Application Report

SLAA323A–July 2006–Revised July 2018

Oversampling the ADC12 for Higher Resolution

ABSTRACT

This application report describes the theory of oversampling. Oversampling can achieve resolutions greater than the available bits for an analog-to-digital converter (ADC). An example is shown that uses this technique for the ADC12 module of an MSP430™ microcontroller (MCU) to obtain greater than 12 bits of resolution. The example code and the Gerber files for the PCB can be downloaded from www.ti.com/lit/zip/slaa323.

Contents

Trademarks

MSP430 is a trademark of Texas Instruments.
All other trademarks are the property of their respective owners.

1 Introduction

Choosing the resolution for the integrated ADC in an MCU application involves a balance between cost and performance. Higher ADC resolution generally equates to higher cost. However, the MCU itself can enhance the performance of the ADC using software. These technique allows an inexpensive integrated ADC solution to achieve an improved resolution. Performance enhancements such as calibration, linearization, oversampling, and digital filtering can be achieved by software. This document describes how to use oversampling to achieve extra bits of resolution for the ADC12 module. The ADC12 module of the MSP430 MCUs that is used in the example is a 12-bit ADC with conversion speeds in excess of 200000 samples per second.

2 Resolution of an ADC

One ADC step in the digital domain is defined as 1 least-significant bit (LSB), and this is often used as the reference unit for ADC specifications. It is also the measure of the resolution of the converter, because it defines the number of counts for the full-scale analog input range.

Many applications require fine resolution to measure small changes in a parameter over a large input range. For example, an ADC may be required to detect a change of less than 40 µV over a range of 0 to 2500 mV. This requires at least 16 bits of resolution. The LSB in a 16-bit measurement would represent the voltage shown in Equation 1:

$$\text{Full-scale voltage range} / (2^{16} - 1) = (2.5\text{ V} - 0\text{ V}) / 65535 = 38\ \mu\text{V} \quad (1)$$

2.1 Signal-to-Noise Ratio (SNR)

For a waveform reconstructed from digital samples derived from the conversion, the signal-to-noise ratio (SNR) is the ratio (in dB) of the full-scale analog input RMS (root mean squared) value to the RMS quantization error. Increasing the effective resolution increases the signal-to-noise ratio of the conversion and vice versa. The theoretical limit of the SNR of an ADC measurement is based on the quantization noise that arises due to the quantization error inherent in the analog-to-digital conversion process. The SNR for an ideal ADC driven with a sine wave, whose peak voltage is the ADC full-scale input voltage, is given by Equation 2:

$$\text{SNR (dB)} = (6.02 \times N) + 1.76 \quad (2)$$

Where N is the number of bits representing the digital value.

The dynamic range of the input signal must match the full range of the ADC; otherwise, the SNR will be lower than calculated using Equation 2. For example, the SNR for an ideal 12-bit ADC is 74 dB. Equation 2 can also be used to reverse calculate the number of ADC bits required to achieve a given SNR.

Due to the presence of quantization noise, thermal noise, reference noise, clock jitter, and so on, an N-bit ADC has an effective number of bits (ENOB) that is less than N. The ENOB can be characterized by sampling a pure sinusoidal input and performing an FFT on the collected data. The signal-to-noise and distortion ratio (SINAD) is the ratio of the magnitude of the fundamental frequency to the RMS of all other frequencies, including harmonics. ENOB can be calculated by replacing SNR in Equation 2 with the SINAD, which includes the distortion noise, and N with ENOB. The SINAD and SNR can be obtained from the above mentioned dynamic FFT testing.

2.2 Improving the Resolution of an ADC

Oversampling is a popular method to improve ADC resolution. The input is sampled at a rate higher than the minimum required Nyquist sampling rate, f_s. For example, when using an N-bit ADC without oversampling, an input signal of 100 Hz is sampled at 200 Hz (2 × 100 Hz) to get the digital output with the native ENOB of the ADC. When oversampling with a factor of k = 16, the same 100-Hz input signal is sampled at 3200 Hz (k × 2 × 100 Hz). The samples obtained by oversampling are low-pass filtered and decimated using a digital filter to achieve a reduction of the quantization noise. The signal at the frequency band of interest is not affected by the filter, and the result is an improved SNR. The improved SNR results in a higher ENOB performance. Equation 3 shows the relationship between improved SNR, N, and the oversampling factor, k.

