate is strictly designed for the signal with 60Hz fundamental frequency, the final down sampling rate may varies if the fundamental frequency changes, and the harmonics with higher frequency (larger than 960 Hz) might also be considered in the future, 4 kHz down sampling rate is selected for endpoint software.

2. Filter Design

A standard down sampling procedure with an original sampling frequency 64 kHz and a down sampling factor 16 contains two steps: 1. Filter the signal to ensure that the sampling theorem is satisfied. This filter should, theoretically, be the sinc filter with frequency cutoff at 2 kHz. 2. Reduce the data by picking out every 16th sample in the filtered 64 kHz signal, data rate is reduced to 4 kHz in this step.

An 5th order IIR anti aliasing low pass filter which can meet the requirement with much lower order than FIR is considered. The filter is designed in MATLAB on PC and embedded into endpoint software.

3. Software Implement

Since BF537 is a fixed point 16-bit DSP, to run the IIR filter on BF537, the filter coefficients designed in the previous steps are converted to fixed point number and a down sampling function with fixed point computation is implemented on the endpoint software. Table 6 shows the filter coefficients conversion.

To keep enough information, every filtered sample will be converted to a 24-bit variable, 16-bit for integer and 8-bit for decimal (the size is equal to 3 “char”s). The endpoint software is configured with one 64 kHz raw data buffer, one 64 kHz 24-bit filtered data buffer and one 4 kHz 24-bit down sampled data buffer, the testing version of endpoint software mechanism and timeline are shown in the Figure 11. If data preprocessing method is acceptable, the actual version will not transfer 64 kHz raw data.

Table 6. Filter coefficients

Filter	a (10^{-05})	B	Filter	a	b
1	3.85840	1.00000	Integer	2	2048
			Decimal	34645	0
2	-2.22673	-4.46032	Integer	-1	-9134
			Decimal	30101	47546
3	3.42845	7.98406	Integer	2	16351
			Decimal	16179	23257
4	3.42845	-7.16746	Integer	2	-14678
			Decimal	16179	63363
5	-2.22673	3.22620	Integer	-1	6607
			Decimal	30101	16942
6	3.85840	-0.58238	Integer	2	-1192
			Decimal	34645	46779

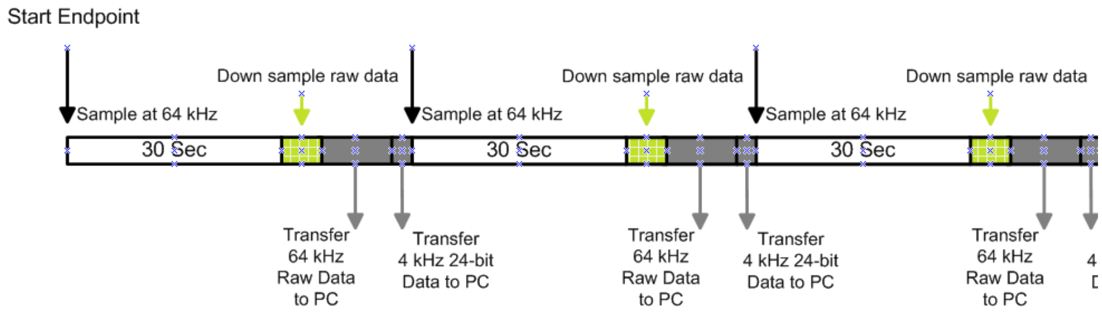

Start Endpoint
Sample at 64 kHz
Down sample raw data
30 Sec
Transfer 64 kHz Raw Data to PC
Transfer 4 kHz 24-bit Data to PC
Sample at 64 kHz
Down sample raw data
30 Sec
Transfer 64 kHz Raw Data to PC
Transfer 4 kHz 24-bit Data to PC
Sample at 64 kHz
Down sample raw data
30 Sec
Transfer 64 kHz Raw Data to PC
4
D

Figure 11. Timeline down sample

Since the data received by PC will be 4 kHz, the data processing procedure on PC will also needs to be changed correspondingly, which can be represented in Figure 12:

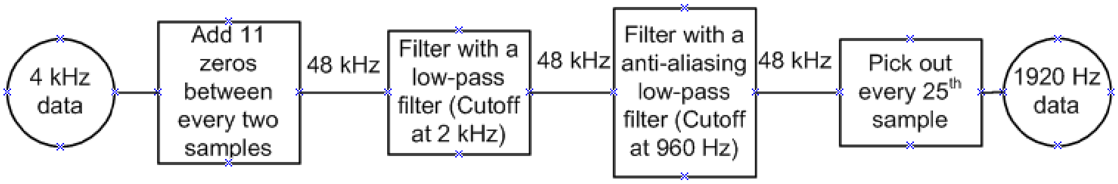

4 kHz data
Add 11 zeros between every two samples
48 kHz
Filter with a low-pass filter (Cutoff at 2 kHz)
48 kHz
Filter with a anti-aliasing low-pass filter (Cutoff at 960 Hz)
48 kHz
Pick out every 25th sample
1920 Hz data

Figure 12. 4000 to 1920 down sample

4. Bit Resolution Verification

To verified the endpoint software filter performance (time consumption and accuracy) and bit resolution improvement, both 64 kHz raw data and 8 kHz raw data are sampled on the endpoint; all 64 kHz raw data, 4 kHz down sampled data and 8 kHz raw data are transferred from the endpoint to PC; some procedures which are similar with the verification in Section II.A are performed; also a down sample function which is the same as the endpoint down sampling procedure is programmed on PC (with fixed point arithmetic). The overall mechanism and timeline are described in Figure 13 and Figure 14.

Time cost by down sampling is recorded and transferred to PC to benchmark, and transfer timestamps are recorded on PC to further verify the time cost by endpoint down sampling function. Several comparisons are made, #1 (A and B) compares the down sampling function on the endpoint and on PC to verify that the endpoint down sampling function can work properly; #2 (D and E) tries to verify that as long as the system samples at 64 kHz and down samples to 1920Hz, the bit resolution will not changes very much even the sub steps in between are different, #3 (D and C, D and G) tries to verify that the new software system can achieve the bit resolution improvement compared to the original software system.

