Application Report
SLAA694A–February 2016–Revised April 2016

General Oversampling of MSP ADCs for Higher Resolution

Ryan Brown and Sameer Singh MSP Applications

ABSTRACT

Multiple MSP ultra-low-power microcontrollers offer analog-to-digital converters (ADCs) to convert physical quantities into digital numbers, a function that is widely used across numerous applications. There are times, however, when a customer design demands a higher resolution than the ADC of the selected MSP can offer. This application report, which is based on the previously-published *Oversampling the ADC12 for Higher Resolution* (SLAA323), therefore describes how an oversampling method can be incorporated to increase ADC resolution past the currently available number of bits. Code example projects (available for download from http://www.ti.com/lit/zip/slaa694) and detailed instructions are provided for the MSP430FR4133 and MSP432P401R microcontrollers but can be modified so that oversampling is supported for a myriad of MSP MCU derivatives, sampling frequencies, and extra bits of resolution.

Contents

List of Figures

List of Tables

Trademarks

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

1 Introduction

Most often an MSP microcontroller from the device portfolio can be selected that meets all of the application requirements, including footprint size, memory capacity, peripherals included, and cost. But then it can sometimes be disclosed that the analog-to-digital converter (ADC) resolution is lower than desired, a requirement which may be critical enough to make the user consider other options that do not fit their design needs as adequately. An alternative to this outcome would be incorporating a software-based oversampling method to obtain extra bits of ADC resolution. The following sections of this application report examine the theory behind oversampling and discuss in detail how software algorithms are employed to achieve the extra resolution desired. An example project that has been designed for operation across various ultra low-power MSP MCU modules, sample frequencies, and desired extra bits of resolution is also provided to instruct and advise the user regarding how their current application can be modified to integrate the ADC oversampling solution with ease.

2 Oversampling Process

This section describes the fundamentals of ADC operation and discusses the equation involved in calculating the signal-to-noise ratio (SNR) given the number of bits (N) available from the ADC. This section also describes the relationship between this data and the signal-to-noise and distortion ratio (SINAD) as well as the effective number of bits (ENOB). Implementation of the oversampling method to improve ADC resolution is then detailed. All information provided is sourced from *Oversampling the ADC12 for Higher Resolution* (SLAA323).

2.1 ADC Operation

N-bit ADCs have 2^N number of steps that can be used across a large input range to measure small changes in a parameter. The input range itself depends on the minimum and maximum references selected, V_{R-} and V_{R+}, which are determined by the MSP ADC registers. The expected resolution, ADC_{RES}, can therefore be calculated as shown in Equation 1:

$$ADC_{RES} = (V_{R+} - V_{R-}) \div 2^N \tag{1}$$

For analog waveforms that are being reconstructed from the digital samples taken using such a conversion, the SNR is the ratio in dB of the full-scale analog input root mean squared (RMS) value to the RMS quantization error. Increasing the resolution increases the SNR and vice versa, where the theoretical limit of the ADC SNR comes from quantization error inherent to the conversion process. The SNR for an ideal ADC driven with a sine wave whose dynamic range eclipses the full range of V_{R-} to V_{R+} is found using Equation 2:

$$SNR_{dB} = (6.02 \times N) + 1.76 \tag{2}$$

In this equation, SNRDB increases by $20 \times \log_{10}(2) = 6.02$ with every step increase in N, and 1.76 is a constant caused by the characteristics of a sine wave. Due to the presence of quantization, thermal, or reference noise and clock jitter, an N-bit ADC can be expected to have an ENOB less than N. The ENOB can be characterized by sampling a pure sinusoidal input and performing an FFT on the collected data. This can then be applied to Equation 2 by replacing N to calculate the effective SINAD in place of the SNR.

2.2 Improving ADC Resolution

Oversampling is a widely popular method for improving ADC resolution. The ADC converts continuous time input signals to a discrete time output. To avoid aliasing and maintain signal fidelity, the ADC sampling rate, F_S, needs to be at least twice the maximum frequency information present in the input F_{IN}, also referred to as the Nyquist theorem. Using any sampling rate above this frequency is considered to be oversampling, and its ratio to F_{SNYQ} is called the oversampling ratio (k), shown in Equation 3:

$$k = \frac{F_S}{F_{SNYQ}} \tag{3}$$

The effect of oversampling on the noise can be seen clearly in the frequency domain (see Figure 1). When the input is sampled, the entire continuous domain information is translated to and contained within a frequency band of $\pm F_S/2$. The total noise power from contributors of white noise, such as thermal and quantization sources, thereby reduces the power spectral density or noise floor in the ADC output's FFT.

