AVR121: Enhancing ADC resolution by oversampling

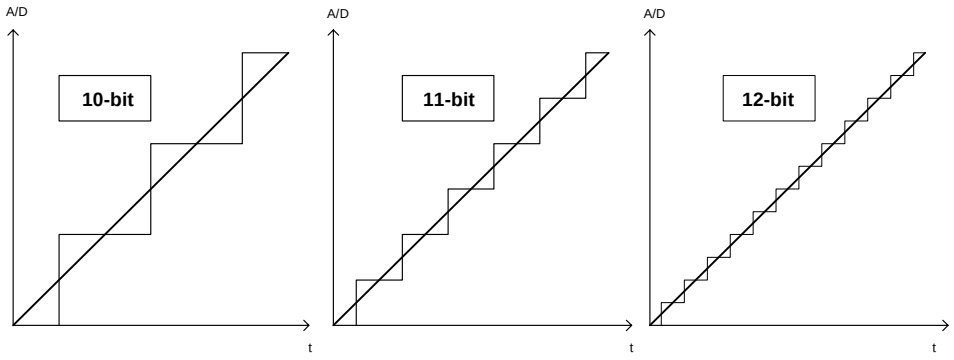
Features

- **Increasing the resolution by oversampling**
- **Averaging and decimation**
- **Noise reduction by averaging samples**

1 Introduction

Atmel's AVR controller offers an Analog to Digital Converter with 10-bit resolution. In most cases 10-bit resolution is sufficient, but in some cases higher accuracy is desired. Special signal processing techniques can be used to improve the resolution of the measurement. By using a method called 'Oversampling and Decimation' higher resolution might be achieved, without using an external ADC. This Application Note explains the method, and which conditions need to be fulfilled to make this method work properly.

Figure 1-1. Enhancing the resolution.



8-bit AVR® Microcontrollers

Application Note

Rev. 8003A-AVR-09/05

2 Theory of operation

Before reading the rest of this Application Note, the reader is encouraged to read Application Note AVR120 - 'Calibration of the ADC', and the ADC section in the AVR datasheet. The following examples and numbers are calculated for Single Ended Input in a Free Running Mode. ADC Noise Reduction Mode is not used. This method is also valid in the other modes, though the numbers in the following examples will be different.

The ADCs reference voltage and the ADCs resolution define the ADC step size. The ADC's reference voltage, V_{REF}, may be selected to AVCC, an internal 2.56V / 1.1V reference, or a reference voltage at the AREF pin. A lower V_{REF} provides a higher voltage precision but minimizes the dynamic range of the input signal. If the 2.56V V_{REF} is selected, this will give the user ~2.5mV accuracy on the conversion result, and the highest input voltage that is measured is 2.56V. Alternatively one could consider using the ADC input channels with gain stage. This will give the user the possibility of measuring an analog signal with better voltage precision, at the expense of the ADCs dynamic range. If it is not acceptable to trade dynamic range for better voltage resolution, one could choose to trade oversampling of the signal for improved resolution. This method is however limited by the characteristic of the ADC: Using oversampling and decimation will only lower the ADCs quantization error, it does not compensate for the ADCs integral non-linearity.

2.1 Sampling frequency

The Nyquist theorem states that a signal must be sampled at least twice as fast as the bandwidth of the signal to accurately reconstruct the waveform; otherwise, the high-frequency content will *alias* at a frequency inside the spectrum of interest (passband). The minimum required sampling frequency, in accordance to the Nyquist Theorem, is the Nyquist Frequency.

Equation 2-1. The Nyquist Frequency

$$f_{nyquist} > 2 \cdot f_{signal}$$

Where f_{signal} is the highest frequency of interest in the input signal. Sampling frequencies above $f_{nyquist}$ are called 'oversampling'. This sampling frequency, however, is just a theoretical absolute minimum sampling frequency. In practice the user usually wishes the highest possible sampling frequency, to give the best possible representation of the measured signal, in time domain. One could say that in most cases the input signal is already oversampled.

The sampling frequency is a result of prescaling the CPU clock; a lower prescaling factor gives a higher ADC clock frequency. At a certain point, a higher ADC clock will decrease the accuracy of the conversion as the Effective Number Of Bits, ENOB, will decrease. All ADCs has bandwidth limitations, AVRs ADC is no exception. According to the datasheet, to get a 10 bits resolution on the conversion result, the ADC clock frequency should be 50kHz – 200kHz. When the ADC clock is 200kHz, the sampling frequency is ~15kSPS, which confines the upper frequency in the sampled signal to ~7.5kHz. According to the datasheet, the ADC clock can be driven on frequencies up to 1Mhz, though this will lower the ENOB.