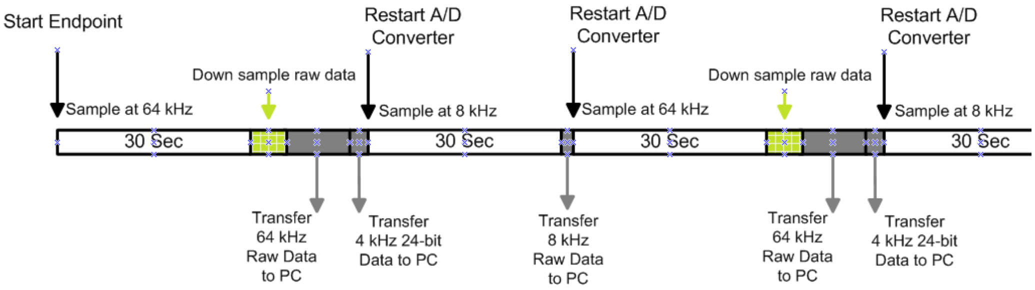


Figure 13. Timeline of software for verification

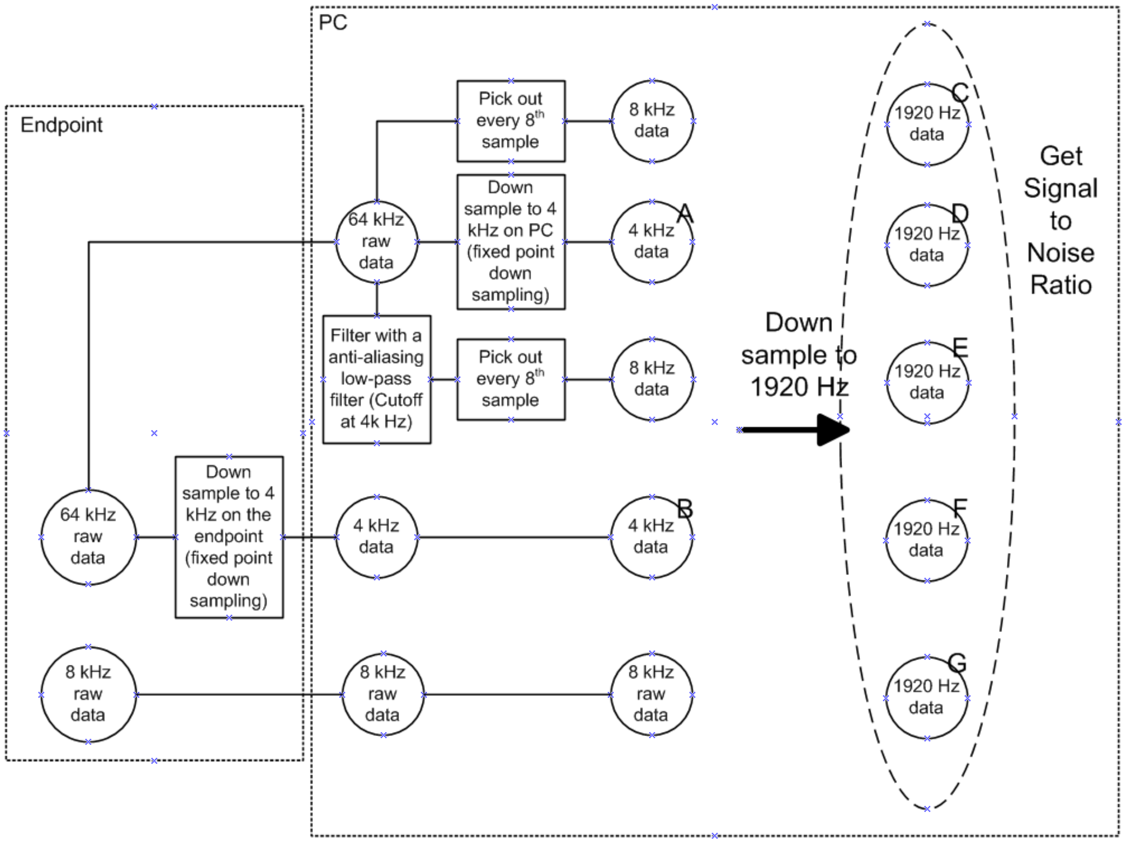

PC
Endpoint
Pick out every 8th sample
8 kHz data
Down sample to 4 kHz on PC (fixed point down sampling)
64 kHz raw data
4 kHz data
A
Filter with a anti-aliasing low-pass filter (Cutoff at 4k Hz)
Pick out every 8th sample
8 kHz data
Down sample to 1920 Hz
64 kHz raw data
Down sample to 4 kHz on the endpoint (fixed point down sampling)
4 kHz data
4 kHz data
B
8 kHz raw data
8 kHz raw data
8 kHz raw data
C
1920 Hz data
D
1920 Hz data
E
1920 Hz data
F
1920 Hz data
G
1920 Hz data
Get Signal to Noise Ratio

Figure 14. Comparison over sample, down sample, bit resolution

D. Software Optimization

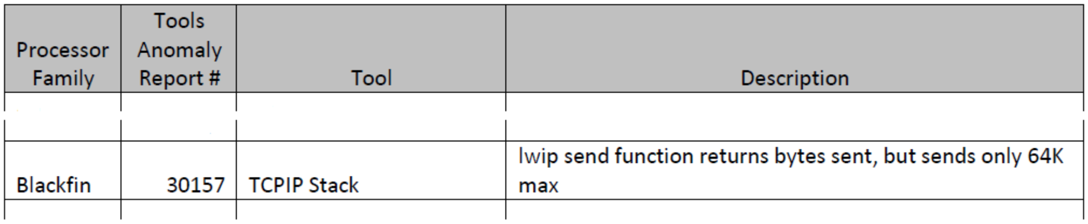
1. Transfer Speed

Based on the previous experience, the transfer speed of the original software complied in VisualDSP++4.0 is pretty much lower than network hardware limitation, since a new TCPIP stack is released with VisualDSP++5.0 and several important updates are made, trying to optimize transfer speed by compiling endpoint software in the latest environment should be considered.

a. Difference between VisualDSP++ 4.0 and 5.0

After the release notes of the recent VisualDSP++ versions is reviewed, an important anomaly related to data transfer speed was fixed in VisualDSP++5.0 Update 4 [20]. The detailed description is shown in Table 7.

Table 7. Release note



Processor Family	Tools Anomaly Report #	Tool	Description
Blackfin	30157	TCPIP Stack	lwip send function returns bytes sent, but sends only 64K max

b. Endpoint Software Implement in VisualDSP++5.0

In the original software, to properly send the data, one data set was divided into several 60KB packages; all these packages were sent to PC by the "lwip send function" one by one separately. This mechanism greatly impacts the transfer speed.

Simply compiling the endpoint software in the latest environment without removing the above mechanism will not help much on transfer speed. To optimize the transfer speed completely, when trying to compile the endpoint software in VisualDSP++5.0, the "lwip send function" is modified to send the data set to PC in a single package rather than several 60KB small packages.

2. Threads Configuration

In the original endpoint software mechanism, the A/D converter interrupt is disabled before the raw data is being transferred, and re-enabled after raw data transferred, there is no data collected when transfer is in progress, this indicates that data collection and transfer are configured in series, so time could be saved if data collection and transfer can be configured in parallel.

Removing the above parts in end point software is the first thing to start with, while A/D converter is not disabled during data transfer. The data collection thread and transfer thread are running in parallel. But one single raw data buffer will not be enough in this configuration, because new data will be written to the buffer while the content of the same buffer is transferred to PC. So implementation of two raw data buffer is required, during one cycle, one buffer is receiving raw data from the A/D converter and the other one is sending raw data to PC, two processes are running simultaneously, the first finished process will wait for the other one to be finished. After both processes are finished, two buffers will be swapped and the software will start the next cycle, the buffer which sent data in the previous cycle will receive raw data from the A/D converter, and the buffer which received data from the A/D converter in the previous cycle will send data to PC. These cycles will be running one after one when the endpoint software is running. The parallel threads timeline is shown in Figure 15.

