ANALOG DEVICES

AN-578 APPLICATION NOTE

One Technology Way • P.O. Box 9106 • Norwood, MA 02062-9106 • Tel: 781/329-4700 • Fax: 781/326-8703 • www.analog.com

RMS Calculation for Energy Meter Applications Using the ADE7756

By Etienne Moulin

INTRODUCTION

This application note describes the implementation of an rms signal processing algorithm using the ADE7756 and a microcontroller PIC16C63 from Microchip. The solution is implemented in an energy meter based on the ADE7756 (i.e., AN-564 reference design) and, in addition to the active energy measurement, provides the voltage and current rms measurement.

The AN-564 describes the implementation of the ADE7756 with a microcontroller. The communication between the ADE7756 and the microcontroller is done through a serial interface (SPI). The SPI port allows the user to calibrate various components of the meter including gain, offset, and phase errors. The micro-controller sends display data to the LCD and manages supervisory functions of the meter. An EEPROM is used to store various calibration parameters of the meter and to store the meter's data during a power-down.

The ADE7756 comprises of two ADCs, a reference circuit, and all the signal processing necessary for the calculation of active energy. Circuitry is provided to null out various system errors including gain, phase, and offset errors. Additional circuitry provides waveform sampling, programmable interrupts, and power line monitoring. All registers of the ADE7756 are available through the SPI port. Please refer to the ADE7756 data sheet for their description. The data sheet provides detailed information on the functionality of the ADE7756 and will be referenced several times in this application note. The AN-564 application note provides detailed information on the implementation of the ADE7756 into a microprocessor based electronics power meter.

The waveform sampling mode of the ADE7756 is used to process rms calculation into the microcontroller. This mode is used in conjunction with active energy measurement without any drawback in the performances of the meter.

The entire meter is calibrated through an external calibration routine by a PC.

DESIGN GOALS

Single-phase energy meters are low cost systems that cannot afford any programmable DSP component. The microcontroller based reference design AN-564 uses a standard 8-bit microcontroller with 192 bytes of RAM, 4K × 14 of program memory, and a 4 kbytes external EEPROM. This solution measures the active energy and drives an LCD to display the accumulation of the energy consumed.

Besides the active energy measurement, applications also need to measure the rms value of the line voltage and the rms value of the current used. These values can be used to process various electrical parameters, such as power factor and reactive energy. The objective of this document is to demonstrate that a cost-effective architecture, such as the AN-564 reference design, can support these additional measurements with sufficient accuracy when the correct design choices are made.

The goal of the solution is to comply within 1% accuracy in the rms measurement display. This accuracy is reached over the current dynamic range of a Class 100 ANSI meter extended to 500 and in the range of 10:1 for the voltage input.

In addition, this meter is based on the AN-564 and complies with the ANSI C12.16 specifications. This reference design is for a single element, Class 100 meter in a form 2S designation. Although the design in this application is limited to the ANSI standard, the achieved accuracy is well within the accuracy requirements of the IEC1036 standards for a Class 1 meter. Please refer to AN-564 for further details on the accuracy achievements of this reference design.

THEORY OF CALCULATION

This section describes the theory of rms, power factor, and reactive energy calculation, the different options to process the measurement and the interest, and the drawbacks of each of them.

RMS Calculation

Root mean square (RMS) is a fundamental measurement of the magnitude of an ac signal. Its definition can be both practical and mathematical. Defined practically, the rms value assigned to an ac signal is the amount of dc required to produce an equivalent amount of heat in the same load.

Defined mathematically, the rms value of a continuous signal V(t) is defined as

$$V_{RMS} = \sqrt{\frac{1}{T} \times \int_0^T V^2(t)dt} \quad (1)$$

For time sampling signals, rms calculation involves squaring the signal, taking the average, and obtaining the square root

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

In the solution proposed in this paper, the samples of the input waveform are used to process the rms calculation, and Equation 2 is applied.

For undistorted sine wave, the mean absolute value (V_{MAV}) of the signal is commonly used to measure the rms value. The rms value is obtained by applying a scale factor to the mean absolute value

$$V_{RMS} = \frac{\pi}{2 \times \sqrt{2}} \times V_{MAV}$$

This solution works fine as long as the input signal is an undistorted sine wave; for any other waveform, the ratio of rms/MAV changes and significant errors develop.