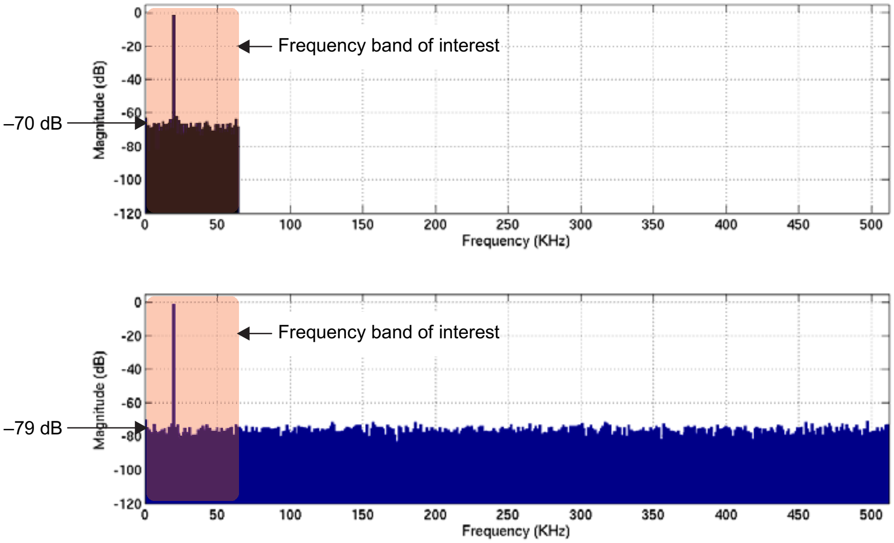


Figure 1. Noise Floor Comparison of ADCOUT With Sampling Rate of 128 and 1024 kHz

By using a digital filter, it is possible to reduce the quantization noise, and the resulting SNR is improved because the signal at the frequency band of interest is not affected by the filter. For every octave increase in sampling frequency the noise floor can be seen to go down by ~3 dB. Hence if only the noise of the signal band of interest is considered, SNR can be improved as such:

$$\mathrm{SNR_{OSR}} = \mathrm{SNR_{NYO}} + 10 \times \log_{10}(k) \tag{4}$$

FromEquation 3 and Equation 4, the required sampling rate for a desired increase of β in ENOB is:

$$F_S = 4^{\beta} \times F_{SNYO} \tag{5}$$

Figure 2 shows the signal flow diagram for the oversampling method. Quantization noise is modeled as white noise added to the input signal while sampling, oversampling in turn provides approximately 3 dB (or a half bit of resolution gain) for each doubling of k. The value of k required is therefore determined by taking four to the power of the extra bits of resolution desired.

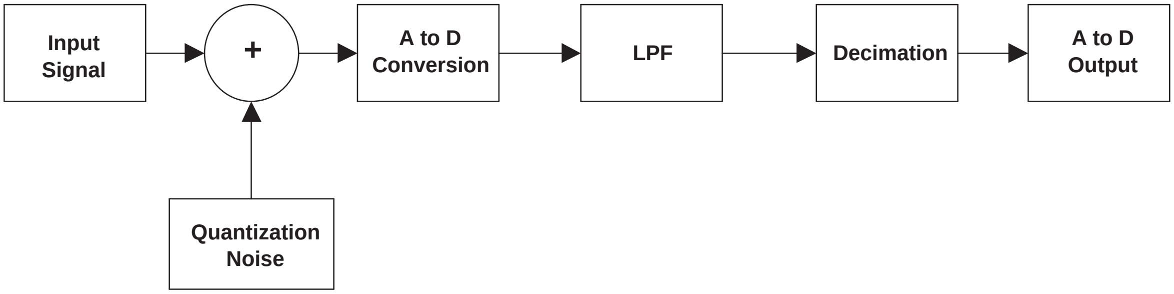


Figure 2. Signal-Flow Diagram for Oversampling Method

Table 1 describes the relationship between k, SNR, and the achievable extra bits of resolution.

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 Hardware Description

This application report comes with an example project that tests the oversampling process with two different ADC peripherals from distinctive MSP devices: a 10-bit ADC on the MSP430FR4133 and 14-bit ADC on the MSP432P401R. Meaningful analog input data is provided by connecting to and communicating with a DAC8560EVM board, whose description and features are summarized in Section 3.1. The appropriate connections between the DAC8560 and each LaunchPad™ development kit, the MSP-EXP430FR4133 and MSP-EXP432P401R, are also described in this section.

3.1 DAC8560EVM

The DAC8560EVM is an evaluation module designed to easily evaluate the functionality and performance of the single-channel serial input DAC with up to 16-bits of resolution. The device has a 2.5 V internal reference that is enabled by default and can be output to a MSP ADC external reference pin. The EVM has a serial interface to allow for communication with a host Microprocessor, including any of the numerous LaunchPad development kits supplied by TI, and can furthermore be solely supplied by the MSP V_{CC} rail. Table 2 lists the proper DAC8560EVM jumper orientations needed to run the example as intended. More information about the DAC can be found in the *DAC8560EVM User's Guide* (SLAU211).

Table 2. DAC8560EVM Jumper Orientation

Jumper	Orientation
W1	AVDD to +3 VA
W2	1 to 2
W3	Open
W4	Open
W5	2 to 3
W6	1 to 2
W7	Open
W8	Open
W15	Open
J6	5 to 6

3.2 *MSP-EXP430FR4133 and MSP-EXP432P401R Connections*

The example MSP430FR4133 firmware utilizes A3 as the ADC input to be oversampled, a 2.5-V Veref+ sourced from the DAC8560, UCB0 SPI for controlling DAC output values, and UCA0 UART to update the host PC with steps and ADC readings for further analysis. The MSP-EXP432P401R likewise uses A5, Veref+, UCA1 SPI, and UCA0 UART for the same purposes. More information on firmware specifics is provided in Section 4. Further details regarding each LaunchPad development kit can be found in their respective user's guides, the *MSP430FR4133 LaunchPad Development Kit (MSP-EXP430FR4133) User's Guide* (SLAU595) and the *MSP432P401R LaunchPad Development Kit (MSP-EXP432P401R) User's Guide* (SLAU597). Table 3 lists all essential connections between the MSP-EXP430FR4133 or MSP-EXP432P401R and DAC8560EVM for operating the demo.