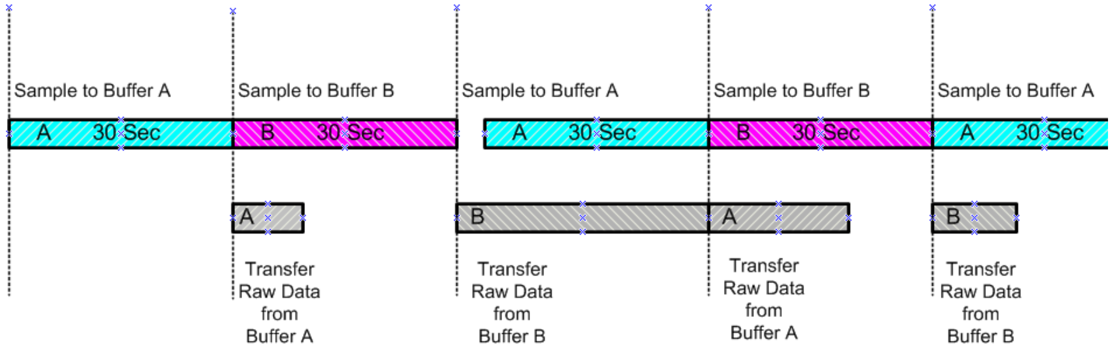


Figure 15. Timeline parallel treads configuration

The raw data buffer is designed to contain only newest raw data, if the transfer can not be initiated when the raw data buffer is fully filled; the oldest sample will be overwritten by the newest sample, so only the latest 30 seconds raw data is stored in the buffer.

$$DR = \frac{3600\,[sec/\,hour]}{Max_{average}(\text{Sampling Time (30 sec), Transfer Time})} \tag{5}$$

By implementing this algorithm, the way to compute “DAT” will be also changed. The original way is make a sum, and equation reflects the upgraded way is getting the maximum instead of sum in parallel treads configuration software (Equation (5)). Among all four proposed approaches mentioned in Chapter I, the method discussed in this Chapter is the only one which tries to increase “DR” by changing the way to compute “DAT” without reducing the time cost by individual section.

The A/D converter disabling and re-enabling parts are removed from the end point software and multiple buffers mechanism is added to the endpoint software.

E. Continuous Waveform Sampling

Since the software optimizations are independent with the preprocessing function, in this section, the preprocessing function on the endpoint software will be combined with transfer speed optimization and parallel threads configuration to pursue maximum “DR”.

1. Combination

Because the endpoint software with parallel threads configuration and the endpoint software with data down sampling are already been compiled in VisualDSP++5.0 separately, so the combination of transfer speed optimization and parallel threads configuration (Results shown in section IV.B) and the combination of transfer speed optimization and data preprocess (Results shown in section IV.C) are already been tested. When trying to combine parallel thread configuration and data preprocess, the most critical problem is the memory issue, if the above two features are simply integrated, two raw data buffers (two 64 kHz buffers) are required for parallel threads, one filtered data buffer (64 kHz 24-bit buffer) and one down sampled data buffer (4 kHz buffer) are required for preprocess. Since all those buffers above will totally require 82.8 MB memory and there is only a 64 MB memory block implemented in the endpoint, so simple integration of two methods without modifications is not feasible.

Since the down sampling mechanism on the endpoint has been verified in the previous chapter, so it is really unnecessary to send the 64 kHz raw data from the endpoint to PC. And also to get one filtered sample by applying the low pass filtering function once, only a few samples rather than the whole data set are needed, the number of the samples required is determined by the order of the filter.

Based on this feature, in order to save memory on the endpoint, a filtering function initiated by the A/D hardware interrupt is implemented; the filtering function will run once to get one filtered sample when a new raw sample is acquired from the A/D converter; only the newest few raw samples and filtered samples are store in a small raw data buffer and a small filtered data buffer (both buffers are milliseconds length) to maintain the filtering function. The oldest sample will be immediately replaced by the newest sample when it is ready. For every 16th times the filtering function runs, one filtered sample will be stored to the down sampled 4 kHz 24-bit data buffer (30 seconds length) which is designed to store 4 kHz data to finish the down sampling process. In this case, since the order of the IIR filter is 5, instead of two 64 kHz swapping raw data buffer, one 64 kHz filtered data buffer and one 4 kHz down sampled buffer; only one 6 samples raw data buffer, one 6 samples 24-bit filtered data buffer and two 4 kHz 24-bit swapping down sampled buffers are required if this real-time down sampling function is implemented, this will only require 3.60018 MB memory (actually, the memory required by the two 64 kHz data buffers can be ignored).The endpoint software mechanism is shown in Figure 16.

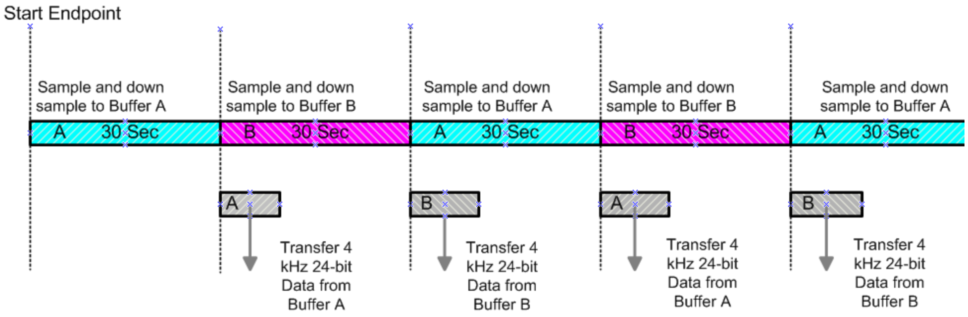

Start Endpoint
Sample and down sample to Buffer A
Sample and down sample to Buffer B
Sample and down sample to Buffer A
Sample and down sample to Buffer B
Sample and down sample to Buffer A
A 30 Sec
B 30 Sec
A 30 Sec
B 30 Sec
A 30 Sec
A
B
A
B
Transfer 4 kHz 24-bit Data from Buffer A
Transfer 4 kHz 24-bit Data from Buffer B
Transfer 4 kHz 24-bit Data from Buffer A
Transfer 4 kHz 24-bit Data from Buffer B

Figure 16. Endpoint swapping data buffer software timeline

2. Continuous Waveform Acquisition

Originally, the minimum time length requirement of one data set for the advanced data processing procedure in PC software is 30 seconds, this is the reason that on the endpoint data are stored and send in a 30 seconds set, but if the time domain continuity between every two consecutive data sets can be guaranteed, more flexibility and possibility can be achieved for future software development because two consecutive data sets can be simply connected to increase the length. After the combination of transfer speed optimization, parallel threads configuration and data preprocess is realized in the previous section, transfer time is reduced and the A/D converter is no longer been disabled after endpoint startup, continuous data acquisition feature becomes feasible to be explored.

The first step to try to achieve continuous data acquisition feature is to replace the two 4 kHz swapping data buffers with one 60 seconds 4 kHz pool data buffer, because there may be sample losses when swapping the buffers and it is easier to maintain one pointer from one larger buffer than two pointers from two small buffers. And in order to guarantee that data which is being transferred will not be corrupted by sampling function, a 4 kHz transfer buffer is also implemented; this buffer will copy 4 kHz data from 60 seconds 4 kHz pool buffer when the transfer is request and the new data are ready, the value of the start pointer of every data copy will always be one plus the value of the end pointer of the previous data copy unless the real-time pointer for storing the down sampled data to 60 seconds 4 kHz pool buffer reaches the end pointer of the previous copy so there is no way to maintain the time domain continuity, when this happens, the pointer for copying data to 4 kHz transfer buffer will be reset, and newest data will be copied to transfer buffer. The mechanism and timeline of final endpoint software with continuous waveform acquisition feature are shown in Figure 17.

