
ADC Oversampling with tinyAVR® 0- and 1-series, and megaAVR® 0-series

Introduction

Author: Rupali Honrao, Microchip Inc.

The Microchip tinyAVR® 0- and 1-series, and megaAVR® 0-series 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. For example, by using 10-bit ADC, a 12-bit result could be achieved with oversampling technique. This application note explains the method and conditions needed to be fulfilled to make this method work properly. This application note also provides source code according to the explained theory to achieve this oversampling technique.

Features

- Increasing the ADC resolution for the Microchip tinyAVR® 0- and 1-series, and megaAVR® 0-series devices by oversampling
- Averaging and decimation
- Software has been implemented as an Atmel START example project for the ATtiny817 to achieve 12-bit resolution from 10-bit resolution
- Shows configuration option in the source code to select:
 - ADC input pin
 - ADC sampling rate
- ADC results are sent through USART to the serial terminal:
 - Measured analog input voltage (in volts) is displayed
 - For comparison, both oversampled and normal results are displayed.

Table of Contents

Introduction.....	1
Features.....	1
1. Relevant Devices.....	3
1.1. tinyAVR 0-series.....	3
1.2. tinyAVR 1-series.....	3
1.3. megaAVR [®] 0-series.....	4
2. Theory of Operation.....	5
2.1. Sampling Frequency.....	5
2.2. Oversampling and Decimation.....	5
2.3. Noise.....	5
2.4. Averaging.....	8
2.5. When Will 'Oversampling and Decimation' Work?.....	9
3. Get Source Code from Atmel START.....	11
4. Source Code Overview.....	12
5. Macro Configurations.....	13
6. Application Flow Diagram.....	14
7. How Oversampling Demo Application Works.....	15
8. Revision History.....	16
The Microchip Web Site.....	17
Customer Change Notification Service.....	17
Customer Support.....	17
Microchip Devices Code Protection Feature.....	17
Legal Notice.....	18
Trademarks.....	18
Quality Management System Certified by DNV.....	19
Worldwide Sales and Service.....	20

1. Relevant Devices

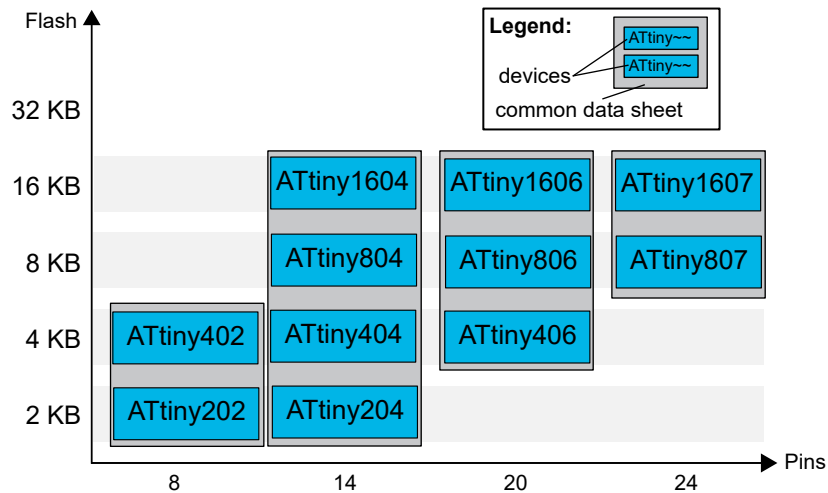
This chapter lists the relevant devices for this document.

1.1 tinyAVR 0-series

The figure below shows the tinyAVR 0-series, laying out pin count variants and memory sizes:

- Vertical migration is possible without code modification, as these devices are fully pin- and feature compatible.
- Horizontal migration to the left reduces the pin count and, therefore, the available features.