$$SNR\ (dB) = (6.02 \times N) + 1.76 + 10 \times \log_{10}(k)$$

where

- $k = f_s / (2 \times f_{max})$
- f_s is the sampling frequency
- $2 \times f_{max}$ is the Nyquist frequency (3)

Figure 1 shows the signal flow diagram for the oversampling method. The quantization noise is modeled as white noise that is additive to the input signal while sampling. Oversampling using white noise provides about 3 dB or half bit of resolution gain for each doubling of the oversampling rate. To obtain 16 bits of resolution with this method using a 12-bit ADC, the oversampling factor required is 256.

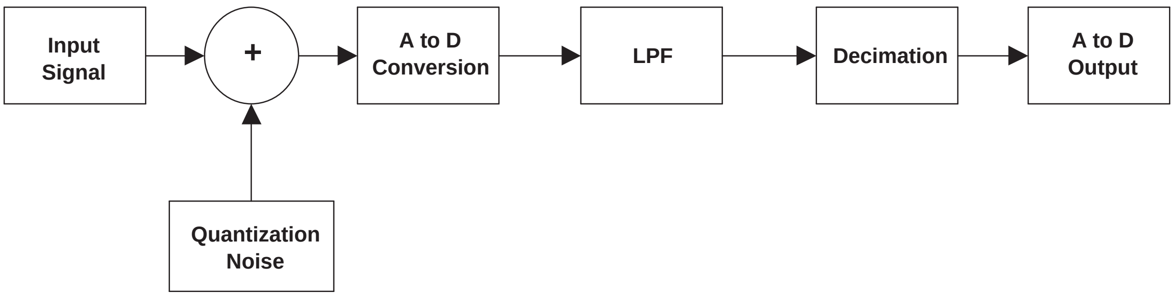

Figure 1. Signal-Flow Diagram For Oversampling Method

Table 1 lists the relationships among oversampling factor k, SNR, and the extra bits of resolution that can be achieved.

Table 1. Relationship Between Oversampling Factor, SNR and Extra Bits of Resolution

Oversampling Factor, k	SNR Improvement (dB)	Extra Bits of Resolution
2	3	0.5
4	6	1
8	9	1.5
16	12	2
32	16	2.5
64	18	3
128	21	3.5
256	24	4
512	27	4.5
1024	30	5
2048	33	5.5
4096	36	6

3 Demonstration Application

3.1 Circuit Description

The demonstration board for this application uses a MSP430FG439 device with an LCD display powered using a 3-V lithium battery. The Gerber files and schematic for the PCB used to demonstrate this application are available in a zip file associated with this application report. Resistors R1 and R2 and capacitors C1, C2, C6, and C7 are used as the RC filter for AVCC and DVCC to filter the noise on the MCU power supply. Such filtering is recommended for improved analog performance. Capacitors C4 and C11 are the storage capacitors across the V_{ref}, which are required to supply the additional current during a conversion.[2] See the schematic for these components.

3.2 Software Operation

The board has four modes of operation and a calibration mode. Each mode displays the oversampled and averaged value every 250 ms on the LCD. The ADC12 is continuously converting, and samples are averaged using a 256-tap moving average filter in the FIR structure within the 250-ms period.

3.2.1 100-µV Mode

This is the default mode of the board when it is powered up. The voltage is displayed with a resolution of 100 µV. To enter this mode when in any other mode, press Switch1 (SW1). The input value can be changed using the on-board potentiometer or an external voltage source to observe the corresponding change in the displayed value.

3.2.2 Temperature Mode

To enter this mode press Switch 2 (SW2). The temperature is displayed with a resolution of 0.01°C.

NOTE: This is not the accuracy of the on-chip temperature sensor. This mode is used only to demonstrate the resolution achieved by oversampling.

3.2.3 16-Bit Voltage Measurement Mode

To enter this mode press Switch 3 (SW3). The voltage is displayed with 16-bit resolution. The potentiometer or an external voltage source can be used to change the input voltage to observe the results.

3.2.4 OFF Mode (LPM4)

While in the 100-µV mode, press SW1 to enter this mode. To return to the 100-µV mode, press SW1 again. Pressing SW2 or SW3 has no effect. In this mode, the LCD and all clocks are disabled, and the device is put to LPM4 mode.