3 Theory

3.1 Oversampling and decimation

The theory behind ‘Oversampling and decimation’ is rather complex, but using the method is fairly easy. The technique requires a higher amount of samples. These extra samples can be achieved by oversampling the signal. For each additional bit of resolution, n, the signal must be oversampled four times. Which frequency to sample the input signal with, is given by Equation 3-1. To get the best possible representation of a analog input signal, it is necessary to oversample the signal this much, because a larger amount of samples will give a better representation of the input signal, when averaged. This is to be considered as the main ingredient of this Application Note, and will be further explained by the following theory and examples.

Equation 3-1. Oversampling frequency

$$f_{oversampling} = 4^n \cdot f_{nyquist}$$

3.2 Noise

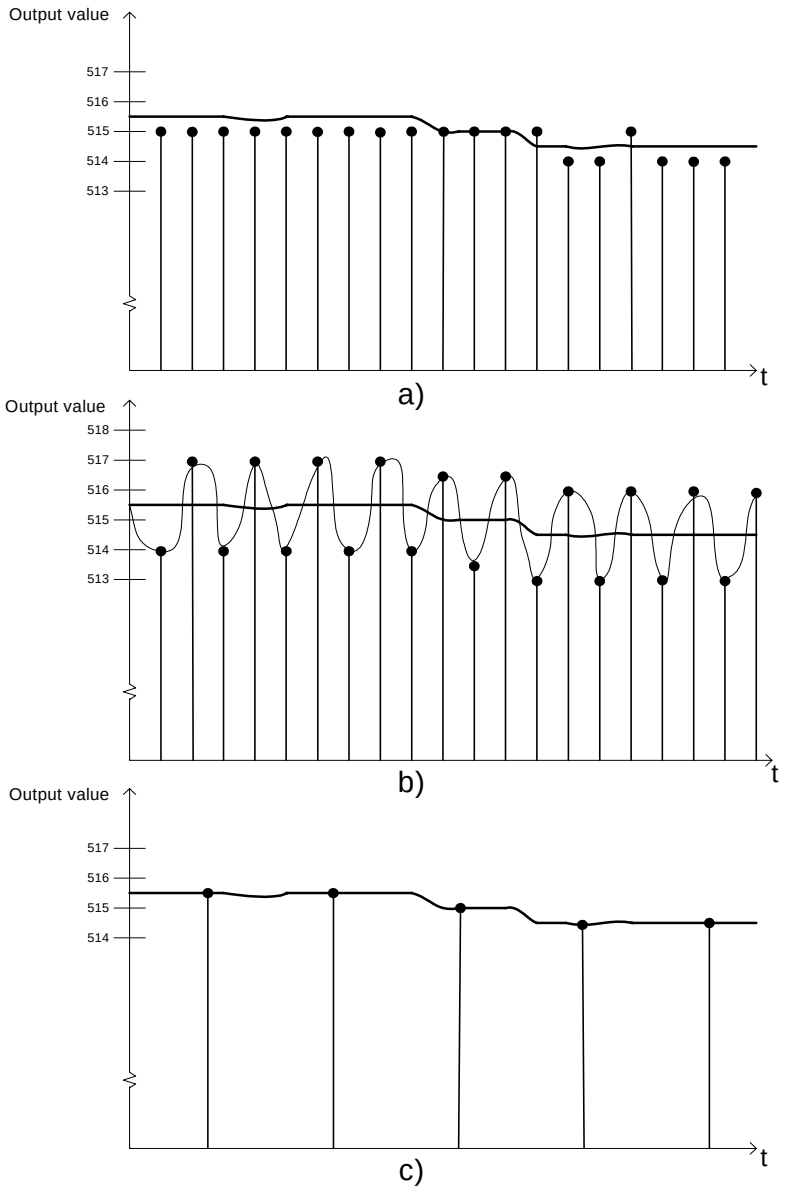
To make this method work properly, the signal-component of interest should not vary during a conversion. However another criteria for a successful enhancement of the resolution is that the input signal has to vary when sampled. This may look like a contradiction, but in this case variation means just a few LSB. The variation should be seen as the noise-component of the signal. When oversampling a signal, there should be noise present to satisfy this demand of small variations in the signal. The quantization error of the ADC is at least 0.5LSB. Therefore, the noise amplitude has to exceed 0.5 LSB to toggle the LSB. Noise amplitude of 1-2 LSB is even better because this will ensure that several samples do not end up getting the same value.

Criterias for noise, when using the decimation technique:

- The signal-component of interest should not vary significantly during a conversion.
- There should be some noise present in the signal.
- The amplitude of the noise should be at least 1 LSB.