Figure 1-1. tinyAVR® 0-series Overview



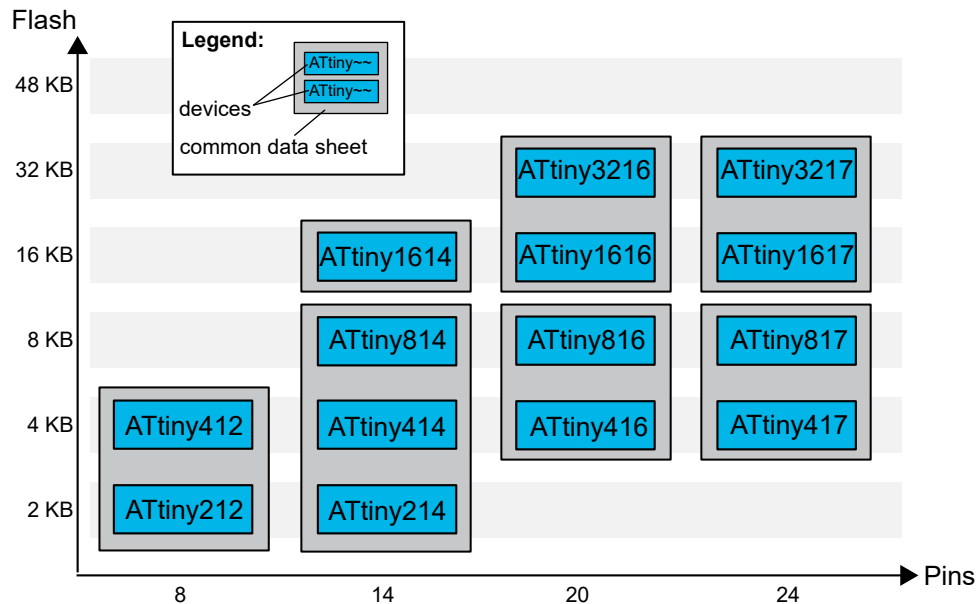
Devices with different Flash memory size typically also have different SRAM and EEPROM.

1.2 tinyAVR 1-series

The following figure shows the tinyAVR 1-series devices, laying out pin count variants and memory sizes:

- Vertical migration upwards is possible without code modification, as these devices are pin compatible and provide the same or more features. Downward migration may require code modification due to fewer available instances of some peripherals.
- Horizontal migration to the left reduces the pin count and, therefore, the available features.

Figure 1-2. tinyAVR® 1-series Overview



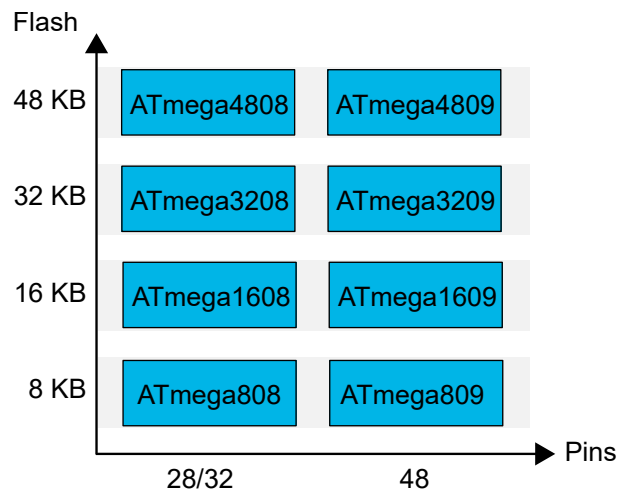
Devices with different Flash memory size typically also have different SRAM and EEPROM.

1.3 megaAVR® 0-series

The figure below shows the megaAVR 0-series devices, laying out pin count variants and memory sizes:

- Vertical migration is possible without code modification, as these devices are fully pin and feature compatible.
- Horizontal migration to the left reduces the pin count and, therefore, the available features.

Figure 1-3. megaAVR® 0-series Overview



Devices with different Flash memory size typically also have different SRAM and EEPROM.

2. Theory of Operation

This chapter explains how oversampling works with all the necessary mathematical details.

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 with the Nyquist Theorem, is the Nyquist frequency.

Equation 2-1. The Nyquist Frequency

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

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 and 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 the time domain. 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.

2.1.1 ADC Clock Limit

For the Microchip tinyAVR® 0- and 1-series, and megaAVR® 0-series devices to get a 10-bits resolution on the conversion result, the ADC clock frequency may be maximum 1.5 MHz. When the ADC clock is 1.5 MHz, the sampling frequency is 150 kps, which confines the upper frequency in the sampled signal to ~75 kHz.

2.2 Oversampling and Decimation

The oversampling 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 4^n times. The frequency the signal has to be sampled with is given by the equation below:

Equation 2-2. Oversampling Frequency

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

2.3 Noise

To make this method work properly, the signal component of interest may not vary greatly during a conversion. However, another criterion for a successful enhancement of the resolution is that the input signal has to vary slightly when sampled. This may look like a contradiction, but in this case, variation means just a few Least Significant Bytes (LSB). The variation may be seen as the noise component of the signal. When oversampling a signal, there may be noise present to satisfy this demand of small variations in the signal. The quantization error of the ADC is at least 0.5 LSB. 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.