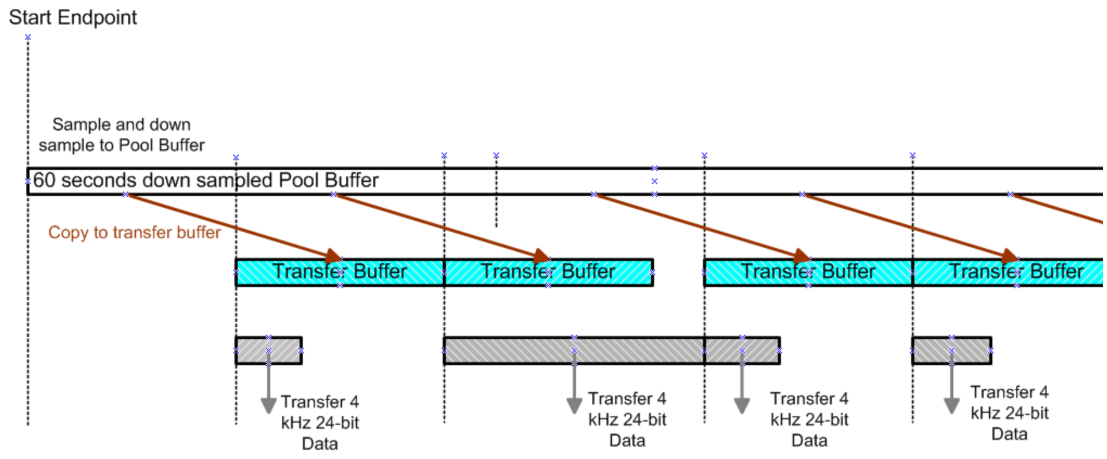

Start Endpoint
Sample and down sample to Pool Buffer
60 seconds down sampled Pool Buffer
Copy to transfer buffer
Transfer Buffer
Transfer Buffer
Transfer Buffer
Transfer Buffer
Transfer 4 kHz 24-bit Data
Transfer 4 kHz 24-bit Data
Transfer 4 kHz 24-bit Data
Transfer 4 kHz 24-bit Data

Figure 17. Endpoint continuous data acquisition software timeline

3. Performance and Verification

Since the transfer time and bit resolution improvements are being tested in the previous sections. The only feature which is needed to be verified is the continuous data acquisition. To test this, a testing endpoint software version with the following mechanism is implemented. The timeline of this software is shown in Figure 18.

Both raw data and down sampled data will be transferred to PC to verify time domain continuity, In order to avoid similar memory issue mentioned above, in this section, all data buffers lengths will be reduced to 1/3.

One 4 kHz 24-bit 20 seconds down sampled pool buffer is implemented; instead of transferring data per 10 seconds, endpoint will transfer one 10 seconds data set per 5 seconds, this means that every data set received by PC will have 5 seconds overlap with the previous one, if the network condition is good enough, the connection of even data sets can match the connection of odd data sets, therefore the continuous data acquisition can be verified. Together with 4 kHz data, the pointer and buffer fulfilled counter of the 60 seconds pool buffer are also transferred to PC to further verify the continuity.

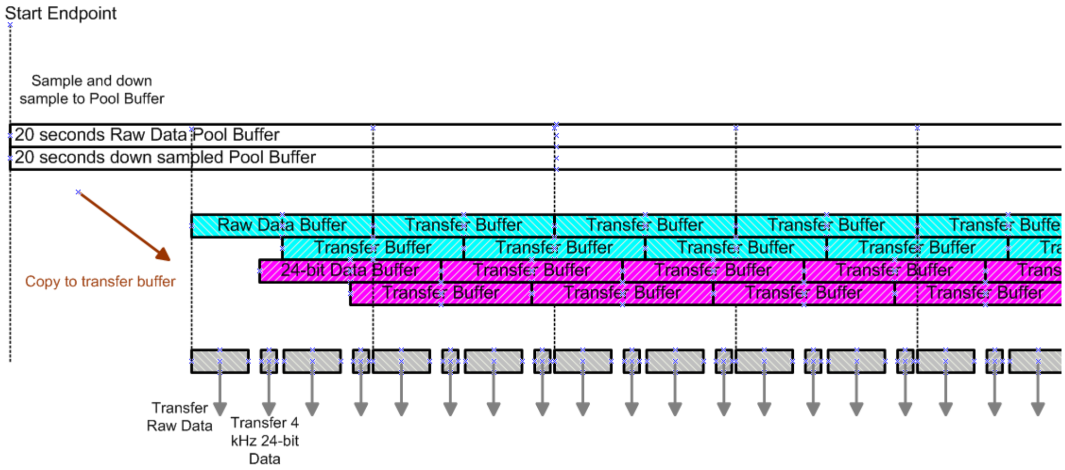

Start Endpoint
Sample and down sample to Pool Buffer
20 seconds Raw Data Pool Buffer
20 seconds down sampled Pool Buffer
Copy to transfer buffer
Raw Data Buffer
Transfer Buffer
24-bit Data Buffer
Transfer Raw Data
Transfer 4 kHz 24-bit Data

Figure 18. Endpoint overlap down buffer software timeline

The raw data time domain continuity is also need to be verified by the similar method. In this test version, instead of discarding raw data, the endpoint software will store the raw data to a 64 kHz 20 seconds pool buffer. And also endpoint will send 10 seconds raw data with 5 seconds overlap. Then the connection and comparison procedures are similar with down sampled data time domain continuity verification.

All versions above are running on an endpoint configured to Ethernet mode because the network load will be several times heavier than the standard version.

F. Chapter Summary

In this chapter, methods to verify the bit resolution by oversampling are developed initially. Two proposed approaches are also implemented and tested, DEFLATE lossless compression method is eliminated for its low compression ratio and long compression time. Also two types of software optimization are considered to further improve the software performance. A combination of data preprocessing and software optimization which tries to provide continuous waveform sampling in both wired and wireless network modes is presented finally.

CHAPTER IV

EXPERIMENTAL RESULTS

A. Oversampling for Bit Resolution Improvement

30 sets of 8 kHz raw data and 30 sets of 64 kHz raw data are sampled and transferred to PC, the detailed signal processing procedures on PC are described in section III.A. From each data processing route, one SNR can be computed. All SNR are listed in Table 8.

Since the signal is a sine wave, assuming the quantization noise approximates a saw tooth wave with peak-to-peak amplitude of one quantization level, and uniform distribution. In this case, the SNR can be shown in Equation (6) approximately.

$$SNR \approx 20\log_{10}\left(2^{N}\sqrt{\frac{3}{2}}\right) = N \cdot 20\log_{10} 2 + 10\log_{2}\frac{3}{2} \approx N \cdot 6.02 + 1.761 \tag{6}$$

The comparison is done between the final averaged SNR because average SNR among time will reduce the impact of signal quality variance. From the comparison above, about 7 to 8 db SNR improvements are achieved, according to the relationship between signal-to-noise-ratio and bit resolution in Equation (6), the improvement in SNR indicates approximate 1.2 bits resolution improvement, this is lower than expectation but still be significant to make the changes in software.

During this test, when 64 kHz raw data are transferred to PC, the transfer time is much longer than transferring 8 kHz raw data. This is the exact problem mentioned in Chapter I.