Normally there will be some noise present during a conversion. The noise can be thermal noise, noise from the CPU core, switching of I/O-ports, variations in the power supply and others. This noise will in most cases be enough to make this method work. In specific cases though, it might be necessary to add some artificial noise to the input signal. This method is refereed to as Dithering. Figure 3-1 (a) shows the problem of measuring a signal with a voltage value that is between two quantization steps. Averaging four samples would not help, since the same low value would be the result. It may only help to attenuate signal fluctuation. Figure 3-1 (b) shows that by adding some artificial noise to the input signal, the LSB of the conversion result will toggle. Adding four of these samples halves the quantization steps, producing results that gives better representations of the input value, as shown in Figure 3-1 (c). The ADCs ‘virtual resolution’ has increased from 10 to 11-bit. This method is refereed to as Decimation and will be explained further in section 3-3.

Figure 3-1. Increasing the resolution from 10-bit to 11-bit.

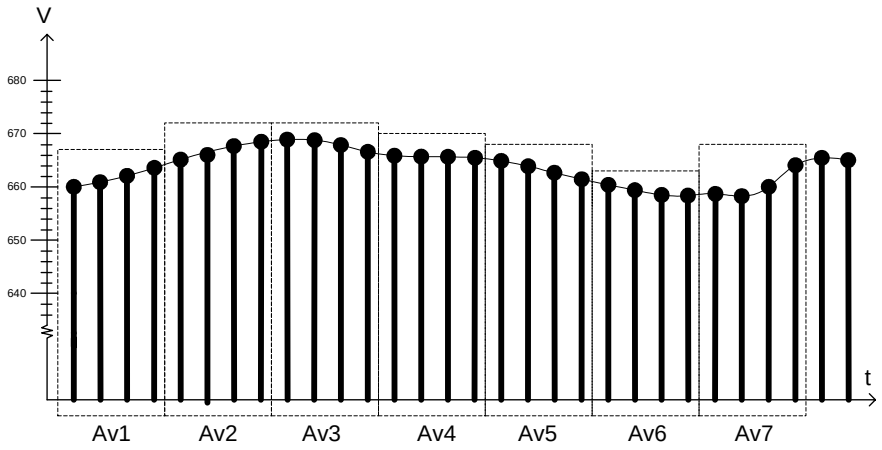


Another reason to use this method is to increase the Signal to Noise Ratio. Enhancing the Effective Number Of Bits, ENOB, will spread the noise over a greater binary number. The noises influence on each binary digit will decrease. Doubling the sampling frequency will lower the in-band noise by 3dB, and increase the resolution of the measurement by 0.5 bits.

3.3 Averaging

The conventional meaning of averaging is adding *m* samples, and dividing the result by *m*. Refereed to as *normal* averaging. Averaging data from an ADC measurement is equivalent to a low-pass filter and has the advantage of attenuating signal fluctuation or noise, and flatten out peaks in the input signal. The Moving Average method is very often used to do this. It means taking *m* readings, place them in a cyclic queue and average the most recent *m*. This will give a slight time delay, because each sample is a representation of the last *m* samples. This can be done with or without overlapping windows. Figure 3.2 shows seven (Av1-Av7), independently Moving Average result without overlapping.

Figure 3-2. Moving Average principle



It is important to remember that normal averaging does not increase the resolution of the conversion. Decimation, or Interpolation, is the averaging method, which combined with oversampling, which increases the resolution. Digital signal processing that oversamples and lowpass-filters a signal is often referred to as interpolation. In this sense, interpolation is used to produce new samples as a result of 'averaging' a larger amount of samples. The higher the number of samples averaged is, the more selective the low-pass filter will be, and the better the interpolation. The extra samples, *m*, achieved by oversampling the signal are added, just as in normal averaging, but the result are not divided by *m* as in normal averaging. Instead the result is right shifted by *n*, where *n* is the desired extra bit of resolution, to scale the answer correctly. Right shifting a binary number once is equal to dividing the binary number by a factor of 2. As seen from Equation 3-1, increasing the resolution from 10-bits to 12-bits requires the summation of 16 10-bit values. A sum of 16 10-bit values generates a 14-bit result where the last two bits are not expected to hold valuable information. To get 'back' to 12-bit it is necessary to scale the result. The scale factor, *sf*, given by Equation 3-2, is the factor, which the sum of 4^n samples should be divided by, to scale the result properly. *n* is the desired number of extra bit.

Equation 3-2.