Criteria for noise when using the decimation technique:

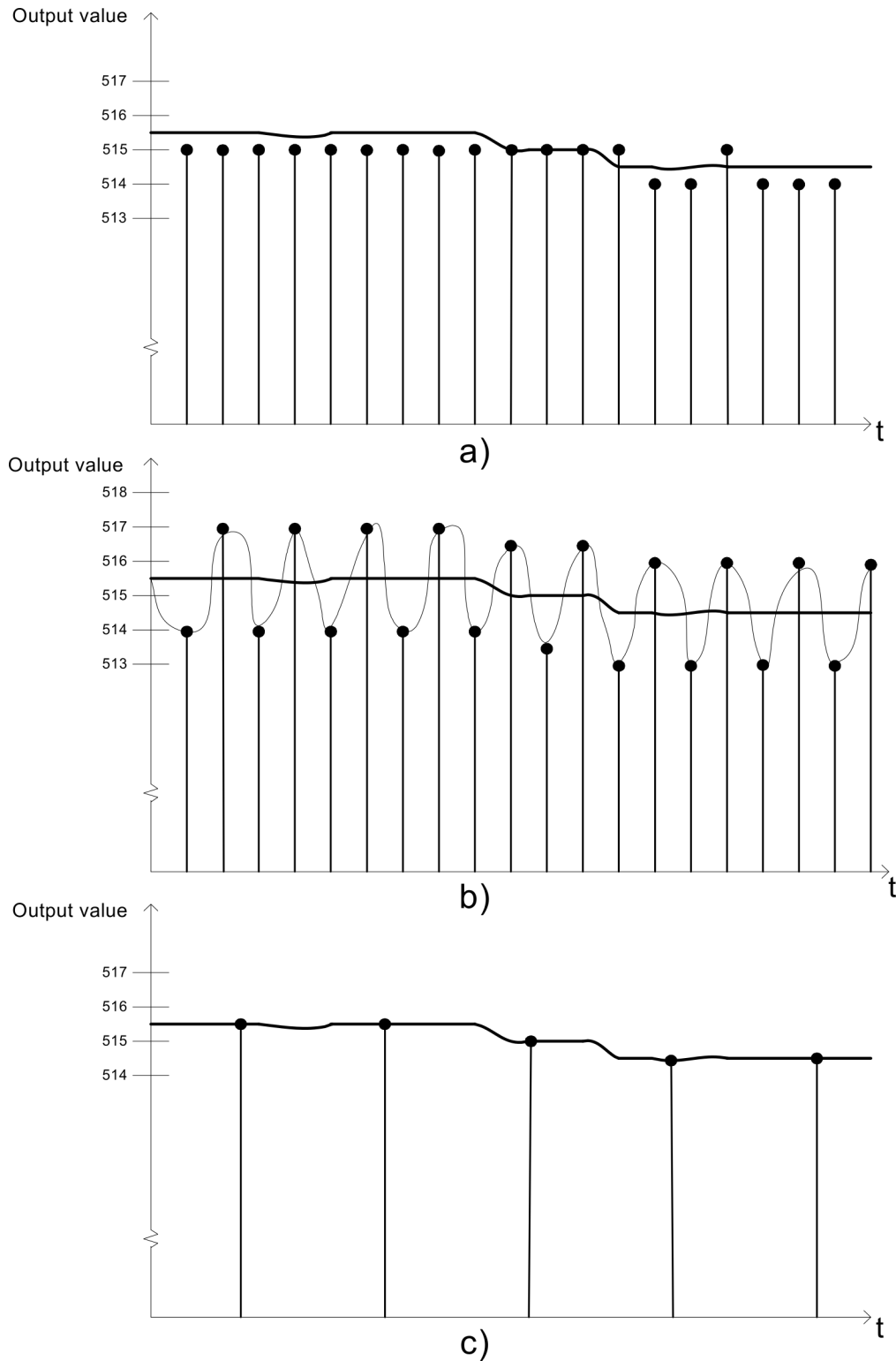
- The signal component of interest may not vary significantly during a conversion
- There may be some noise present in the signal
- The amplitude of the noise may be at least 1 LSB

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 special cases though, it might be necessary to add some artificial noise to the input signal. This method is referred to as dithering.

[Figure 2-1](#) (a) shows the problem of measuring a signal with a voltage value that is between two quantization steps. Averaging four samples will not help, since the same low value will be the result.