Table 3. LaunchPad Development Kit Hardware Connections

DAC8560EVM	MSP-EXP430FR4133	MSP-EXP432P401R
J6 1 (VCC)	3V3	3V3
J2 11 (SDI)	5.2 (UCB0SIMO)	2.6 (PM_UCA1SIMO)
J2 3 (SCLK)	5.1 (UCB0CLK)	2.5 (PM_UCA1CLK)
J2 1 (CS)	5.0 (UCB0STE)	2.4 (PM_UAC1STE)
W4 2 (VREF)	1.1 (Veref+)	Veref+
J4 2 (VOUT)	1.3 (A3)	5.0 (A5)
J2 18 (GND)	GND	GND
J6 8 (+3.3VA)	3V3	3V3

4 Firmware Overview

Demonstration code comes with this application report in the associated zip file. The code is designed to function as follows: after push button S1 is pressed to start the program, a SPI command is sent from the MSP to the DAC8560 to set the DAC output level sourcing the ADC channel of the MSP microcontroller. After employing the oversampling method on a given number of input readings, the MSP then sends the ASCII character representations of both the expected DAC output step and the resultant ADC oversampling value to a host PC terminal (such as Tera Term) through UART communication. The DAC output level is then incremented according to the extra bits of resolution desired and the process repeats until the full range of the ADC reference (from 0 to 2.5 V) is covered. When this process is finished, this data can then be extracted from the terminal program and imported into a numerical computing environment program, such as Microsoft Excel or MATLAB, for further analysis. Figure 3 shows a software flow diagram of this procedure.

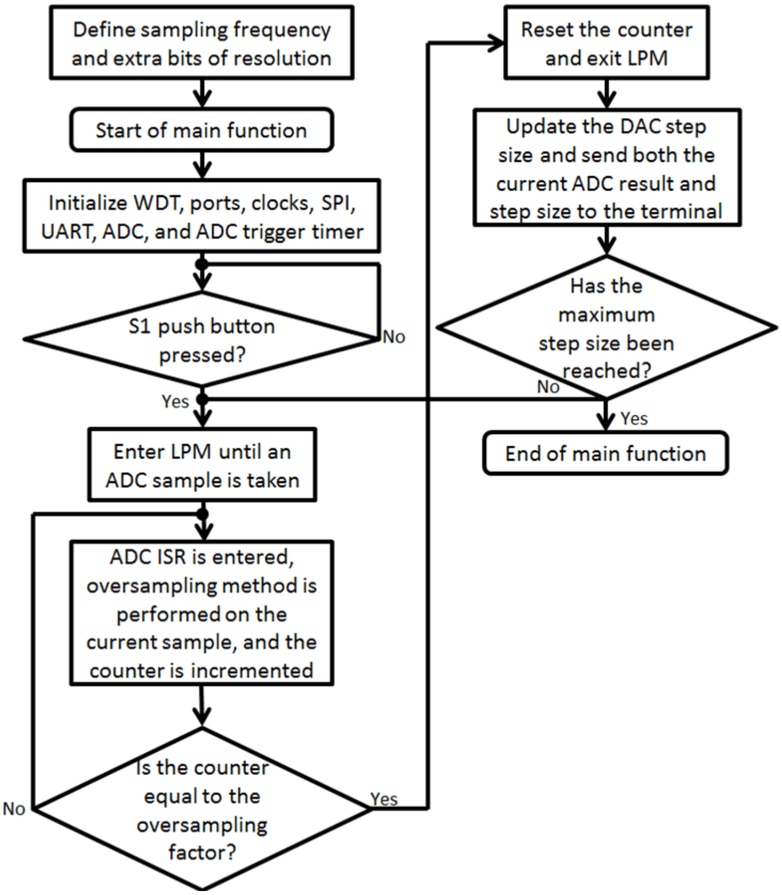

Define sampling frequency and extra bits of resolution
Start of main function
Initialize WDT, ports, clocks, SPI, UART, ADC, and ADC trigger timer
S1 push button pressed?
No
Yes
Enter LPM until an ADC sample is taken
ADC ISR is entered, oversampling method is performed on the current sample, and the counter is incremented
Is the counter equal to the oversampling factor?
No
Yes
Reset the counter and exit LPM
Update the DAC step size and send both the current ADC result and step size to the terminal
Has the maximum step size been reached?
No
Yes
End of main function