Power Factor Calculation

The Power Factor (PF) registers the ratio between the Active Power and the Apparent Power consumed by a load.

$$PF = \frac{Active\ \ Power}{Apparent\ \ Power}$$

$$Active\ Power\ (t) = Mean\left(V(t) \times I(t)\right) \quad (3)$$

$$Apparent\ Power = V_{RMS} \times I_{RMS}$$

In the literature, signals are assumed to be undistorted sine waves. In this case, the power factor represents cos(Φ), where Φ is the phase difference between the voltage and the current channels. This definition is commonly used and accepted to define PF, but a measurement of PF should use the voltage rms, the current rms, and the mean active power values to cover any types of signals.

Reactive Energy

True energy is the quadratic sum of the active energy and the reactive energy. When the active power and the real power are known, the reactive power can be calculated

$$Reactive\ Power = \sqrt{Apparent\ Power^2 - Active\ Power^2} \quad (4)$$

Averaging

The calculation of the rms value is based on an averaging method. For a stabilized and unchanged signal, the more samples used for rms calculation, the more accurate it is. For changing inputs, the accuracy of the measurement depends on both measurement time averaging and time latency during which the input signal is not subjected to change. A compromise should be made between time averaging and information update rate.

For sine wave signals, amplitude and frequency are two possible changing parameters. An evaluation of the variation of these parameters in terms of dynamic range and frequency of variation should be done to determine the averaging time needed for an accurate rms measurement.

In our development, we made the following assumption:

Amplitude Dynamic Range:	Current 500:1 Voltage 10:1
Period of Change of the Amplitude: 3 Seconds	
Frequency of the Signal:	50 Hz ± 2% IEC1036 60 Hz ± 5% ANSI C12.16

Windowing Effect

The windowing effect affects many discrete time calculations (i.e., Discrete Fourier Transformation). This effect appears when the equivalent sampling time of the collected samples is not an exact integer number of cycles of a sine wave input.

A commonly used method for canceling the windowing effect is to apply a shaping window to the samples. This method diminishes the contribution of the beginning and ending samples to the signal processing and can be applied when the design uses a powerful DSP.

For the solution proposed, taking many samples and using averaging methods reduces the windowing effect. This solution is chosen because of the signal processing limitations of the 8-bit microcontroller.

Noise contribution to the rms calculation is independent of the chosen algorithm. Both solutions are sensitive to input noise, which should be addressed specifically.

RMS Calculation Errors Compensation

The expected outcome of an rms calculation is unbiased and stable information. Because signals and signal processing are not ideal, the behavior of the rms calculation can be affected by the offset in the input signal and the ripple effect of the multiplication.

For example, after the square operation, a sine wave signal with a small offset such as $V(t) = V_{OS} + V_{COS}(\omega t)$ yields:

$$V_{OS}^2 + V^2/2 + 2 \times V_{OS} \times V \times \cos(\omega t) + V^2/2 \times \cos(2\omega t)$$

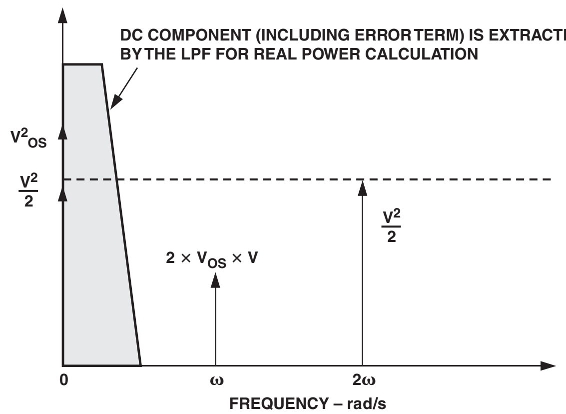

Figure 1. Effect of Input Offset on RMS Calculation

If the averaging is not perfect (the duration is too short), ripple frequencies (ω and 2 ω) components affect the result of the rms calculation. A low-pass filter at the output of the square function eliminates this ripple frequency noise. The cutoff frequency of the filter should be low enough to attenuate the ripple frequency by at least 40 dB (Figure 1).