[Figure 2-1](#) (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 give better representations of the input value, as shown in [Figure 2-1](#) (c). The ADCs 'virtual resolution' has increased from 10 to 11 bits. This method is referred to as decimation, and will be explained further in section [2.4 Averaging](#).

Figure 2-1. Increasing the Resolution from 10-Bit to 11-Bit

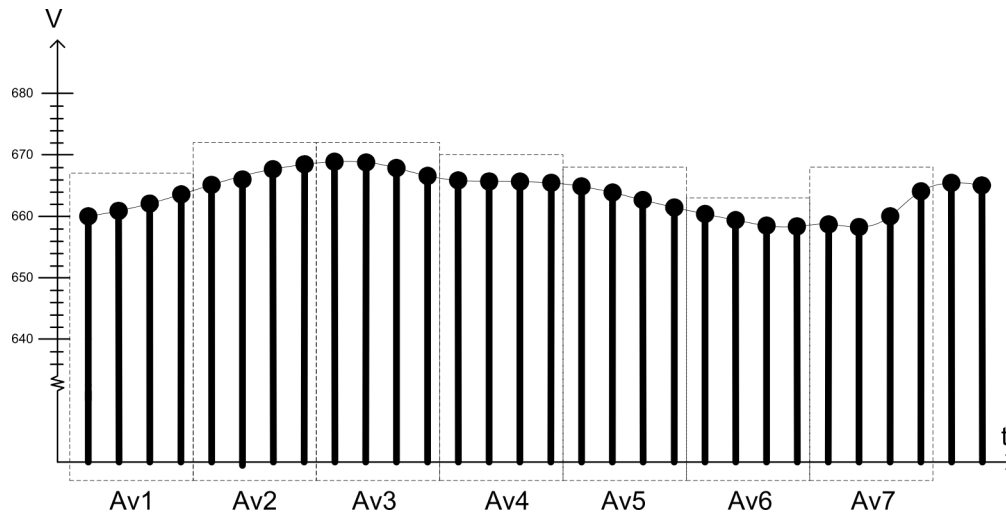


Another reason to use this method is to increase the signal-to-noise ratio. Enhancing the Effective Number of Bits, ENOB, will spread the noise over an increased binary number. The noise influence on each binary digit will decrease. Doubling the sampling frequency will lower the in-band noise by 3 dB,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

2.4 Averaging

The conventional meaning of averaging is adding m samples, and dividing the result by m , which is referred 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 it will flatten out peaks in the input signal. The moving average method is very often used to do this. It works by 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. The figure below shows seven (Av1-Av7) independently moving average results without overlapping.

Figure 2-2. 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 increases the resolution.

The extra samples, m , achieved by oversampling the signal are summed up, just as in normal averaging, but the result is 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 2-2](#), increasing the resolution from 10 to 12 bits (that is, additional 2-bit resolution), requires the summation of 4^2 (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 representation, it is necessary to scale the result. The scale factor, sf , given by the equation below, 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 2-3. Scale Factor