Figure 3. ADC Oversampling Demonstration Software Flow Diagram

4.1 Setting Constants

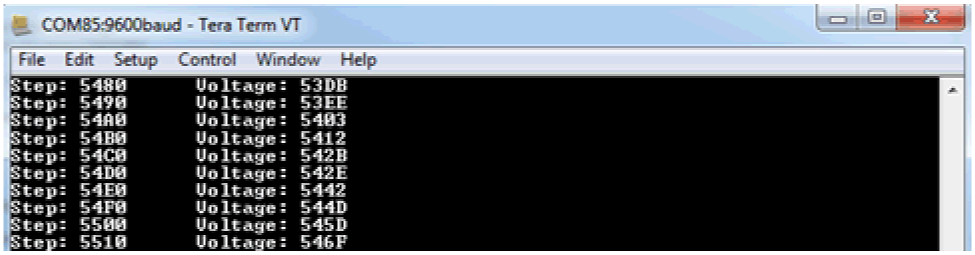
The firmware has been designed so that the user can easily change the value of a few constants to adjust the amount of test data taken and speed at which the data is produced. To alter the needed extra bits of resolution, modify the *extrabits (β)* constant along with the size of *AVGBuffer*, whose value should reflect four to the power of *extrabits*. The *period* constant determines the oversampling frequency and hence the sampling speed. This value follows the ADC trigger timer sourced from ACLK, depends on *extrabits* and the chosen sampling frequency f_{samp}, and can be calculated by Equation 6.

$$\text{period} = \text{ACLK} \div (2 \times 4^{\text{extrabits}} \times f_{\text{samp}}) \quad (6)$$

Depending on the value of *extrabits*, the expected sampling frequency may not be achievable. To increase f_{samp}, either *extrabits* can be decreased or the ADC trigger timer can be sourced by a faster clock source. This may come in the form of selecting a different ACLK source, using an external crystal with a higher frequency, or sourcing the timer with a clock faster than ACLK.

4.2 Running the Demonstration

To view the example code performing as intended, first program the LaunchPad development kit using the corresponding CCS project. Then, after ensuring that power is not supplied to the LaunchPad development kit, make the necessary connections to the DAC8560EVM (see Table 2 and Table 3). When finished, connect the LaunchPad development kit to the host PC through the micro-USB connector. Open the selected terminal program and select the Serial COM Port that corresponds to the LaunchPad development kit Application/User UART. After setting the communication speed to 9600 baud, 8 bit data, and 1 bit stop with no parity, press the S1 push button on the LaunchPad development kit and watch the terminal log steps from 0x0000 to 0xFFFF (see Figure 4). When complete, the information can be saved to a text file or other format and used in a separate software program for interpretation.



```
Step: 5480      Voltage: 53DB
Step: 5490      Voltage: 53EE
Step: 54A0      Voltage: 5403
Step: 54B0      Voltage: 5412
Step: 54C0      Voltage: 542B
Step: 54D0      Voltage: 542E
Step: 54E0      Voltage: 5442
Step: 54F0      Voltage: 544D
Step: 5500      Voltage: 545D
Step: 5510      Voltage: 546F
```

Figure 4. Using a Terminal Program to Actively Log ADC Oversampling Results

4.3 Oversampling Method Extraction

Although useful for demonstrating the capabilities of the ADC oversampling method, it is understood that many applications do not need the extra functionality of the demonstration or would like to use a different analog front end. For these instances, the process of extracting the oversampling technique from the example CCS projects is straightforward. All initialization, subfunction, and main code commands related to USCI peripherals can be removed. This leaves the crystal and clock initialization, ADC setup, ADC trigger timer configuration, and ADC interrupt intact, all of which are all required for oversampling. The analog input channel and ADC reference can be changed to fit the end application but require changes to the respective registers.

5 Results

5.1 MSP-EXP430FR4133 and DAC8560 Oversampling

Figure 5 through Figure 7 show results from the example code, in which the oversampling method is implemented on the MSP430FR4133 10-bit ADC to produce 12-bit and 15-bit resolutions. 10-bit results have also been supplied as a control. Although the test was run from 0 to 2.5 V, the graphs are limited between the equivalents of 2.2 V to 2.3 V to show the variation in precision. These results are similar to those found in *Oversampling the ADC12 for Higher Resolution* (SLAA323). Results for the MSP432P401R were also similar and therefore are not included.