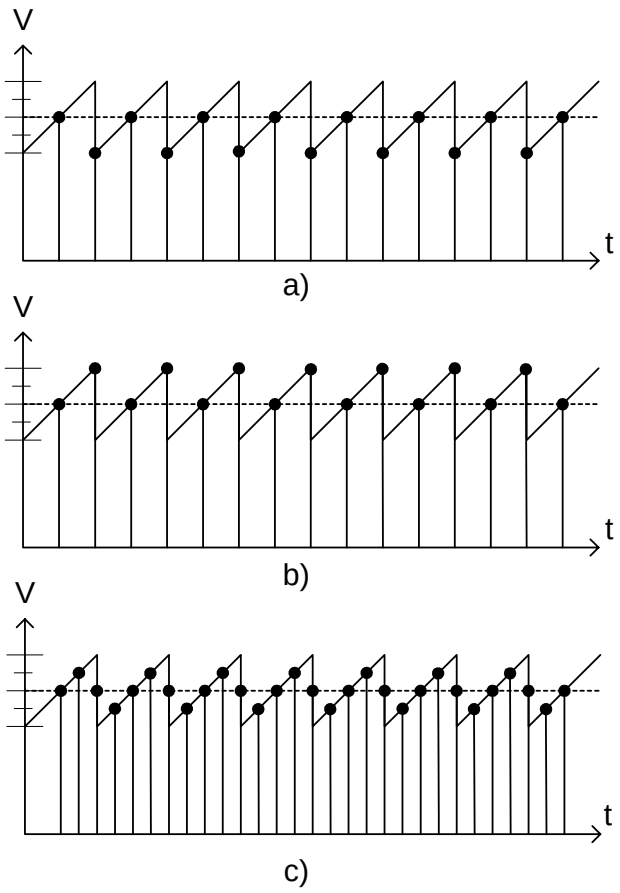
$$sf = 2^n$$

3.4 When will ‘Oversampling and Decimation’ work?

Normally a signal contains some noise, this noise very often has the characteristic of Gaussian noise, more commonly known as White noise or Thermal noise, recognized by the wide frequency spectrum and that the total energy is equally divided over the entire frequency range. In these cases the method of ‘Oversampling and decimation’ will work, if the amplitude of the noise is sufficient to toggle the LSB of the ADC conversion.

In other cases it might be necessary to add artificial noise signal to the input signal, this method is referred to as Dithering. The waveform of this noise should be Gaussian noise, but a periodical waveform will also work. What frequency this noise signal should have depends on the sampling frequency. A rule of thumb is: ”When adding *m* samples, the noise signals period should not exceed the period of *m* samples”. The amplitude of the noise should be at least 1 LSB. When adding artificial noise to a signal, it is important to remember that noise has mean value of zero; insufficient oversampling therefore may cause an offset, as shown in Figure 3-3.

Figure 3-3. Offset caused by insufficient sampling.



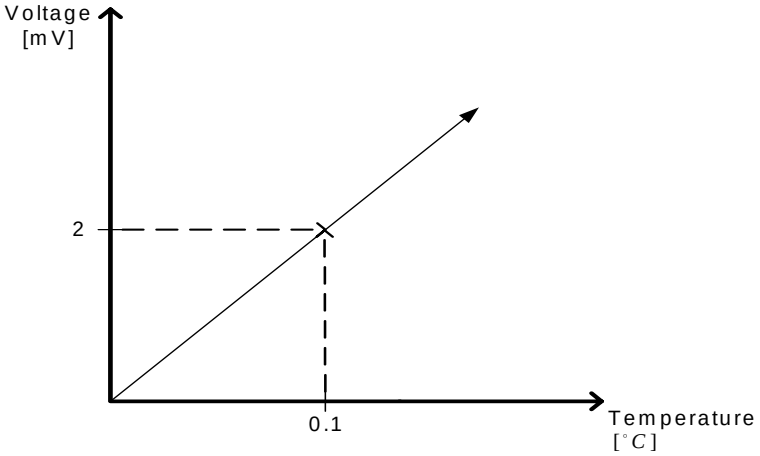
The stippled line illustrates the averaged value of the sawtooth signal. Figure 3-3 (a) will cause a negative offset. Figure 3-3 (b) will cause a positive offset. In Figure 3-3

(c) the sampling is sufficient, and offset is avoided. To create an artificial noise signal, one of the AVRs counters can be used. Since the counter and the ADC use the same clock source, this gives the possibility of synchronizing the noise and the sampling frequencies to avoid offset.

3.5 Example 1

A Brew Master in Dublin wants to measure the temperature of a process in his brewery. A slow varying signal represents the temperature measurement, and the nominal voltage in its environmental temperature is 2.5 V. Figure 3-4 shows the characteristic of the temperature-measuring device.