$$sf = 2^n$$

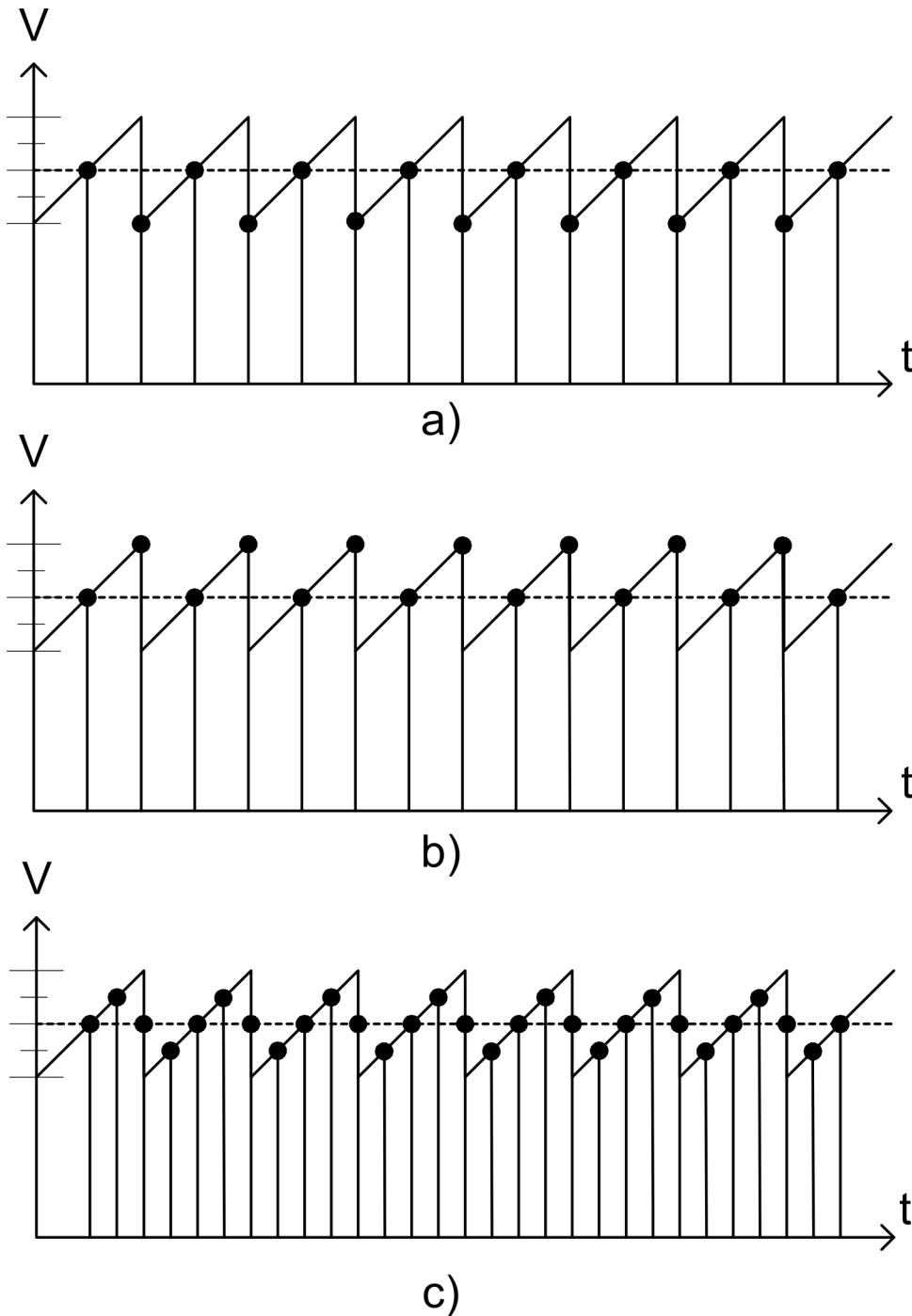
As explained in the case above (increasing resolution from 10-bit to 12-bit), the scaling factor, sf , is 2^2 , which is equal to 4.

2.5 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 by the fact 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 an artificial noise signal to the input signal. This method is referred to as dithering. The waveform of this noise may be Gaussian noise, but a periodical waveform will also work. What frequency this noise signal may have depends on the sampling frequency. A rule of thumb is: “When adding m samples, the noise signals period may not exceed the period of m samples”. The amplitude of the noise may be at least 1 LSB. When adding artificial noise to a signal, it is important to remember that noise has a mean value of zero; insufficient oversampling therefore may cause an offset, as shown in the following figure.

Figure 2-3. Offset Caused by Insufficient Sampling



The stippled line illustrates the averaged value of the saw-tooth signal. The sampling shown in figure (a) above will cause a negative offset, while the sampling in (b) will cause a positive offset. In figure (c), the sampling is sufficient, and offset is avoided. To create an artificial noise signal, one of the AVR[®] counters can be used. Since the counter and the ADC are using the same clock source, this gives the possibility of synchronizing the noise and the sampling frequencies to avoid offset.

3. Get Source Code from Atmel | START

The example code is available through Atmel | START, which is a web-based tool that enables configuration of application code through a Graphical User Interface (GUI). The code can be downloaded for both Atmel Studio and IAR Embedded Workbench® via the direct example code-link below or the *Browse examples* button on the Atmel | START front page.

Atmel | START web page: <http://start.atmel.com/>

Example Code

- ADC Oversampling with tinyAVR® 1-series:
 - http://start.atmel.com/#example/Atmel:adc_oversampling_with_tinyavr_1_series:1.0.0::Application:ADC_Oversampling_with_tinyavr_1-series:
- ADC Oversampling with megaAVR® 0-series:
 - http://start.atmel.com/#example/Atmel:adc_oversampling_with_megaavr_0_series:1.0.0::Application:ADC_Oversampling_with_megaAVR_0-series:

Click *User guide* in Atmel | START for details and information about example projects. The *User guide* button can be found in the example browser, and by clicking the project name in the dashboard view within the Atmel | START project configurator.