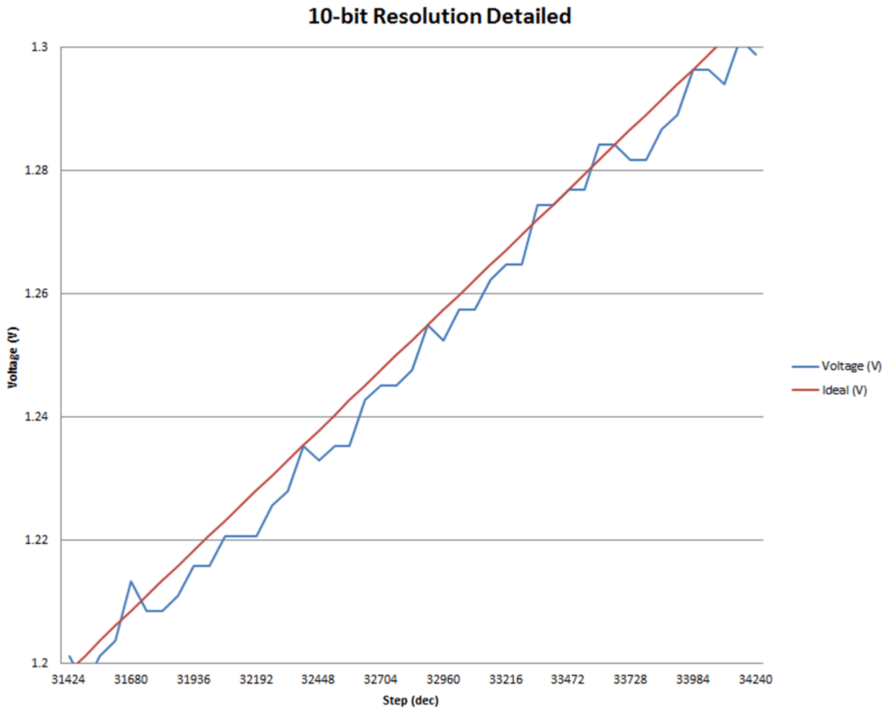


Figure 5. 10-Bit Resolution by Oversampling the MSP430FR4133 ADC

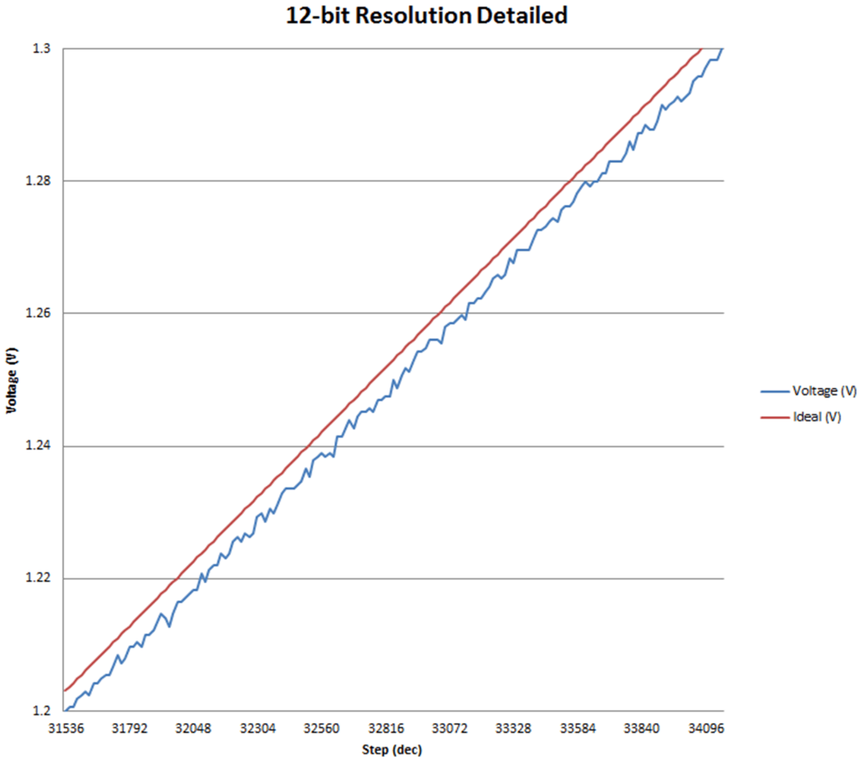


Figure 6. 12-Bit Resolution by Oversampling the MSP430FR4133 ADC

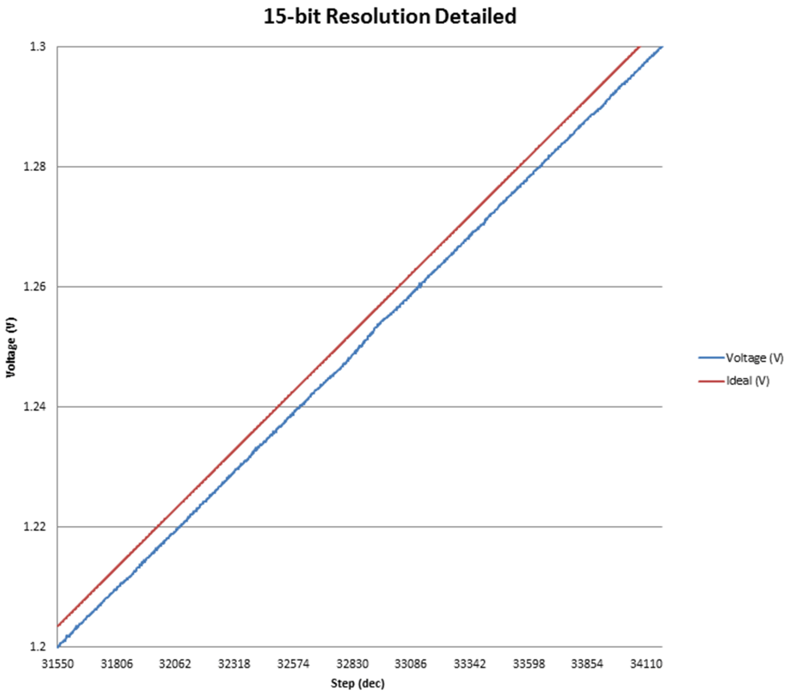


Figure 7. 15-Bit Resolution by Oversampling the MSP430FR4133 ADC

Figure 6 shows the full spectrum of results for one of the oversampling example runs, in which two extra bits of resolution are produced to create a 12-bit ADC for the MSP430FR4133. A noticeable gap between the ideal and recorded voltages can be seen, and this gap is due to offset generated by the DAC8560. To compensate for the offset, add a constant value to the ADC results as shown in Figure 9 (the full spectrum of 12-bit resolution is shown in Figure 8). Offset separation varies depending on the end application and the analog device that is used.