Figure 3-4. Voltage / Temperature function



The Brew Master doesn't want to minimize the dynamic range of the input signal and chooses a 5V reference voltage for the ADC. In this case a 10-bit ADC cannot provide a conversion result accurate enough, since the result's LSB represents a ~5mV 'step'. This is unacceptable since this will give a result that may be up to 0.25°C off. The Brew Master desires the result to have 0.1°C accuracy, which demands a voltage resolution below 2mV. If the measurement was represented by a 12 bits ADC, the voltage 'step' representing LSB would decrease to ~1.22mV. What the Brew Master needs to do is to transform the 10-bit ADC to a virtual 12-bit ADC. The input signal is varying very slowly; a very high sampling frequency is therefore not required. According to the datasheet, the ADC clock frequency should be between 50kHz and 200kHz to ensure 10-bit effective resolution. The Brew Master therefore chooses a 50kHz ADC clock frequency. Then the sampling frequency becomes ~3800 SPS. At one point the DC value that represent the temperature measurement is 2.4729V. Table 3-1 shows the different resolution options measuring this value when Vin = 2.4729V and VREF = 5V.

Table 3-1. Resolution options.

Resolution	Voltage resolution	Oversampled	Right shifted	Ideal decimated result	Ideal voltage representation	Maximum Bandwidth
10 bit	~5 mV	NA	NA	NA	2.4658V	~7600Hz (1)
11 bit	~2.5 mV	4X	1X	1012	2.4707V	~1900Hz (1)
12 bit	~1.22mV	16X	2X	2025	2.4719V	~475Hz (1)
13 bit	~610 uV	64X	3X	4051	2.4725V	~118Hz (1)
14 bit	~300 uV	256X	4X	8103	2.4728V	~29Hz (1)
15 bit	~150 uV	1024X	5X	16206	2.4728V	~7Hz (1)
16 bit	~75 uV	4096X	6X	32413	2.4729V	~3Hz (1)

Notes: 1. ADC Clock = 200kHz

The result of a single conversion is 505, which at first glance may seem correct. But this binary number also corresponds to for instance 2.4683V. This makes the user uncertain and causes errors in the temperature measurement. In certain cases this might be critical. As concluded before; a signal normally includes enough noise to make the decimation method feasible.

To increase the resolution by one bit, four samples from the same 'neighborhood' are added. These samples have values that differ from each other by a few LSB, because of the noise. These four samples are added: 508 + 507 + 505 + 505 = 2025. According to the decimation principle the answer now need to be scaled back to 11-bit. It needs to be right shifted n times, where n is the desired extra number of bits. The result is 1012. After increasing the resolution, it suddenly is possible to achieve samples between the original quantization steps. Still, the signal is oversampled enough to increase the resolution further, to 12 bit. Adding 16 10-bit samples and right-shifting the result 2 times will do this. The result is 2025. This number is more reliable, since the error margin is reduced to ~1.22mV using a 12 bit result. This example shows that the user who started off with a slow varying signal, sampled 3800 times per second, with a voltage accuracy of ~5mV, now has 240 samples per second with a 12-bit resolution, and a voltage accuracy of ~1.22mV.

The user might still want to even out signal fluctuations by averaging 16 12-bit samples, the conventional way. This is done by adding 16 samples and dividing the result by 16. At the end the user has 15 SPS consisting of 16 averaged 12-bit adjacent samples, (15 • 16 • 16 = 3840).

- Normal averaging will minimize the consequences of random noise,
- 'Oversampling and decimation' will utilize the noise to enhance the resolution

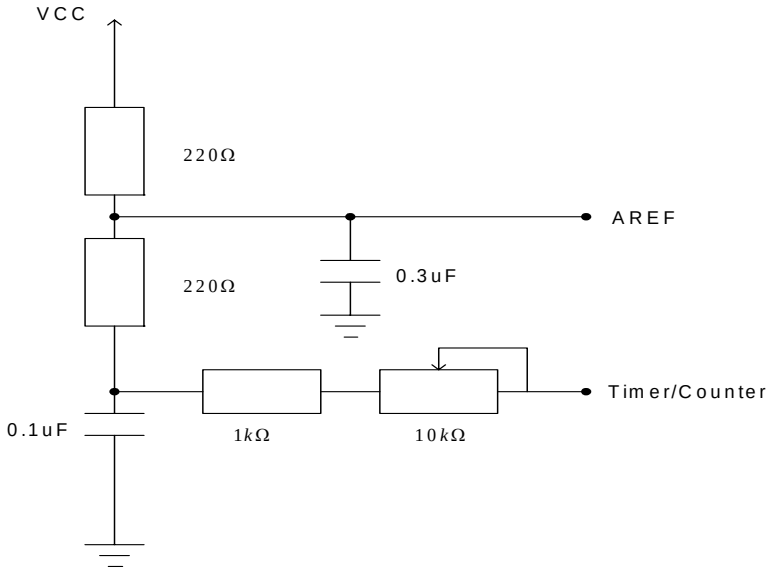
3.6 Example 2

To show the efficiency of this method, the following example will show that it is not necessary to use an external ADC to get higher accuracy. A signal generator is used to produce a linear ramp signal from 0V to 5V. In a 'low noise' environment, with a signal generator and an AVR controller plugged into an STK500 board, there may not be enough noise to toggle the last few bits of the 10-bit signal. It is therefore necessary to add artificial 'noise' to the input signal, to make the LSB toggle. Four methods were used successfully:

- Adding noise, generated by a signal generator, directly to the input signal.
- Generating noise with the AVR, using PWM, and adding it to the input signal.
- Adding noise, generated by the AVR, to AREF when using AVCC as VREF.
- Adding noise, generated by the AVR, to AREF when using AREF as VREF.