Table 8. Comparison #1 and #2

	Current Channels Average			Voltage Channels Average		
	Comparison #1			Comparison #1		
		Comparison #2			Comparison #2	
	8 kHz Pick up	64 kHz Raw	8 kHz Raw	8 kHz Pick up	64 kHz Raw	8 kHz Raw
#1	92.6149	100.0564	89.5940	96.2849	101.5967	93.5896
#2	93.4604	98.7273	91.6959	92.0264	100.9825	96.4165
#3	92.4515	99.1181	92.4360	99.8192	99.6987	95.1918
#4	91.4210	98.5680	95.1438	96.8318	101.5964	92.8022
#5	92.1911	100.3344	92.7562	92.5314	99.1466	94.4256
#6	93.7689	101.0399	91.4378	94.3644	102.3067	92.8594
#7	91.4963	98.8671	91.4333	94.2302	101.6071	93.9735
#8	92.8515	99.8290	92.3050	92.9894	101.2684	92.7647
#9	91.0055	101.4810	90.2501	91.5896	100.2028	91.8314
#10	92.0610	96.9731	93.0810	91.9599	100.2678	92.1416
#11	96.0363	100.6243	91.3837	95.0318	101.6650	93.5521
#12	90.1855	99.4840	77.3578	94.0955	100.8867	93.6186
#13	92.6735	99.1745	90.8646	94.2683	102.5280	94.3054
#14	90.9795	100.6783	92.1319	98.2476	100.2306	94.6645
#15	92.8689	99.8642	90.7070	92.2061	99.2261	93.8469
#16	91.5520	99.0797	92.8560	94.2015	101.2490	91.4112
#17	91.6552	98.3112	92.3674	95.6864	101.2493	97.1579
#18	91.2133	100.2681	91.7468	93.0761	101.8423	98.2948
#19	93.2612	100.9099	90.0588	91.7344	100.3710	94.3275
#20	93.4840	99.6947	93.0585	94.6698	101.4732	93.5774
#21	92.5980	99.1383	92.4152	93.8248	105.3129	93.4124
#22	90.5211	100.9572	94.0503	93.1475	101.5750	92.5296
#23	92.0950	99.1691	92.9111	92.9738	100.2681	93.6669
#24	92.5504	105.3881	89.9685	92.3068	100.9256	93.2321
#25	91.9126	99.9765	90.4026	97.8772	99.7534	95.2257
#26	91.4360	104.1769	97.4400	94.4687	97.7328	93.3556
#27	93.1754	99.5024	90.0128	93.7479	105.7964	94.0919
#28	92.0077	102.3750	90.3904	92.2992	99.1199	96.7721
#29	92.7101	98.0866	96.4320	91.4984	102.9829	93.2283
#30	92.0060	100.4463	91.4557	98.7369	101.8402	94.3323
Average	**92.2748**	**100.0767**	**91.6048**	**94.2242**	**101.1567**	**94.0200**

B. Oversampling and Preprocessing

30 sets of 8 kHz raw data and 30 sets of 64 kHz raw data are sampled; 4 kHz data preprocessed from 64 kHz raw data together with all raw data are transferred to PC, the detailed procedures are described in Chapter III. From each data processing route, one SNR can be computed. All timestamps are listed in Table 9 and all SNR are listed in Table 10 and Table 11.

Table 9. Timestamps for down sampling and transfer

	Ethernet			Wireless		
	Down Sampling Time recorded by Endpoint	Transfer Time recorded by PC	DAT recorded by PC	Down Sampling Time recorded by Endpoint	Transfer Time recorded by PC	DAT recorded by PC
#1	6.25	0.62	37	6.23	3.52	40
#2	6.38	0.60	37	6.27	3.98	41
#3	6.38	0.59	37	6.23	3.74	40
#4	6.35	0.59	37	6.36	3.61	40
#5	6.42	0.63	37	6.51	3.62	40

Table 10. Down sampling, SNR comparison (current)

Current					
		#1		#1	
		#2	#2		
	#3	#3		#3	
	C	D	E	F	G
#1	92.6149	100.0564	100.4002	100.0564	89.5940
#2	93.4604	98.7273	100.0256	98.7273	91.6959
#3	92.4515	99.1181	100.5230	99.1181	92.4360
#4	91.4210	98.5680	99.5442	98.5680	95.1438
#5	92.1911	100.3344	100.3351	100.3344	92.7562
#6	93.7689	101.0399	101.1143	101.0399	91.4378
#7	91.4963	98.8671	98.9465	98.8671	91.4333
#8	92.8515	99.8290	100.2113	99.8290	92.3050
#9	91.0055	101.4810	102.2192	101.4810	90.2501
#10	92.0610	96.9731	98.6238	96.9731	93.0810
#11	96.0363	100.6243	100.9629	100.6243	91.3838
#12	90.1855	99.4840	100.1844	99.4840	87.3578
#13	92.6735	99.1745	99.0581	99.1745	90.8646
#14	90.9796	100.6783	101.1076	100.6783	92.1319
#15	92.8689	99.8642	100.9436	99.8642	90.7070
#16	91.5520	99.0797	99.8150	99.0797	92.8560
#17	91.6552	98.3112	100.0493	98.3112	92.3674
#18	91.2134	100.2681	100.0861	100.2681	91.7468
#19	93.2612	100.9099	101.8689	100.9099	90.0588
#20	93.4840	99.6947	99.5186	99.6947	93.0585
#21	92.5980	99.1383	99.6636	99.1383	92.4152
#22	90.5211	100.9572	102.3562	100.9572	94.0503
#23	92.0950	99.1691	99.7739	99.1691	92.9111
#24	92.5504	105.3881	105.7398	105.3881	89.9685
#25	91.9126	99.9765	100.5350	99.9765	90.4026
#26	91.4360	104.1769	103.4587	104.1769	97.4400
#27	93.1754	99.5024	99.5312	99.5024	90.0128
#28	92.0077	102.3750	102.4354	102.3750	90.3904
#29	92.7101	98.0866	99.7598	98.0866	96.4320
#30	92.0061	100.4463	100.8309	100.4463	91.4557
Average	**92.2748**	**100.0767**	**100.6541**	**100.0767**	**91.9381**

Table 11. Down sampling, SNR comparison (voltage)

Voltage					
		#1		#1	
		#2	#2		
	#3	#3		#3	
	C	D	E	F	G
#1	93.7590	100.9231	100.3647	100.9231	92.0672
#2	94.0013	100.8377	100.6545	100.8377	92.8798
#3	94.4119	99.0991	100.2162	99.0991	92.5313
#4	92.5214	100.1262	100.2198	100.1262	97.7657
#5	92.4507	101.1092	100.4247	101.1092	93.9474
#6	94.9292	102.2158	101.5743	102.2158	93.0061
#7	92.2703	98.6017	98.1989	98.6017	92.3580
#8	92.6650	100.6192	100.3253	100.6192	92.9316
#9	91.0954	102.6067	102.6455	102.6067	90.0197
#10	92.7553	99.5226	99.6727	99.5226	93.7286
#11	94.6639	101.6767	101.3781	101.6767	93.8783
#12	94.7853	99.7794	100.0414	99.7794	92.5193
#13	93.0045	101.1264	100.8904	101.1264	93.0507
#14	92.3098	101.9417	101.6292	101.9417	93.8658
#15	92.4125	100.8377	101.7043	100.8377	92.7032
#16	92.2650	99.8701	99.8349	99.8701	93.7136
#17	91.8905	99.5095	101.0228	99.5095	92.5159
#18	93.2100	101.2730	100.3220	101.2730	94.4166
#19	93.2039	102.5100	101.9126	102.5100	91.5728
#20	97.7059	99.7204	99.1515	99.7204	93.6864
#21	93.8934	101.1600	101.3199	101.1600	93.8330
#22	92.3811	105.7704	105.7180	105.7704	95.4885
#23	93.5820	99.6448	99.6680	99.6448	93.4982
#24	93.6604	110.9872	108.9296	110.9872	91.8460
#25	93.2210	100.1188	100.2706	100.1188	92.7900
#26	92.1323	103.1918	101.8648	103.1918	92.2208
#27	94.3986	101.1650	100.5178	101.1650	92.4940
#28	92.0438	102.6871	102.1965	102.6871	90.8014
#29	94.3461	99.1149	100.3939	99.1149	99.3889
#30	92.6892	101.2778	101.5048	101.2778	91.5605
Average	**93.2886**	**101.3008**	**101.1522**	**101.3008**	**93.2360**

The test result shows that the down sampling functions on the endpoint and on PC gives the same down sampled signal (Comparison between A and B in Figure 18). And from Table 10 and Table 11, bit resolution improvement by oversampling is confirmed again (Comparison details shown in Figure 18). According to Table 10 and Table 11, the down sampling function will take approximately 6 to 7 seconds to finish 64 kHz down sampling.