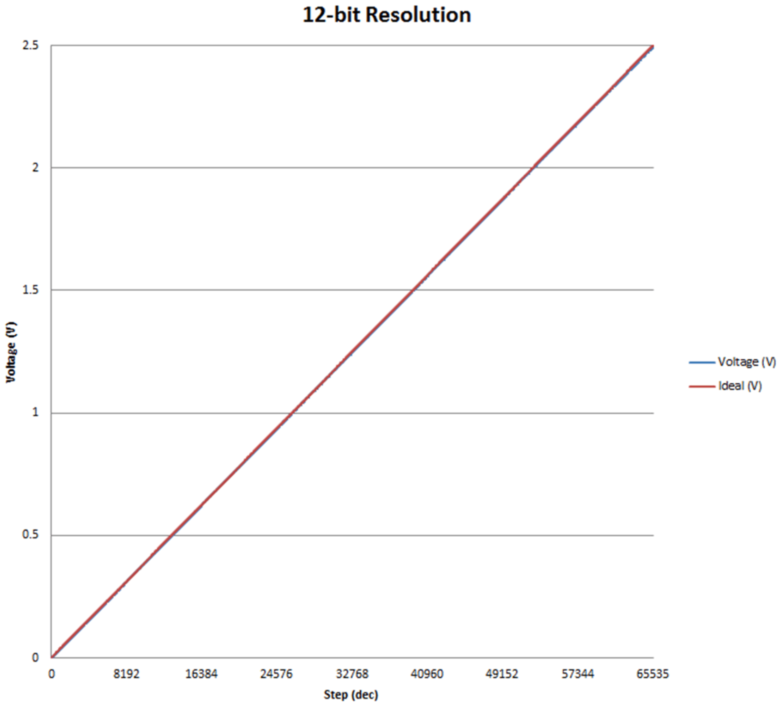


Figure 8. Full Spectrum of 12-Bit Oversampling Results on the MSP430FR4133 ADC

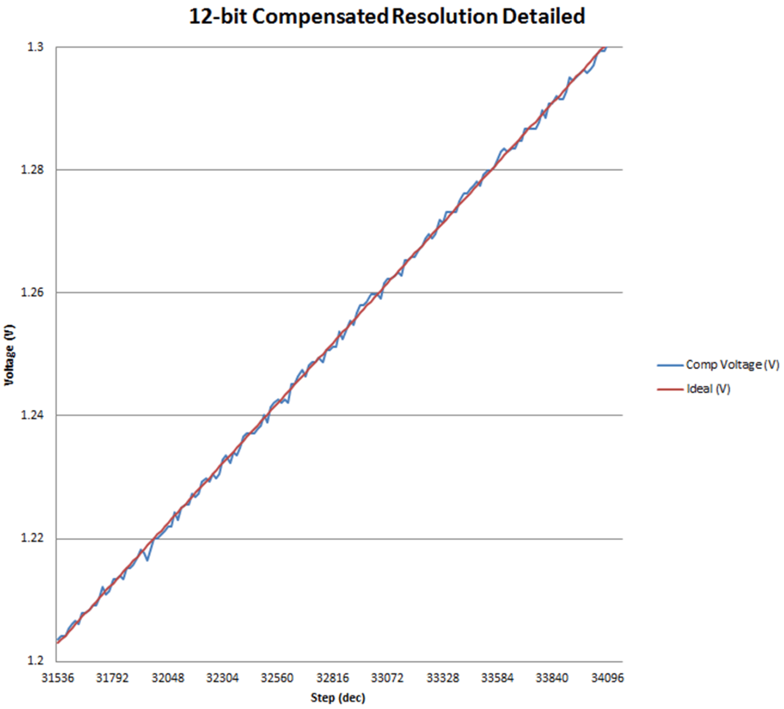


Figure 9. Compensated 12-Bit Oversampling Results on the MSP430FR4133 ADC

5.2 *Silicon Results of Oversampling and Decimation on MSP432P401R*

A simple rolling average filter was implemented as an interrupt subroutine in an MSP432P401R device to characterize the improvement in ENOB. Figure 10 and Figure 11 show the ADC performance is plotted across different F_{IN}, F_S, and k values. With higher OSRs and larger averaging window size, the ADC performance keeps improving. The data only discontinues this trend when the implemented 3-dB bandwidth of the decimation filter becomes lower than the signal frequency itself. It can also be seen that 2.5- and 1.5-dB improvement in ENOB can be achieved at 2- and 20-kHz F_{IN}, respectively. The ADC ENOB was pushed to as high as 15.8 bits by using an OSR of 64 at 1-MSPs sampling rate while maintaining the throughput of data at 1 MHz.