The easiest way to dither a signal is to add white noise directly to the signal, but in most cases the user does not have, or does not want to have, this kind of noise signal in the measuring environments. A more available method is to set up one of the counters in the AVR to produce a PWM signal and then low-pass filter this 'noise' to appear as a DC with a ripple peak-to-peak value of a few LSB. An example of such a filter's details and component values are shown in Figure 3-5.

Figure 3-5. LP-filter



If VCC = 5V, the filtered signal at the AREF pin appears as 2.5V when the counter's duty-cycle is 0%, and as 5V when the counter's duty-cycle is 100%. In this example the duty cycle of the PWM-signal is 50%, and the base frequency is ~3900Hz. The 10kΩ potentiometer is used to adjust this ripple. The PWM- signal is used either as the reference voltage to the ADC at AREF, or as a noise generator connected to the AREF pin. With AVCC set as ADC reference voltage. The idea is that small variations in the reference voltage will give the same effect as small variations in the input signal, without disturbing the input signal.

Measuring a linear ramp signal as shown in Figure 3-6, gives the four graphs as shown in Figure 3-7, Figure 3-8, Figure 3-9 and Figure 3-10. Figure 3-7 shows a 10-bit discrete representation of the input ramp signal, measured without artificial noise

added. The *quantization steps* are very marked. To increase the resolution, the quantization steps need to be reduced.

Figure 3-8 shows a 12-bit discrete representation of the input signal when AREF is the ADC reference voltage, and AREF is added a few LSB noise. According to Equation 3-1, each 12-bit result consists of 16 10-bit samples. The offset was adjusted for the ADC, and in accordance with Application Note AVR120, the gain-error also needed adjustment. Figure 3-9 shows a 14-bit discrete representation of the input signal and Figure 3-10 shows the 16-bit discrete representation of the input signal. When measuring a signal containing noise, or when the reference voltage is varying like in this example, it is important to remember that the top and bottom values are decreased by the same value as the amplitude of the noise signal, giving a slight reduction in the dynamic range of the measured signal. In this certain case, as a safety margin, the offset was adjusted for 100mV.

Figure 3-6. Ramp Signal, 0-5V, 100% synchronous.