In this case, the "DAT" will be the sum of sampling time, down sampling time and transfer time (Equation (7)). Since the size of the down sampled 4 kHz 24-bit data is 2.16 MB, which is much smaller than the 64 kHz raw data size, the transfer time reduction will be significant.

$$DR = \frac{3600\,[sec/\,hour]}{average(\text{Sampling Time (30 sec)} + \text{Down Sampling Time} + \text{Transfer Time})} \tag{7}$$

With transfer speed optimization (detailed transfer time are listed in Table 12, Table 13, Table 14 and Table 15), according to Table 9, the data preprocessing method will need 6 to 7 seconds more to do down sampling, meanwhile it can save 4 to 5 seconds transfer time in Ethernet mode and about 30 seconds in wireless mode. Compared to the endpoint software which transfers 64 kHz raw data without down sampling, the endpoint software which transfers down sampled 4 kHz data will increase "DR" from 60 to 90 in wireless mode, but it will decrease "DR" in Ethernet mode a little bit from 100 to 95.

Although the "DR" will not be improved in the Ethernet mode, the overall "DR" performance still becomes better. And preprocessing endpoint software can greatly decrease the network load by reducing transfer size, since the network segment is the most unstable part of the entire system, reducing network load will increase the stability of the system. So implementation of the down sampling function on the endpoint is a successful method to improve the overall performance.

C. Software Optimization

1. Transfer Speed

An original software with 60KB send package limitation compiled in VisaulDSP++4.0 and an upgraded software without send size limitation compiled in VisualDSP++5.0 are loaded to the endpoint and tested separately. Both 8 kHz and 64 kHz data transfer are tested in both Ethernet and wireless mode, the results are shown in Table 12, Table 13, Table 14 and Table 15.

Table 12. 8 kHz Ethernet transfer time

Transfer Time	VisualDSP++4.0		VisualDSP++5.0
(Sec)	Case 1	Case 2	
#1	40.81	5.74	0.86
#2	39.28	5.36	0.84
#3	41.02	5.55	0.78
#4	40.53	5.32	0.86
#5	40.25	5.63	0.84

Table 13. 8 kHz Wireless transfer time

Transfer Time (Sec)	VisualDSP++4.0	VisualDSP++5.0
#1	11.28	4.828
#2	10.83	4.75
#3	12.52	4.796
#4	11.23	4.734
#5	12.5	4.797

Table 14. 64 kHz Ethernet transfer time

Transfer Time	VisualDSP++4.0		VisualDSP++5.0
(Sec)	Case 1	Case 2	
#1	322.25	40.13	6.62
#2	321.36	41.35	6.73
#3	319.24	40.36	6.61
#4	322.52	40.72	6.12
#5	320.86	41.58	6.73

Table 15. 64 kHz Wireless transfer time

Transfer Time (Sec)	VisualDSP++4.0	VisualDSP++5.0
#1	95.24	38.62
#2	90.65	38.00
#3	88.23	38.37
#4	92.53	37.87
#5	92.88	38.38

By updating the endpoint software compilation environment from VisualDSP++4.0 to VisualDSP++5.0 (with Update 4 or later), the speed and stability of data transfer is greatly optimized, especially in Ethernet mode, the transfer time and “DAT” are reduced. So updating endpoint software compilation environment should be considered as an additional option to increase “DR”.

2. Threads Configuration

After the endpoint software is successfully configured to parallel threads mode, a functional system which includes entire endpoint software (samples at 64 kHz) and first module (data receiving) of server software is tested. Transfer timestamps are recorded on PC to verify the endpoint software mechanism, when the endpoint software was configured to series mode, the time interval between previous transfer end point and following transfer start point was 30 seconds, when the endpoint software is configured to parallel mode, either the time interval between the start points of two consecutive transfers is 30 seconds (transfer time is less than 30 seconds) or there is no time interval between previous transfer end point and following transfer start point (transfer time is more than 30 seconds).

Both endpoint software complied in VisualDSP++4.0 and VisualDSP++5.0 are tested. The waveform and signal quality of the raw data received and stored on PC are also been analyzed to verified that the validity of the raw data has not been impacted as the endpoint software mechanism changes. Table 16 and Table 17 show the 64 kHz raw data transfer time when the software threads are configured to parallel, compared to Table 14 and Table 15, the transfer time will not be impacted by the endpoint software configuration is. When endpoint software is compiled in VisualDSP++4.0, transfer 64 kHz raw data will generally cost much more than 30 seconds in both Ethernet and Wireless network mode (See detailed performance in Table). When endpoint software is compiled in VisualDSP++5.0, transfer 64 kHz raw data will cost about 8 seconds in Ethernet mode but 30 to 40 seconds in wireless network mode.

Table 16. Parallel mode Ethernet transfer time

Transfer Time (Seconds)	VisualDSP++4.0		VisualDSP++5.0
	Case 1	Case 2	
#1	321.36	41.25	6.75
#2	322.57	40.89	6.53
#3	320.88	42.03	6.88
#4	323.52	42.21	6.21
#5	322.68	41.13	6.03

Table 17. Parallel mode Wireless transfer time

Transfer Time (Seconds)	VisualDSP++4.0	VisualDSP++5.0
#1	97.43	38.23
#2	92.52	38.05
#3	90.39	38.74
#4	95.43	37.91
#5	89.87	39.26

The test results indicate that when endpoint software with parallel threads configuration is compiled in VisualDSP++5.0 (i.e. with transfer speed optimization) and the endpoint is configured to work in Ethernet mode, since the transfer time will always be less than 30 seconds, the data acquisition system is working in its ideal condition. So there is a potential that all analog signal can be sampled and transferred to PC; “DR” can reach the maximum value which is 120. (To get a real 120 “DR”, more modifications and verifications are needed, as explained in section III.E). If the endpoint is configured to wireless mode, the transfer time will still be longer than the sampling time even the software is compiled in VisualDSP++5.0.

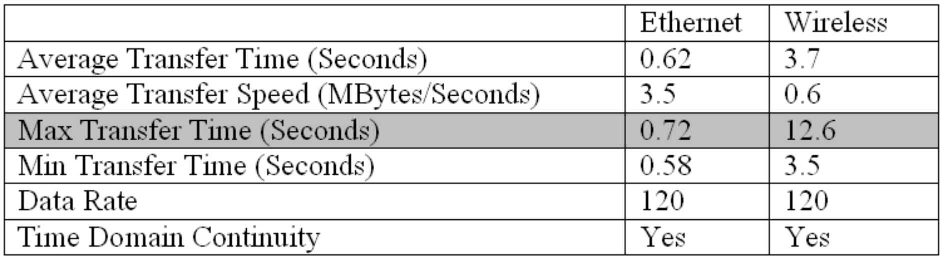
D. Continuous Waveform Sampling

1. Waveform Continuity and Software Stability

The waveform time domain continuities of both raw data and preprocessed data are verified by connecting two consecutive data sets and comparing the connection with the data set which overlaps both of them. The values from the connected data set can exactly match the values from the overlapping data set.

Since the continuous data acquisition is very sensitive to the transfer time, which is highly depended on the network condition, to test the stability of continuous data acquisition, a system with only one endpoint connected to PC through a private network is established; the endpoint software is shown in Figure 17, and the endpoint is configured and tested in Ethernet mode and wireless mode separately, each system (Ethernet and wireless) is tested for 24 hours, during this period of time, according to Table 18, the transfer time are fairly stable and the continuous data acquisition is successfully maintained.