Atmel Studio

Download the code as an `.atzip` file for Atmel Studio from the example browser in Atmel | START, by clicking *Download selected example*. To download the file from within Atmel | START, click *Export project* followed by *Download pack*.

Double click the downloaded `.atzip` file and the project will be imported to Atmel Studio 7.0.

IAR Embedded Workbench

For information on how to import the project in IAR Embedded Workbench, open the [Atmel | START User Guide](#), select *Using Atmel Start Output in External Tools*, and *IAR Embedded Workbench*. A link to the Atmel | START User Guide can be found by clicking *Help* from the Atmel | START front page or *Help And Support* within the project configurator, both located in the upper right corner of the page.

4. Source Code Overview

The oversampling demo application has been developed and tested for the ATtiny817 Xplained Pro board.

- CPU clock (default) 3.33 MHz
- Peripherals used:
 - ADC, USART, V_{REF}
 - ADC input channel is AIN 5 pin PA5
 - ADC resolution 10 bits
 - ADC clock CLK_PER divided by 4
 - USART: baud rate 9600, TXD: PB2
 - V_{REF} : ADC reference voltage 2.5V

The project configured in Atmel START generates peripheral driver functions and files, as well as a `'main()'` function that initializes all drivers.

- Driver header and source files are in the *src* and *include* folder
- In `atmel_start.c` file, the function `'atmel_start_init()'` initializes MCU, drivers, and middleware in the project

5. Macro Configurations

Below are the macro configurations in the `main.c` file.

- Maximum input voltage

```
#define MAX_VOL 2.5
```

Maximum input voltage is configured to 2.5 to calculate the voltage for measured ADC reading.

Note: ADC reference has been configured to 2.5V.

- ADC input channel

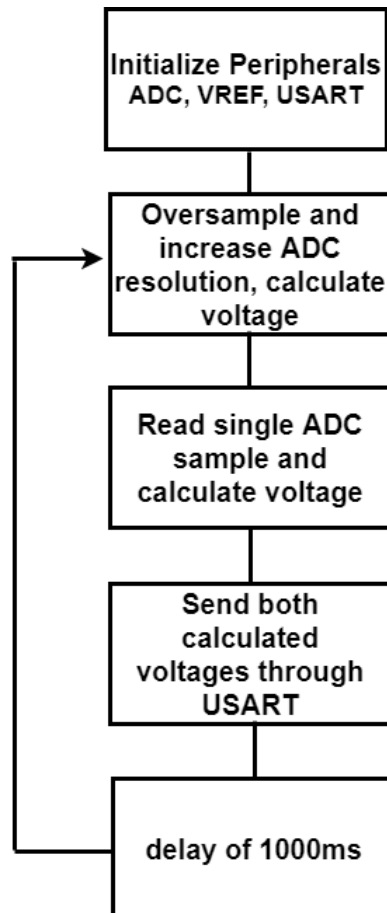
```
#define ADC_CHANNEL 5
```

The ADC input signal has been connected to channel 5: AIN5 (pin PA5).

6. Application Flow Diagram

The overall application flow is as shown in [Figure 6-1](#).

Figure 6-1. Application Flow Diagram



7. How Oversampling Demo Application Works

In the example source code provided, ADC conversion is done in the functions `'process_single_sampled()'` and `'process_oversampled()'`. For comparison, both oversampled and normal ADC results are sent through USART to the serial terminal. Measured analog input voltage (in volt) is displayed.

In the function `'process_single_sampled()'`, the ADC sample accumulator has been configured to 1. The ADC result of only one sample is read. No oversampling is done here.

The function `'process_oversampled()'` demonstrates how the oversampling is done and resolution has been increased from 10 to 12 bits. In the function `'process_oversampled()'`, to get a 12-bit resolution from the 10-bit ADC it is required to read 16 ADC samples and then right shift the sum of the ADC results by 2 (i.e. divide by 4).

To increase the resolution, for each additional bit of ADC resolution, n , the signal must be oversampled 4^n . To achieve 12-bit ADC, you need an additional 2-bit resolution. Hence, the signal has to be sampled 4^2 , i.e. 16 times more samples. The ADC has a configurable accumulator setting. This accumulator is configured to do 16 samples. The result in the ADC result register will then be the sum of the 16 samples. The scale factor, sf , is given by $sf = 2^n$. The scale factor is the number the result may be divided by to scale the result to the desired bit width. In this example, the result is increased by two bits. Hence, the scale factor is $2^2 = 4$. So the result may be divided by 4 or a right shift of 2 can be performed.