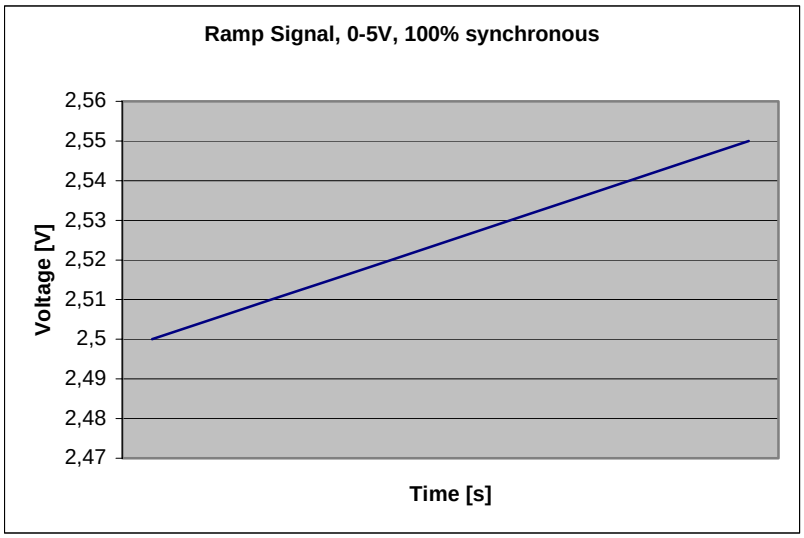


Figure 3-7. Ramp Signal reproduced with 10-bit resolution

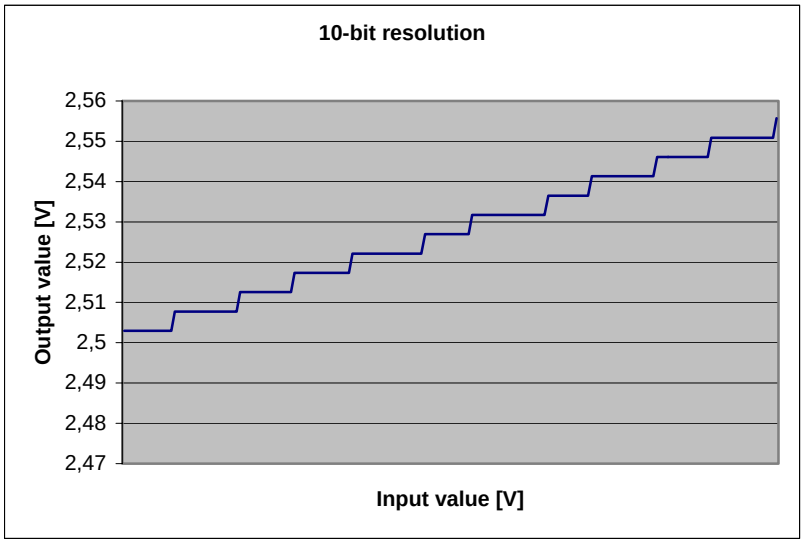


Figure 3-8. Ramp Signal reproduced with 12-bit resolution

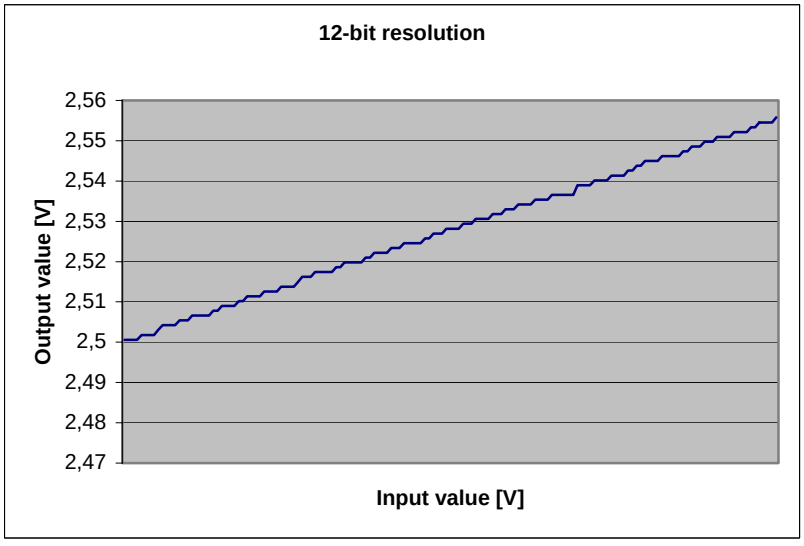


Figure 3-9. Ramp Signal reproduced with 14-bit resolution

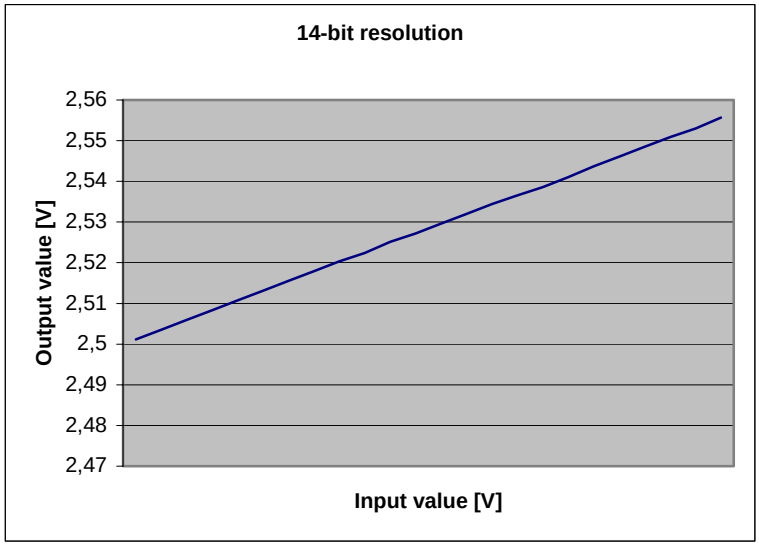
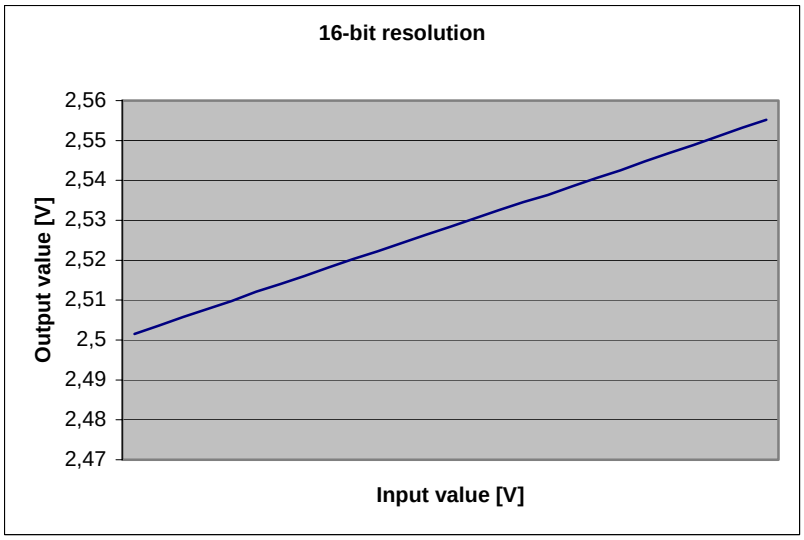