The offset effect on the rms calculation is canceled by calibrating the offset error of the rms measurement and assuming that the offset is a constant variable in the signal processing. This offset calibration is also used to compensate the intrinsic noise integration behavior of the rms calculation.

The implementation of the rms calculation in a low cost microprocessor based design, such as in the AN-564 reference design, is discussed in the following paragraphs.

DESIGN CHOICES

ADE7756 Sampling Mode

The ADE7756 provides the output samples of the ADC through a special mode called waveform sampling mode. In this mode, Channel 1 or Channel 2 ADC samples are available in the waveform register (see the ADE7756 data sheet). The sampling rate of the output data is configurable in the mode register through four choices: CLKIN/128, CLKIN/256, CLKIN/512, and CLKIN/1024. After selecting the update rate and enabling the waveform interrupt in the interrupt enable register, the ADE7756 drives the $\overline{\text{IRQ}}$ pin low for 16 μs when a new sample is available in the waveform register. The microcontroller reads the waveform register and does the signal processing needed with this sample. When the sampling period is over, another interrupt appears to inform that the following sample is available.

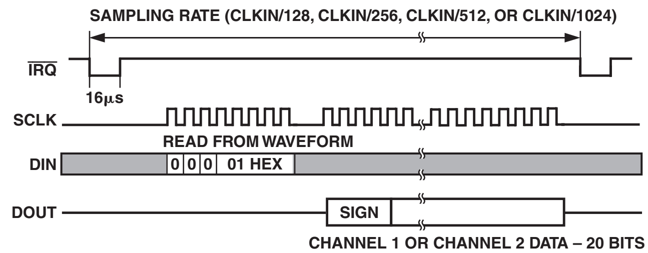

Figure 2. Waveform Sampling of Input Channels

Microcontroller Clock

The limitation of a standard 8-bit microcontroller in signal computation ability leads to the choice of the fastest operating clock possible for the microcontroller. This oscillator would be different from the ADE7756 oscillator. The maximum clock frequency for the microcontroller chosen in the AN-564 (PIC16C63) is 20 MHz.

As discussed later, the rms computation implemented in our solution uses a square routine and some real-time signal processing. These routines should be performed between each sample. A quick evaluation of the time needed to process 16-bit samples at 20 MHz leads to a worst-case computation time of approximately 350 μs.

The communication between the microcontroller and the ADE7756 can be sped up to 5 MHz (see the ADE7756 data sheet). Choosing the highest SPI clock frequency diminishes the time required to obtain data samples from the ADE7756.

Choice of the Sampling Clock

The ADE7756 provides samples of the voltage and the current inputs at a configurable sampling rate. The time needed for the microcontroller to process the data through the required rms signal processing determines the sampling frequency of the input.

The clock frequency of the ADE7756 is 3.579545 MHz. With this frequency, the sampling periods available through the ADE7756 are 35.7 μs, 71.5 μs, 143 μs, and 286 μs. Since the microcontroller needs more time to do the signal processing, the lowest sampling rate is used for the ADE7756 and the samples are read with a decimation factor of 2. The actual sampling rate for the samples used in the digital signal processing is then CLKIN/2048, which represents approximately 1.748 kHz.

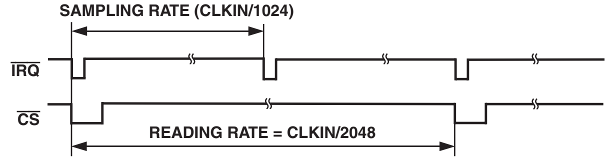

Figure 3. Samples Decimation

Samples Resolution

The ADE7756 provides samples of the input signals with a signed 20-bit resolution register (±262,144d). With the specified full-scale input signal of 1 V (or 0.5 V, 0.25 V, 0.125 V, or 0.0625 V), the ADC will produce an output code that is approximately 63% of its full-scale value (±165,151d). See the ADE7756 data sheet.

Because the microcontroller is a low power signal processor, the correct resolution of the data input in the signal processing should be chosen to reach both the accuracy needed and the minimum time computation. For example, only the absolute value is necessary in the rms calculation signal processing. Depending on the dynamic range necessary, a coarser resolution can be adopted.

The current input channel dynamic range is defined as 500:1, which means the maximum absolute value of the samples can vary from approximately 165,151 to 300. Dividing the input samples by four on the current channel gives a dynamic range of 41,288 to 82, sufficient to reach a 1% accuracy. This dynamic range translates into a 17-bit unsigned register.