3.2.5 Temperature Calibration

1. Press and hold SW2 while the board is powered up.
2. When the LCD displays voltage, press SW2 to enable temperature calibration.
3. The LED blinks to indicate that the user is in temperature calibration mode. The LED stays on (instead of blinking) if the initial temperature display on the LCD is greater than 79°F.
4. Press SW1 or SW2 to calibrate the temperature sensor offset, to display a known temperature reading in your area.
5. Initially, pressing SW1 or SW2 adjusts the second digit after the decimal.
6. Pressing SW1 and SW2 together for one second enables the user to change the first digit after the decimal.
7. Pressing SW1 and SW2 together again for one second enables the user to change the temperature in jumps of 1°F.

8. Pressing SW1 and SW2 together again for one second returns the user to step 5.
9. Then, upon pressing SW3 for one second, the calibrated values are stored in flash, and the device functions in the three modes described above.

3.2.6 Reference Calibration

1. Press and hold SW2 while the board is powered up.
2. When the LCD turns on, release SW2.
3. Press SW3. The device now enters Reference Voltage calibration mode.
4. The LED blinks to indicate this.
5. Adjust the potentiometer to display the Vref voltage on the LCD.
6. Again, SW1 and SW2 can be used to calibrate the reference voltage to match the voltage displayed on the LCD to a known Vref reading, as measured with an accurate voltmeter. Reference voltage (Vref) can be measured on header J3, pin 10.
7. Pressing SW3 again stores the calibrated value in flash, and the device functions in the three modes previously described.

4 Software Description

Upon execution of the code after a reset, a low_level_init and init_sys routines are executed to stop the watchdog, initialize the ports, LCD, and the basic timer. The ADC12 is set up in repeat single-channel mode and Timer_B is used as the sampling timer, causing a sample and convert every 390 µs.

4.1 Main()

The main loop is called every 250 ms to determine the function to be called (100-µV mode, temperature mode, or 16-bit voltage-measurement mode) depending on the switch press, as described in Section 3.2. An interval of 250 ms is used for the LCD to update at a rate that allows the user to easily see the voltage resolution.

4.2 Voltage2()

This function switches the device between 100-µV mode and OFF mode. It sets up the ADC12 to take multiple samples and convert on channel 0 using the internal 2.5-V reference.

4.3 Temperature()

This function is used for taking a temperature measurement using the on-chip temperature sensor and displaying it on LCD. It sets up the ADC12 to take multiple samples and convert on channel 10 using the internal 1.5-V reference.

4.4 Voltage()

This function is used for the 16-bit voltage measurement mode. It sets up the ADC12 to take multiple samples and convert on channel 0 using the internal 2.5-V reference.

4.5 check_cal()

This functions checks to see if flash information memory is blank or if it contains calibration constants. If erased, appropriate values are put in calibration constants.

4.6 Temp_cal()

This function implements the temperature calibration by incrementing or decrementing the offset and storing the calibration constant in flash after it is set.

4.7 Ref_cal()

This function implements the reference voltage calibration by incrementing or decrementing the “refcal” constant and storing the calibration constant in flash after it is set.

5 Results

Figure 2 and Figure 3 show charts of DC input voltage, incremented in steps of 1 LSB. 1-LSB steps are provided by a precision 16-bit DAC. The Ideal values for a 16-bit converter and a 12-bit converter are compared with the measured oversampled values. Analog input voltage is incremented in steps of 1 LSB. Figure 2 data is taken over the entire range of 2^{16} voltage values. Figure 3 is a zoomed-in snapshot of the data, which simplifies looking at the chart. An Excel spreadsheet is available in the zip file associated with this application report and contains the entire data set.