Figure 3-10. Ramp Signal reproduced with 16-bit resolution



One can easily see that by using the oversampling and decimation method, it is possible to increase resolution significantly.

Summary

When the ADC samples a signal, it quantizes the signal in discrete steps. This introduces some error, often referred to as quantization error. Normal averaging will only even out signal fluctuations, while Decimation will increase the resolution. In a 4-times-oversampled signal, four adjacent data points are averaged to produce a new data point. Which frequency to oversample the signal with, can be calculated by equation 3-1. Adding these extra samples and right-shifting the result by a factor *n*, yields a result with resolution increased by n bits. Averaging four ADC results to get a new ADC result is the same as if the ADC sampled at ¼ of the rate, but also has the effect of averaging the quantization noise, which improves SNR. This will increase the ENOB and reduce the quantization error. With the availability of faster ADCs and with low memory cost, the advantages of oversampling are cost effective and desirable.

- Some noise has to be present in the signal, at least 1 LSB.
- If the noise amplitude is not sufficient, add noise to the signal.
- Accumulate 4^n 10-bit samples, where n is the desired extra number of bits in the resolution.
- Scale the accumulated result, by right shifting it *n* times.
- Compensate for errors, according to Application Note AVR120.

Atmel Corporation

2325 Orchard Parkway
San Jose, CA 95131, USA
Tel: 1(408) 441-0311
Fax: 1(408) 487-2600

Regional Headquarters

Europe
Atmel Sarl
Route des Arsenaux 41
Case Postale 80
CH-1705 Fribourg
Switzerland
Tel: (41) 26-426-5555
Fax: (41) 26-426-5500

Asia
Room 1219
Chinachem Golden Plaza
77 Mody Road Tsimshatsui
East Kowloon
Hong Kong
Tel: (852) 2721-9778
Fax: (852) 2722-1369

Japan
9F, Tonetsu Shinkawa Bldg.
1-24-8 Shinkawa
Chuo-ku, Tokyo 104-0033
Japan
Tel: (81) 3-3523-3551
Fax: (81) 3-3523-7581

Atmel Operations

Memory
2325 Orchard Parkway
San Jose, CA 95131, USA
Tel: 1(408) 441-0311
Fax: 1(408) 436-4314

Microcontrollers
2325 Orchard Parkway
San Jose, CA 95131, USA
Tel: 1(408) 441-0311
Fax: 1(408) 436-4314

La Chantrerie
BP 70602
44306 Nantes Cedex 3, France
Tel: (33) 2-40-18-18-18
Fax: (33) 2-40-18-19-60

ASIC/ASSP/Smart Cards
Zone Industrielle
13106 Rousset Cedex, France
Tel: (33) 4-42-53-60-00
Fax: (33) 4-42-53-60-01

1150 East Cheyenne Mtn. Blvd.
Colorado Springs, CO 80906, USA
Tel: 1(719) 576-3300
Fax: 1(719) 540-1759

Scottish Enterprise Technology Park
Maxwell Building
East Kilbride G75 0QR, Scotland
Tel: (44) 1355-803-000
Fax: (44) 1355-242-743

RF/Automotive
Theresienstrasse 2
Postfach 3535
74025 Heilbronn, Germany
Tel: (49) 71-31-67-0
Fax: (49) 71-31-67-2340

1150 East Cheyenne Mtn. Blvd.
Colorado Springs, CO 80906, USA
Tel: 1(719) 576-3300
Fax: 1(719) 540-1759

Biometrics/Imaging/Hi-Rel MPU/
High Speed Converters/RF Datacom
Avenue de Rochepleine
BP 123
38521 Saint-Egreve Cedex, France
Tel: (33) 4-76-58-30-00
Fax: (33) 4-76-58-34-80

Literature Requests
www.atmel.com/literature

8003A-AVR-09/05