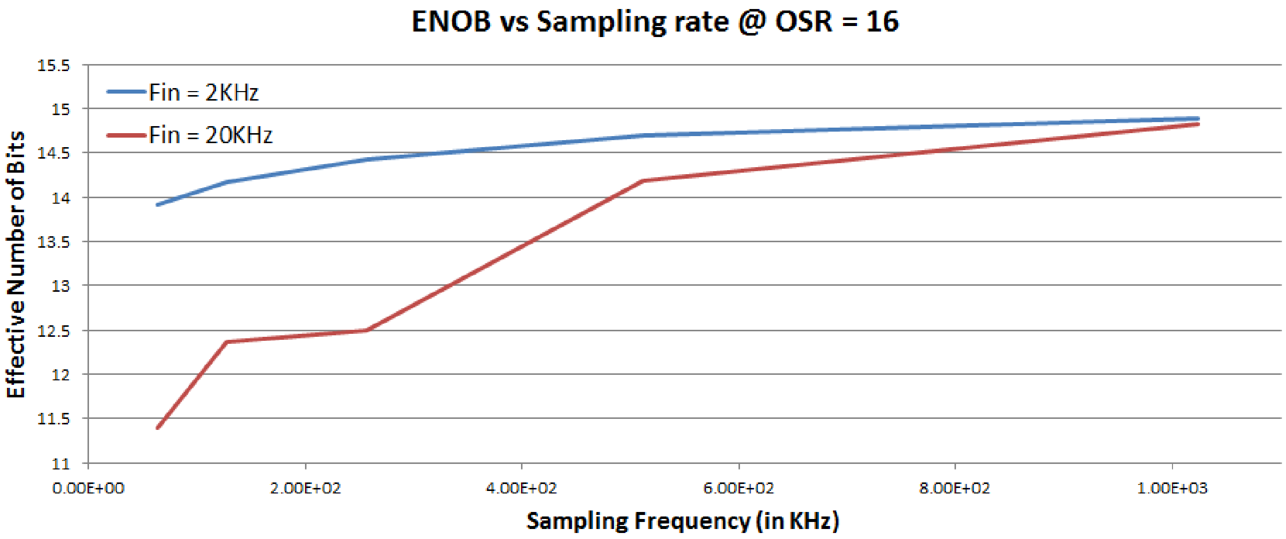


Figure 10. ADC14 Performance Using Oversampling at Different Sampling Frequencies

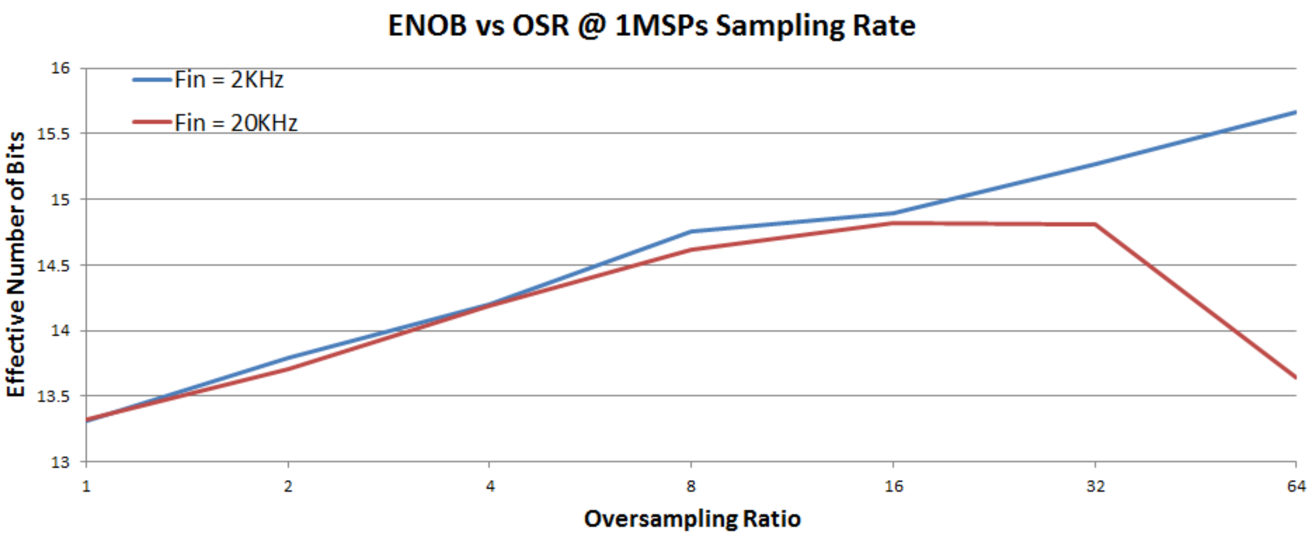


Figure 11. ADC14 Performance with Different Oversampling Ratios

6 Conclusion

This application report, which adds to *Oversampling the ADC12 for Higher Resolution* (SLAA323), provides base knowledge of ADC operation and shows how to achieve higher resolution from ADC modules through use of the oversampling method. This technique is popularly used in analog applications to digitally implement the filtering and decimation process to lower the constraints of the analog anti-aliasing filter and therefore accomplish extra bits of resolution. A demonstration is included and describes the necessary hardware connections between the DAC8560EVM and either MSP-EXP430FR4133 and MSP-EXP432P401R LaunchPad development kits. Firmware variations in the form of CCSv6 projects are provided in an associated zip file for both MSP variations with detailed instructions concerning how the code can be changed to adjust the extra bits of resolution or sampling frequency. Which parts of the example code are solely used for the oversampling method has been described, so that they can be easily extracted from the demo firmware and adapted for the end application. The results presented furthermore prove that the oversampling technique is an efficient means of increasing the resolution of MSP ADCs. It should be mentioned that results may vary between devices due to the quantization noise of each ADC.