The voltage input channel dynamic range is defined as 10:1, which means the maximum absolute value of the samples can vary from approximately 165,151 to 16,516. Dividing the input samples by 128 on the voltage channel will give a dynamic range of 1291 to 129, sufficient to reach a 1% accuracy. This dynamic range translates into a 12-bit unsigned register.

RMS Calculation

Figure 4 shows the algorithm implemented for the rms calculation.

The absolute value of the twos complement input is taken first to reduce the number of bits involved. Because the signal is squared afterwards, this block does not affect the final results.

The upper 12 bit of the voltage channel and 17 bit of the current channel are then extracted to be used in the rms algorithm. The signal is then squared and low-pass filtered to extract the dc component (see Figure 1). After the low-pass filter is settled down, the square root is taken and an offset compensation applied. A gain compensation and conversion to true rms information is then applied to enable the display of the actual value on an LCD display.

REAL-TIME SIGNAL PROCESSING

The algorithm described previously can be split into a real-time part and a post-processing part. The real-time part stands when each sample needs to be processed before another one happens. The post-processing part is defined as the moment when no new inputs are needed. The real-time signal processing comprises the squaring routine and the low-pass filter.

Squaring Routine

The square routine written in the program is a basic 24-bit by 24-bit squaring routine that provides a 6-byte result. The maximum output of the square routine is a 34-bit word (17-bit word squared). This limitation is very important because it reduces the computation time needed to propagate the carry-over of numerous bytes.

Averaging/Low-Pass Filter

Averaging the output of the squared input can be done by applying Equation 2 or by extracting the dc component of $V^2(t)$ with a low-pass filter (Figure 1). The low-pass filtering solution is implemented in the microcontroller with a first order infinite impulse response filter (IIR filter). The architecture of this filter is shown in Figure 5; the z-transfer filter function is described in Equation 5. This filter reduces the number of mathematical operations needed by using 2^{-P} gain parameters and by limiting the number of bits of the operands involved in the signal processing.

However, when the multiplication by 2^{-P} is done, the decimal bits of the result should be kept to avoid any digital oscillation or derivation in the filter response. This is achieved by multiplying the incoming samples by 2^{-P}.

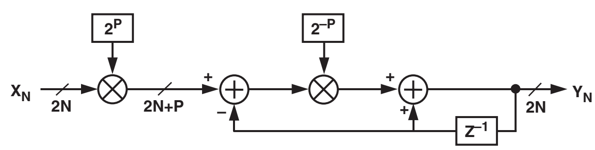

Figure 5. IIR Filter

$$\frac{Y_N}{X_N} = \frac{2^{-P}}{1-(1+2^{-P})Z^{-1}} \quad (5)$$

where 2^{-P} is the operand of the IIR filter.

The equivalent cutoff frequency of the IIR filter is

$F_C = \frac{2^{-P}}{2\pi} \times f_{SAMPLING}$, where 2^{-P} is the actual operand of the filter.

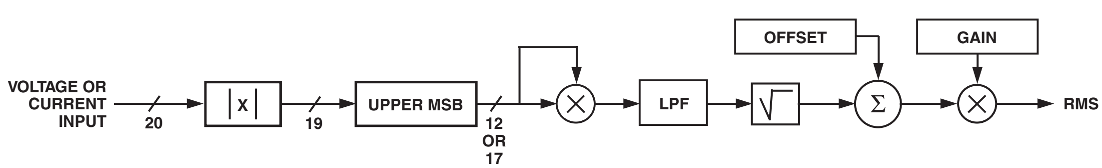

Figure 4. RMS Calculation Signal Path

In the solution proposed, the sampling frequency is 1.748 kHz (3.579545/2048 MHz) and p = 9. The magnitude response of this filter is shown in Figure 6.