In both the `'process_single_sampled()'` and `'process_oversampled()'` functions, the ADC result is read and the measured voltage has been converted to a string using the standard library function `'dtostrf'` and this measured analog input voltage (in volt) has been sent through USART to the serial terminal of the PC every 1 second.

8. Revision History

Doc. Rev.	Date	Comments
C	10/2018	Updated figures 1-1, 1-2, 1-3 in chapter "Relevant Devices". Fixed grammar and punctuation.
B	02/2018	Added support for tinyAVR 0-series and megaAVR 0-series.
A	11/2017	Initial document release.

The Microchip Web Site

Microchip provides online support via our web site at <http://www.microchip.com/>. This web site is used as a means to make files and information easily available to customers. Accessible by using your favorite Internet browser, the web site contains the following information:

- **Product Support** – Data sheets and errata, application notes and sample programs, design resources, user's guides and hardware support documents, latest software releases and archived software
- **General Technical Support** – Frequently Asked Questions (FAQ), technical support requests, online discussion groups, Microchip consultant program member listing
- **Business of Microchip** – Product selector and ordering guides, latest Microchip press releases, listing of seminars and events, listings of Microchip sales offices, distributors and factory representatives

Customer Change Notification Service

Microchip's customer notification service helps keep customers current on Microchip products. Subscribers will receive e-mail notification whenever there are changes, updates, revisions or errata related to a specified product family or development tool of interest.

To register, access the Microchip web site at <http://www.microchip.com/>. Under "Support", click on "Customer Change Notification" and follow the registration instructions.

Customer Support

Users of Microchip products can receive assistance through several channels:

- Distributor or Representative
- Local Sales Office
- Field Application Engineer (FAE)
- Technical Support

Customers should contact their distributor, representative or Field Application Engineer (FAE) for support. Local sales offices are also available to help customers. A listing of sales offices and locations is included in the back of this document.

Technical support is available through the web site at: <http://www.microchip.com/support>

Microchip Devices Code Protection Feature

Note the following details of the code protection feature on Microchip devices:

- Microchip products meet the specification contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is one of the most secure families of its kind on the market today, when used in the intended manner and under normal conditions.
- There are dishonest and possibly illegal methods used to breach the code protection feature. All of these methods, to our knowledge, require using the Microchip products in a manner outside the operating specifications contained in Microchip's Data Sheets. Most likely, the person doing so is engaged in theft of intellectual property.
- Microchip is willing to work with the customer who is concerned about the integrity of their code.

- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of their code. Code protection does not mean that we are guaranteeing the product as “unbreakable.”

Code protection is constantly evolving. We at Microchip are committed to continuously improving the code protection features of our products. Attempts to break Microchip’s code protection feature may be a violation of the Digital Millennium Copyright Act. If such acts allow unauthorized access to your software or other copyrighted work, you may have a right to sue for relief under that Act.

Legal Notice

Information contained in this publication regarding device applications and the like is provided only for your convenience and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications. MICROCHIP MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WHETHER EXPRESS OR IMPLIED, WRITTEN OR ORAL, STATUTORY OR OTHERWISE, RELATED TO THE INFORMATION, INCLUDING BUT NOT LIMITED TO ITS CONDITION, QUALITY, PERFORMANCE, MERCHANTABILITY OR FITNESS FOR PURPOSE. Microchip disclaims all liability arising from this information and its use. Use of Microchip devices in life support and/or safety applications is entirely at the buyer’s risk, and the buyer agrees to defend, indemnify and hold harmless Microchip from any and all damages, claims, suits, or expenses resulting from such use. No licenses are conveyed, implicitly or otherwise, under any Microchip intellectual property rights unless otherwise stated.

Trademarks

The Microchip name and logo, the Microchip logo, AnyRate, AVR, AVR logo, AVR Freaks, BitCloud, chipKIT, chipKIT logo, CryptoMemory, CryptoRF, dsPIC, FlashFlex, flexPWR, Helder, JukeBlox, KeeLoq, Klear, LANCheck, LINK MD, maXStylus, maXTouch, MediaLB, megaAVR, MOST, MOST logo, MPLAB, OptoLyzer, PIC, picoPower, PICSTART, PIC32 logo, Prochip Designer, QTouch, SAM-BA, SpyNIC, SST, SST Logo, SuperFlash, tinyAVR, UNI/O, and XMEGA are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

ClockWorks, The Embedded Control Solutions Company, EtherSynch, Hyper Speed Control, HyperLight Load, IntelliMOS, mTouch, Precision Edge, and Quiet-Wire are registered trademarks of Microchip Technology Incorporated in the U.S.A.