7 References

1. *Oversampling the ADC12 for Higher Resolution* (SLAA323)
2. *MSP430FR4xx and MSP430FR2xx Family User's Guide* (SLAU445)
3. *MSP432P4xx Family Technical Reference Manual* (SLAU356)
4. DAC8560EVM Evaluation Module (DAC8560EVM)
5. MSP430FR4133 LaunchPad Development Kit (MSP-EXP430FR4133)
6. MSP432P401R LaunchPad Development Kit (MSP-EXP432P401R)

Revision History

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

Changes from February 3, 2016 to April 1, 2016 **Page**

- Changed the sentence that starts "In this equation, SNRDB increases by..." in Section 2.1, *ADC Operation* 2
- Added from first paragraph through Equation 5 in Section 2.2, *Improving ADC Resolution* 2
- Changed the first paragraph of Section 4.1, *Setting Constants*.. 6
- Added Section 5.2, *Silicon Results of Oversampling and Decimation on MSP432P401R* 10

IMPORTANT NOTICE FOR TI DESIGN INFORMATION AND RESOURCES

Texas Instruments Incorporated (‘TI”) technical, application or other design advice, services or information, including, but not limited to, reference designs and materials relating to evaluation modules, (collectively, “TI Resources”) are intended to assist designers who are developing applications that incorporate TI products; by downloading, accessing or using any particular TI Resource in any way, you (individually or, if you are acting on behalf of a company, your company) agree to use it solely for this purpose and subject to the terms of this Notice.

TI’s provision of TI Resources does not expand or otherwise alter TI’s applicable published warranties or warranty disclaimers for TI products, and no additional obligations or liabilities arise from TI providing such TI Resources. TI reserves the right to make corrections, enhancements, improvements and other changes to its TI Resources.

You understand and agree that you remain responsible for using your independent analysis, evaluation and judgment in designing your applications and that you have full and exclusive responsibility to assure the safety of your applications and compliance of your applications (and of all TI products used in or for your applications) with all applicable regulations, laws and other applicable requirements. You represent that, with respect to your applications, you have all the necessary expertise to create and implement safeguards that (1) anticipate dangerous consequences of failures, (2) monitor failures and their consequences, and (3) lessen the likelihood of failures that might cause harm and take appropriate actions. You agree that prior to using or distributing any applications that include TI products, you will thoroughly test such applications and the functionality of such TI products as used in such applications. TI has not conducted any testing other than that specifically described in the published documentation for a particular TI Resource.

You are authorized to use, copy and modify any individual TI Resource only in connection with the development of applications that include the TI product(s) identified in such TI Resource. NO OTHER LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE TO ANY OTHER TI INTELLECTUAL PROPERTY RIGHT, AND NO LICENSE TO ANY TECHNOLOGY OR INTELLECTUAL PROPERTY RIGHT OF TI OR ANY THIRD PARTY IS GRANTED HEREIN, including but not limited to any patent right, copyright, mask work right, or other intellectual property right relating to any combination, machine, or process in which TI products or services are used. Information regarding or referencing third-party products or services does not constitute a license to use such products or services, or a warranty or endorsement thereof. Use of TI Resources may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

TI RESOURCES ARE PROVIDED “AS IS” AND WITH ALL FAULTS. TI DISCLAIMS ALL OTHER WARRANTIES OR REPRESENTATIONS, EXPRESS OR IMPLIED, REGARDING TI RESOURCES OR USE THEREOF, INCLUDING BUT NOT LIMITED TO ACCURACY OR COMPLETENESS, TITLE, ANY EPIDEMIC FAILURE WARRANTY AND ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NON-INFRINGEMENT OF ANY THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

TI SHALL NOT BE LIABLE FOR AND SHALL NOT DEFEND OR INDEMNIFY YOU AGAINST ANY CLAIM, INCLUDING BUT NOT LIMITED TO ANY INFRINGEMENT CLAIM THAT RELATES TO OR IS BASED ON ANY COMBINATION OF PRODUCTS EVEN IF DESCRIBED IN TI RESOURCES OR OTHERWISE. IN NO EVENT SHALL TI BE LIABLE FOR ANY ACTUAL, DIRECT, SPECIAL, COLLATERAL, INDIRECT, PUNITIVE, INCIDENTAL, CONSEQUENTIAL OR EXEMPLARY DAMAGES IN CONNECTION WITH OR ARISING OUT OF TI RESOURCES OR USE THEREOF, AND REGARDLESS OF WHETHER TI HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

You agree to fully indemnify TI and its representatives against any damages, costs, losses, and/or liabilities arising out of your non-compliance with the terms and provisions of this Notice.

This Notice applies to TI Resources. Additional terms apply to the use and purchase of certain types of materials, TI products and services. These include; without limitation, TI’s standard terms for semiconductor products http://www.ti.com/sc/docs/stdterms.htm), evaluation modules, and samples (http://www.ti.com/sc/docs/sampterms.htm).

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