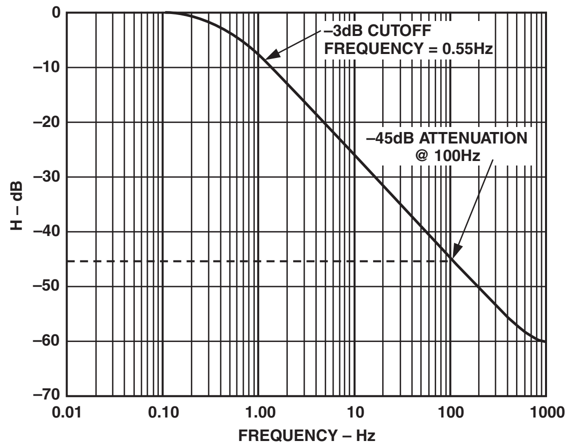

Figure 6. IIR Filter Magnitude Response

The IIR filter attenuates frequencies higher than 100 Hz by more than 45 dB, which means that the ripple frequency represents 0.56% of the output dc signal (see Figure 6). To decrease the effect of the ripple frequency, the cutoff frequency can be lowered but then it takes more samples and so more time for the LPF to settle down.

The settling time of the filter is also analyzed to determine the number of samples necessary to obtain a good accuracy. The step response of this filter is shown in Figure 7. The error between the output of the filter and the expected output is given by Equation 6: more than 3,540 samples are needed to obtain an error lower than 0.1%.

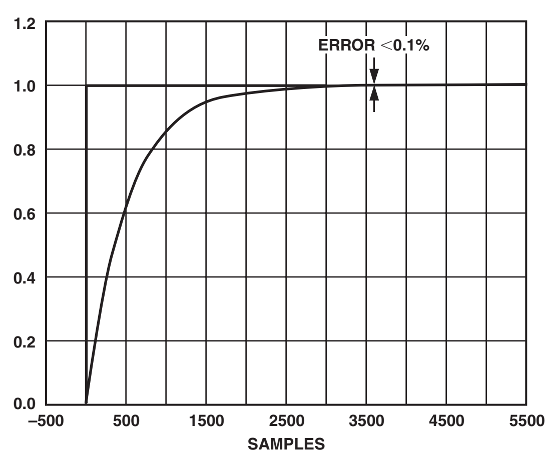

Figure 7. IIR Step Response

At the given sampling frequency, 3,540 samples represents 2.03 seconds.

$$Error\ (\%) = \frac{1}{\left(1+2^{-9}\right)^{n}} \times 100 \qquad (6)$$

where *n* represents the number of samples.

POST PROCESSING

When the output of the low-pass filter has settled, the square of the rms value of the input signal can be extracted. At this point, the microcontroller stops reading the samples from the ADE7756 and performs the post processing of the low-pass filter output. This post processing comprises a square root routine, an offset compensation, a gain compensation, and a binary-to-decimal conversion. It enables the display of the true rms value of the selected input on an LCD display.

Square Root Routine

Various algorithms exist to process the square root. A successive approximation (SAR) is implemented in this application note. The input is compared to the square of the output and the output, depending on the result of the subtraction, is incremented or decremented. The actual value of the low-pass filter is a 20-bit register; the square root of this value is reached after a loop of 20 iterations.

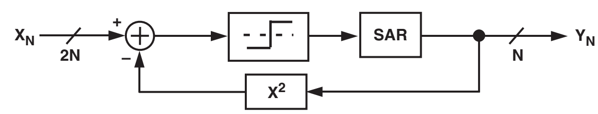

Figure 8. Square Root Algorithm

Offset and Gain Compensation

An offset compensation is introduced to eliminate the offset effect and the noise contribution in the rms value. The offset register is 16 bits wide.

The gain is applied to translate LSBs in volts rms or current rms. These values are then ready to be displayed on the LCD after a binary-to-decimal conversion.

Both offset and gain correction are current channel and voltage channel specifics.

Binary-to-Decimal Conversion

After the offset and gain compensation, the rms result is within the meaning scale: DCh represents 220 V rms. The LCD display used in the AN-564 reference design requires numbers in decimal formats. A binary-to-decimal conversion routine is then used to adapt the result of the digital signal processing to the format of the display.

FIRMWARE ARCHITECTURE

The solution provided in this application note should be compliant with the accuracy requirements of the rms calculation as well as the basic requirements of an electronics energy meter.

The accuracy requirements of the energy measurement itself are listed in the AN-564 application note and this solution complies with these specifications. Because the ADE7756 is used not only for energy accumulation, design issues should be addressed in the firmware to maintain the performance and behavior of the AN-564 reference design.