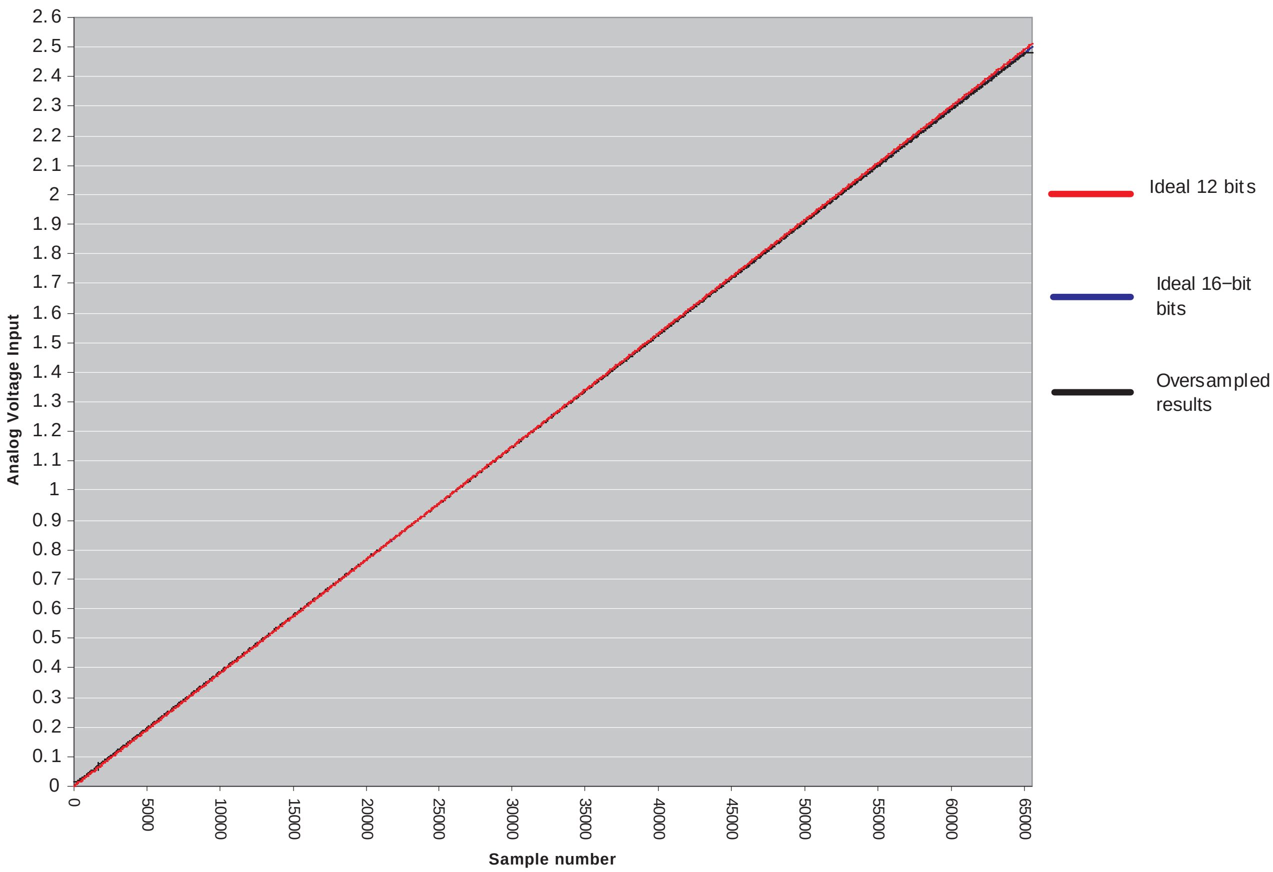

Figure 2. Comparison of Oversampled Data With Ideal Data Over Entire Range

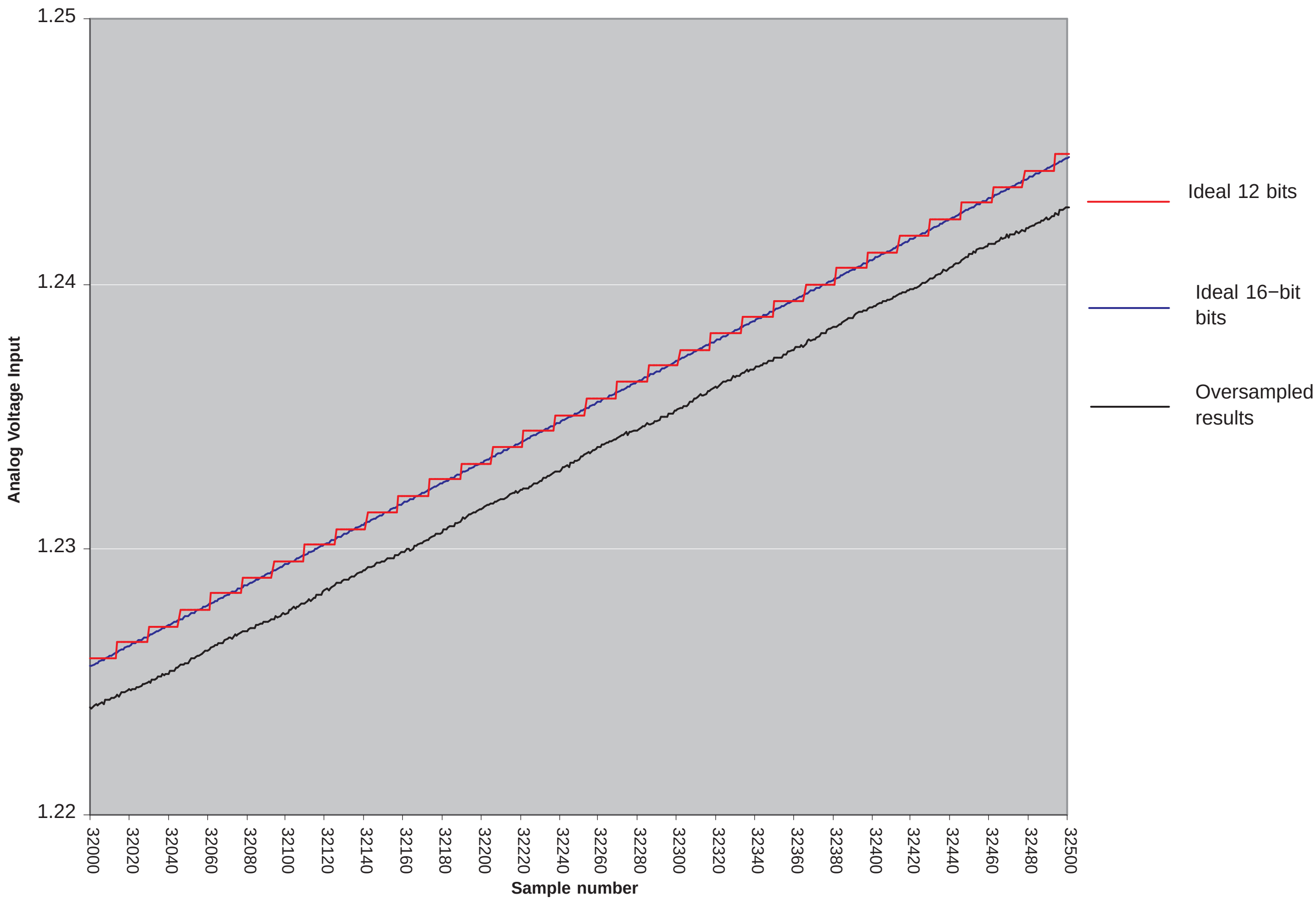

Figure 3. Comparison of Oversampled Data With Ideal Data (Zoomed In)

6 Conclusion

This application report describes how to achieve higher resolution from ADC12 using the oversampling method. Oversampling is a method of achieving higher effective resolutions while lowering the constraints of the analog anti-aliasing filter, by implementing a digital filtering and decimation process. For example, in the case of a time-varying input signal like a sine wave, a dither signal can be added to the input to improve resolution. This approach is discussed in Oversampling Techniques using the TMS320C24x Family. This solution could be achieved by using the MSP430FG439 with its integrated operational amplifiers, timer, and DAC12.[2]

Good layout practices and proper power-supply decoupling improve the analog performance of any high-resolution system. Decoupling filters should be placed as close to the supply as possible.

Oversampling results depend on the quantization noise of the ADC and, hence, may vary between devices. As seen from the results, oversampling technique helps to achieve greater than 12 bits of resolution using the ADC12.

7 References

1. Oversampling Techniques using the TMS320C24x Family
2. MSP430x4xx Family User's Guide
3. General Oversampling of MSP ADCs for Higher Resolution

Revision History

NOTE: Page numbers for previous revisions may differ from page numbers in the current version.

Changes from July 13, 2006 to July 12, 2018	Page
• Editorial updates throughout document	1
• Added link to associated zip file in abstract	1
• Added item (3) in Section 7, *References*	7

IMPORTANT NOTICE FOR TI DESIGN INFORMATION AND RESOURCES

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265