Table 18. Stability tests result



	Ethernet	Wireless
Average Transfer Time (Seconds)	0.62	3.7
Average Transfer Speed (MBytes/Seconds)	3.5	0.6
Max Transfer Time (Seconds)	0.72	12.6
Min Transfer Time (Seconds)	0.58	3.5
Data Rate	120	120
Time Domain Continuity	Yes	Yes

The final endpoint software solution is the version used in the previous continuous data acquisition stability test section, which can be shown in Figure 17.

The first module and part of the second module of the PC software will also need to be correspondingly changed as the endpoint software changes, the final setup of the modified PC software can be described in the Figure 16. In the this final software solution, the maximum “DR” 120 has been achieved during 24 hours time test and verified by time domain continuity (Equation (8)).

$$DR = \frac{3600\,[\mathrm{sec}/\mathit{hour}]}{Max_{average}(\text{Sampling and Down Sampling Time (30 sec), Transfer Time})} \tag{8}$$

2. DSP Usage

Also the overall DSP usage on the endpoint can be checked on the VisualDSP++ interface through the emulator while the software is running; Figure 19 and Figure 20 show the DSP usages in the original software and final solution developed in this research. In the original software, the DSP usage is very low; the usage rises a little bit only when the DSP is trying to transfer data. In the new solution, the capability of the DSP is utilized more efficiently by increasing the sampling frequency and applying preprocessing on the endpoint.

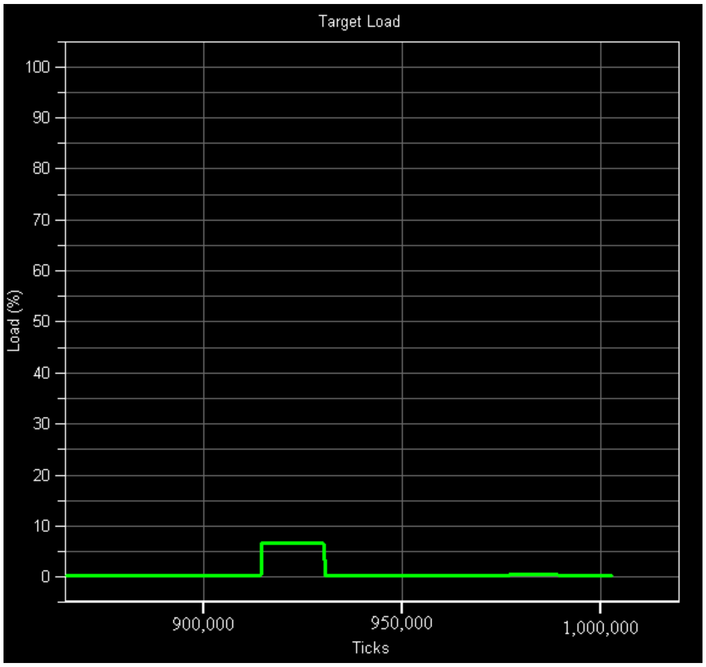


Figure 19. Original endpoint DSP usages

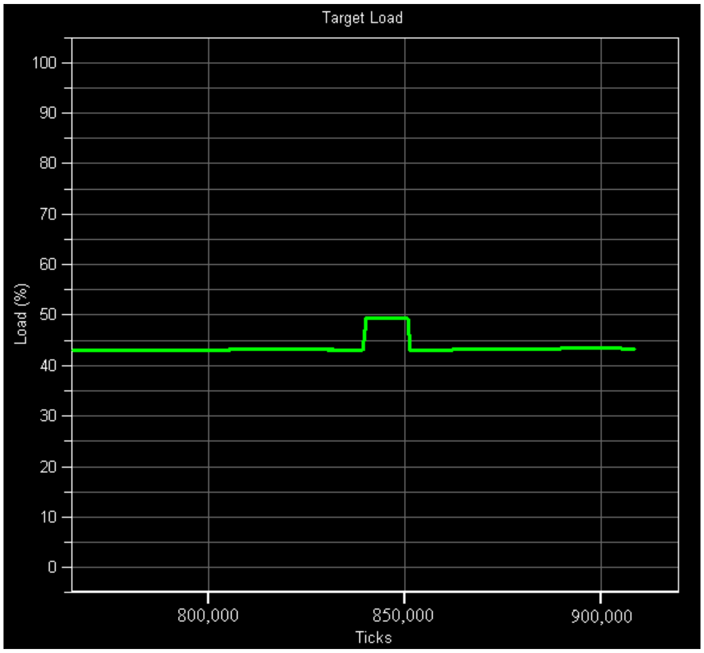


Figure 20. DSP Usages of the new solution

E. Chapter Summary

In this chapter, the bit resolution improvement by oversampling is confirmed. And data preprocessing function which can reduce the transfer data size in the endpoint software is also verified. Finally, the combination of data preprocessing and software optimization has been proved to provide continuous waveform sampling.

CHAPTER V

SUMMARY AND CONCLUSIONS

A. Research Summary

This thesis presented the development and performance of a bit resolution improving and continuous waveform sampling solution for multi-phase energy measurement system without changing the hardware setup. Several possible methods are explored during the development to optimize DR. The final solution of the software sets the A/D converter at its maximum sampling frequency 64 kHz, down samples the raw data and transfers the down sampled data to PC software without disable the A/D sampling function.

B. Conclusion

This software solution achieves 1.2 to 1.3 bit resolution improvement and also maximizes the amount of the data sets received by PC; it can even provide a continuous waveform acquisition feature if the network condition can be guaranteed.

C. Recommendations for Future Work

1. Filter Compensation

Since the IIR filter can meet the requirement with much less order and computation than FIR filter, an IIR filter is selected to finish the down sampling procedure on the endpoint. The IIR filter has a non-linear phase shift feature; this won't bring many troubles since the current data processing mechanism does not concern much about the phase information, but this limits the future software development. To dress this issue, either a proper phase compensation or replacement of an FIR filter should be considered.

2. Network Capacity

In the continuous data acquisition stability test, to guarantee network condition, only one endpoint is setup, this reduces the PC software efficiency. Generally speaking, for every endpoint, 2.16 MB data are transferred per 30 seconds, this is about 72KB/s, this pretty much lower than the network bandwidth limitation (100Mb for Ethernet, 4Mb for wireless), so it is possible to increase the amount of the endpoint in the system without losing data rate. A balance between data rate and PC software efficiency based on network configuration can be investigated in the future.

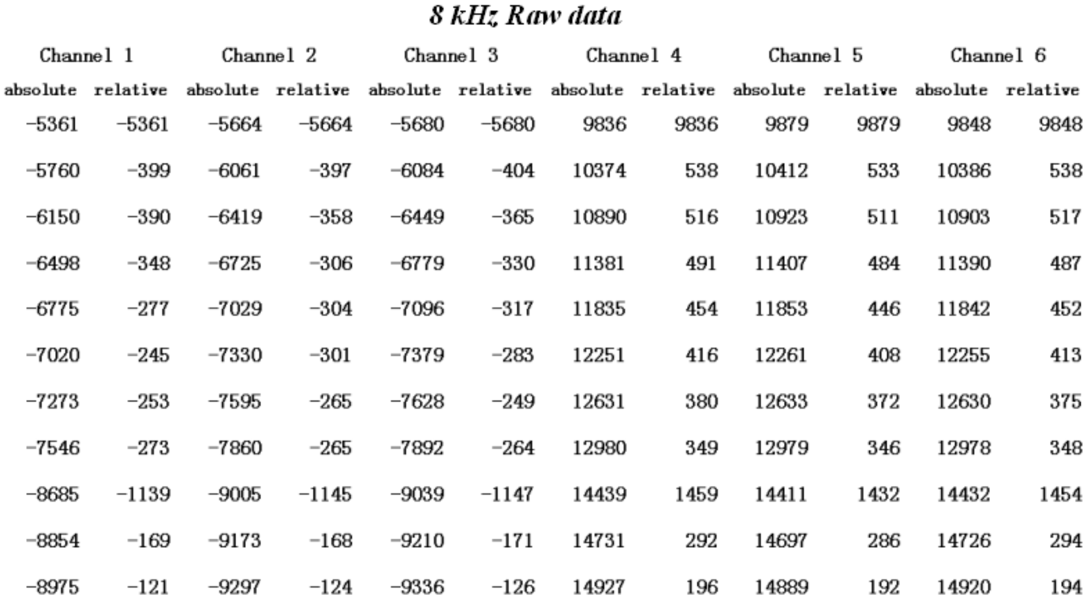
3. Compression

Deflate compression algorithm can not compress the raw data very much because the raw data is treated as random and repeating times of each 16-bits value is very low. But in fact, the raw data sampled from electric power is not really random; it is like a sinusoidal wave with a certain fundamental frequency, if this feature can be utilized, a special compression algorithm for data from electric power may be developed.