Energy Accumulation

When the ADE7756 is used in waveform sampling mode, the active energy register cannot be read at the same time. Nevertheless, the active energy is still accumulated in the ADE7756 active energy register and no energy is lost. The ADE7756 has a 40-bit energy accumulation register that is able to store more than 10 seconds of full-scale ac energy (see the ADE7756 data sheet). As the rms calculation does not last more than two seconds, the active energy register does not overflow. The active energy is read just before and after any rms calculation to assure that no energy loss occurs. This solution is shown in Figure 9. Normal mode stands for active energy accumulation alone without rms calculation.

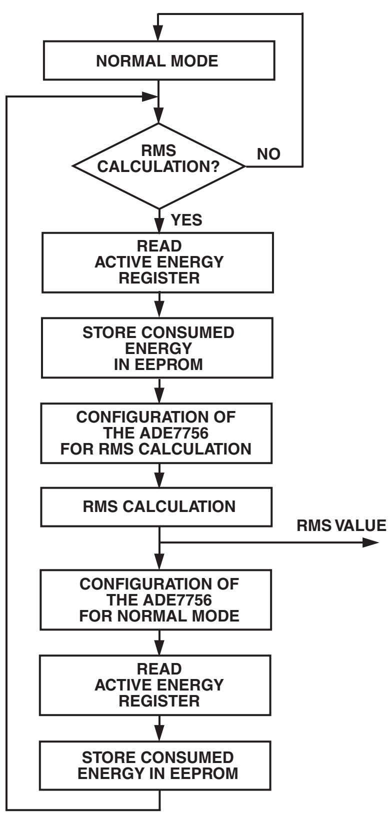

Figure 9. Active Energy Read Routine

Interrupt Servicing

An interrupt servicing routine takes care of all the timers' interrupts of the PIC16C63 as well as the external interrupts coming from the ADE7756.

In normal mode, the ADE7756 generates an interrupt only when the line voltage is below a predetermined threshold (SAG detection). The microcontroller services the interrupts (internal and external) as described in the AN-564 reference design.

In rms calculation mode, the ADE7756 generates an interrupt when one of two events occurs: SAG detection or waveform sample available. In this mode, the interrupt routine services only the interrupts coming from the ADE7756. When an interrupt occurs, the microcontroller checks the SAG pin input from the ADE7756 to determine if the interrupt corresponds to a low line voltage detection (SAG pin = low) or a waveform sample available (SAG pin = high). If a SAG event is detected, the rms routine in progress is abandoned to enter in a power-down mode. (Please refer to AN-564 documentation.) The power-down mode saves all the data prior to the loss of power. Data includes the accumulated energy stored in the microcontroller as well as the energy in the ADE7756's register. This functionality is very important to prevent any loss of energy in the event of loss of power. Figure 10 describes the servicing routine in rms calculation mode. The power-down mode is described in more detail in the AN-564 application note.

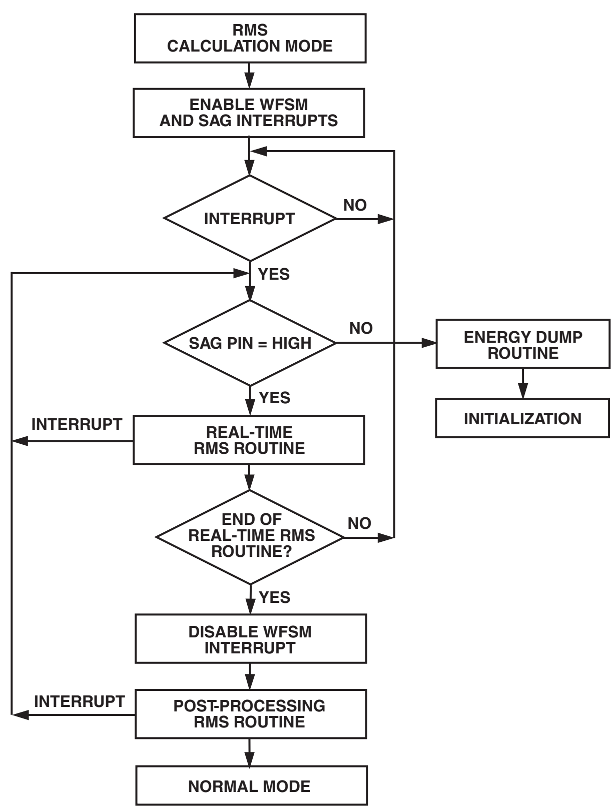

Figure 10. Interrupt Servicing Routine

RMS Calculation Routine

The rms routine is described in Figure 4. It should be pointed out that the rms routine can be interrupted at any moment by a decay of the power line. In this case, priority is given to the energy save routine to avoid any loss of consumed energy in the electronic meter. All the information of the interrupted rms routine will be lost and a complete recalculation should be done to get a new rms value.

Design Flexibility

The solution proposed offers some design flexibility. If the rms information is not needed every three seconds, the corner frequency of the low-pass filter can be changed to reduce the effect of the ripple frequency. The drawback of this change is that it increases the settling time of the low-pass filter and would alter the update frequency of the rms information.

The signal processing of the power factor, the apparent power, and the reactive energy can easily be done as the active energy and rms values are available. Equations 3 and 4 can be used and implemented in a microprocessor to deliver this information. There are no additional firmware routines necessary to process this information as multiplication, division, and square root routines are already included.

RESULTS

The solution described is implemented in the reference design AN-564 based on a microchip microcontroller.

4,096 samples are used in the algorithm. The linearity results over the different dynamic ranges have been evaluated and are shown in Figures 11 and 12. The error represented in these figures is the maximum error observed over 200 rms calculations. The percentage error is defined as

$$Percentage\,Error = \frac{True\,RMS - RMS\ Measurement}{True\ RMS} \times 100\%$$

The linearity is measured from 220 V to 22 V for the voltage input (10:1 ratio) and from 100 A to 200 mA for the current input (500:1 ratio). The calibration points used for rms calculation are 50 A and 200 mA for the current side and 220 V and 22 V for the voltage side.

The update rate that can be reached with this solution is approximately 3.5 seconds with a 20 MHz microprocessor clock. The real-time part of the algorithm represents 2.34 seconds and the post-processing part represents the remaining 1.16 seconds.

The total current consumption used by the energy meter during rms calculation is 10.2 mA with a 20 MHz oscillator, compared to 7 mA when the meter is in an energy measurement only mode with a 3.58 MHz oscillator.

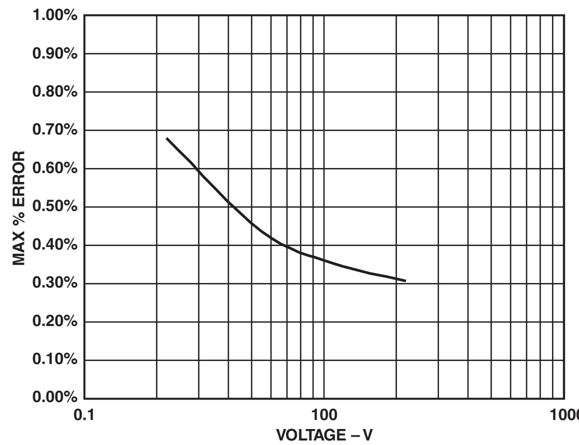

Figure 11. Voltage RMS Calculation Linearity

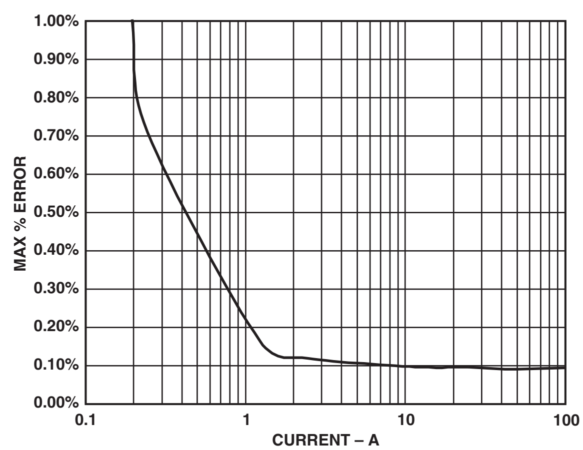

Figure 12. Current RMS Calculation Linearity

E02495–0–2/03(A)

PRINTED IN U.S.A.