A simple thought on compression has been tested to verify the potential compression possibility. Instead of recording absolute value of every sample, a relative value which is the difference between the two consecutive samples is calculated and recorded.

One set of 64 kHz raw data and one set of 8 kHz raw data have been tested. Except for the first sample, all other samples are represented by the difference between itself and the previous one.

Table 19. Difference between absolute value and relative value



8 kHz Raw data

Channel 1		Channel 2		Channel 3		Channel 4		Channel 5		Channel 6	
absolute	relative	absolute	relative	absolute	relative	absolute	relative	absolute	relative	absolute	relative
−5361	−5361	−5664	−5664	−5680	−5680	9836	9836	9879	9879	9848	9848
−5760	−399	−6061	−397	−6084	−404	10374	538	10412	533	10386	538
−6150	−390	−6419	−358	−6449	−365	10890	516	10923	511	10903	517
−6498	−348	−6725	−306	−6779	−330	11381	491	11407	484	11390	487
−6775	−277	−7029	−304	−7096	−317	11835	454	11853	446	11842	452
−7020	−245	−7330	−301	−7379	−283	12251	416	12261	408	12255	413
−7273	−253	−7595	−265	−7628	−249	12631	380	12633	372	12630	375
−7546	−273	−7860	−265	−7892	−264	12980	349	12979	346	12978	348
−8685	−1139	−9005	−1145	−9039	−1147	14439	1459	14411	1432	14432	1454
−8854	−169	−9173	−168	−9210	−171	14731	292	14697	286	14726	294
−8975	−121	−9297	−124	−9336	−126	14927	196	14889	192	14920	194

For 8 kHz raw data, if it is stored by absolute value, the range is from -14576 to 15422, which is within -16384 to 16383 and can be represented by a 15-bit long number. If it is stored by relative value, the range is from -1147 to 1459, which is within -2048 to 2047 and can be represented by a 12-bit long number. A piece of 8 kHz data is shown in Table 19.

Table 20. Proposed compression ratio

Sampling Rate	Absolute value size (bit/sample)	Relative value size (bit/sample)	Compression Raito
8 kHz	15	12	1.25
64 kHz	16	10	1.6

For 64 kHz raw data, if it is stored by absolute value, the range is from -28864 to 25668, which is within -32768 to 32767 and can be represented by a 16-bit long number. If it is stored by relative value, the range is from -356 to 400, which is within -512 to 511 and can be represented by a 10-bit long number. In the Table 20, a possible compression ratio is calculated; and as the raw data sampling frequency increases, the compression ratio may get higher.

REFERENCES

[1] A. G. Parlos, K. Kim, and R. M. Bharadwaj, “Sensorless detection of mechanical faults in electromechanical systems,” *Mechatronics*, vol. 14, no. 4, pp. 357-380, 2004.

[2] Determining Electric Motor Load and Efficiency, U.S. Department of Energy: Motor Challenge, 2001.

[3] Enhancing ADC resolution by oversampling, Atmel Corporation, San Jose, CA, 2005. Available: http://www.atmel.com/dyn/resources/prod_documents/doc8003.pdf

[4] D. A. Goodsell, Wireless data acquisition system for multi-phase electric power equipment, M.S. Thesis, Texas A&M Univ., College Station, TX, 2008.

[5] AD73360 Data Sheet, Analog Device Inc., Norwood, MA, 2000. Available: http://www.analog.com/static/imported-files/data_sheets/AD73360.pdf

[6] ADSP-BF537 Data sheet, Analog Device Inc., Norwood, MA, 2010. Available: http://www.analog.com/static/imported-files/data_sheets/ADSP-BF534_BF536_BF537.pdf

[7] ADE7878 Data sheet, Analog Device Inc., Norwood, MA, 2010. Available: http://www.analog.com/static/imported-files/data_sheets/ADE7878.pdf

[8] AD7656 Data sheet, Analog Device Inc., Norwood, MA, 2010. Available: http://www.analog.com/static/imported-files/data_sheets/AD7656_7657_7658.pdf

[9] An introduction to delta sigma converters, Uwe Beis, 2007. Available: http://www.beis.de/Elektronik/DeltaSigma/DeltaSigma.html#MultiBit

[10] J. Titus, “Oversample on purpose,” *Design News*, vol. 63, no. 15, pp. 20-20, 2008.

[11] T-Y Yu, G. Zhang, and A. B. Chalamalasetti, “Resolution enhancement technique using range oversampling,” *Journal of Atmospheric and Oceanic Technology*, vol. 23, no. 2. pp. 228-240, Feb. 2006.

[12] J. Ziv and A. Lempel, “A universal algorithm for sequential data compression,” *IEEE Transactions on Information Theory*, vol. 23, no. 3, pp. 337–343, 1977.

[13] O. L. Astrachan, Huffman Coding: A CS2 Assignment, 2004. Available: http://www.cs.duke.edu/csed/poop/huff/info/

[14] J. L. Gailly and M. Adler, Introduction of zlib, 2010. Available: http://zlib.net/

[15] O. N. Gerek and D. G. Ece, “Compression of power quality event data using 2D representation,” *Electric Power Systems Research*, vol. 78, no 6, pp. 1047-1052, Jun. 2008.

[16] B. Lerner, “Application dictates choice: fixed- or floating-point DSP,” *EDN*, vol. 51, no. 22, pp. 99-102, 2006.

[17] B. Pendleton, Doing It Fast, 1997. Available: http://www.gameprogrammer.com/4-fixed.html

[18] E. Pinheiro, O. Postolache, and P. Girão, “Digital notch filters implementation with fixed-point arithmetic,” in Proc, *XIX IMEKO World Congress: Fundamental and Applied Metrology*, 2009, pp. 491-496.

[19] WLNG-ET-DP100 Data sheet, Quatech Inc., Akron, OH, 2007. Available: http://www.quatech.com/pdf/g_ab_wireless_bridge_module.pdf

[20] VisualDSP++® 5.0 update 6 release notes, Analog Device Inc., Norwood, MA, 2009. Available:

http://download.analog.com/tools/patches/VisualDSP++5.0_Update_6_Release_Notes.pdf

VITA

Gang Li received his Bachelor of Science degree from Tsinghua University at Beijing, P.R. China in 2005. He entered the Mechanical Engineering program at Texas A&M University in September 2007. His research interests include signal processing, motor fault detection and diagnosis and embedded system programming.

Mr. Gang Li may be reached at Department of Mechanical Engineering, Texas A&M University, College Station, TX 77843-3123. His email address is lgsz@tamu.edu or ligangsizhu@gmail.com.