Adjacent Key Suppression, AKS, Analog-for-the-Digital Age, Any Capacitor, AnyIn, AnyOut, BodyCom, CodeGuard, CryptoAuthentication, CryptoAutomotive, CryptoCompanion, CryptoController, dsPICDEM, dsPICDEM.net, Dynamic Average Matching, DAM, ECAN, EtherGREEN, In-Circuit Serial Programming, ICSP, INICnet, Inter-Chip Connectivity, JitterBlocker, KlearNet, KlearNet logo, memBrain, Mindi, MiWi, motorBench, MPASM, MPF, MPLAB Certified logo, MPLIB, MPLINK, MultiTRAK, NetDetach, Omniscient Code Generation, PICDEM, PICDEM.net, PICKit, PICtail, PowerSmart, PureSilicon, QMatrix, REAL ICE, Ripple Blocker, SAM-ICE, Serial Quad I/O, SMART-I.S., SQI, SuperSwitcher, SuperSwitcher II, Total Endurance, TSHARC, USBCheck, VariSense, ViewSpan, WiperLock, Wireless DNA, and ZENA are trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

SQTP is a service mark of Microchip Technology Incorporated in the U.S.A.

Silicon Storage Technology is a registered trademark of Microchip Technology Inc. in other countries.

GestIC is a registered trademark of Microchip Technology Germany II GmbH & Co. KG, a subsidiary of Microchip Technology Inc., in other countries.

All other trademarks mentioned herein are property of their respective companies.

© 2018, Microchip Technology Incorporated, Printed in the U.S.A., All Rights Reserved.

ISBN: 978-1-5224-3674-4

Quality Management System Certified by DNV

ISO/TS 16949

Microchip received ISO/TS-16949:2009 certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona; Gresham, Oregon and design centers in California and India. The Company's quality system processes and procedures are for its PIC[®] MCUs and dsPIC[®] DSCs, KEELOQ[®] code hopping devices, Serial EEPROMs, microperipherals, nonvolatile memory and analog products. In addition, Microchip's quality system for the design and manufacture of development systems is ISO 9001:2000 certified.

Worldwide Sales and Service

AMERICAS	ASIA/PACIFIC	ASIA/PACIFIC	EUROPE
Corporate Office 2355 West Chandler Blvd. Chandler, AZ 85224-6199 Tel: 480-792-7200 Fax: 480-792-7277 Technical Support: http://www.microchip.com/support Web Address: www.microchip.com	Australia - Sydney Tel: 61-2-9868-6733 China - Beijing Tel: 86-10-8569-7000 China - Chengdu Tel: 86-28-8665-5511 China - Chongqing Tel: 86-23-8980-9588 China - Dongguan Tel: 86-769-8702-9880 China - Guangzhou Tel: 86-20-8755-8029 China - Hangzhou Tel: 86-571-8792-8115 China - Hong Kong SAR Tel: 852-2943-5100 China - Nanjing Tel: 86-25-8473-2460 China - Qingdao Tel: 86-532-8502-7355 China - Shanghai Tel: 86-21-3326-8000 China - Shenyang Tel: 86-24-2334-2829 China - Shenzhen Tel: 86-755-8864-2200 China - Suzhou Tel: 86-186-6233-1526 China - Wuhan Tel: 86-27-5980-5300 China - Xian Tel: 86-29-8833-7252 China - Xiamen Tel: 86-592-2388138 China - Zhuhai Tel: 86-756-3210040	India - Bangalore Tel: 91-80-3090-4444 India - New Delhi Tel: 91-11-4160-8631 India - Pune Tel: 91-20-4121-0141 Japan - Osaka Tel: 81-6-6152-7160 Japan - Tokyo Tel: 81-3-6880-3770 Korea - Daegu Tel: 82-53-744-4301 Korea - Seoul Tel: 82-2-554-7200 Malaysia - Kuala Lumpur Tel: 60-3-7651-7906 Malaysia - Penang Tel: 60-4-227-8870 Philippines - Manila Tel: 63-2-634-9065 Singapore Tel: 65-6334-8870 Taiwan - Hsin Chu Tel: 886-3-577-8366 Taiwan - Kaohsiung Tel: 886-7-213-7830 Taiwan - Taipei Tel: 886-2-2508-8600 Thailand - Bangkok Tel: 66-2-694-1351 Vietnam - Ho Chi Minh Tel: 84-28-5448-2100	Austria - Wels Tel: 43-7242-2244-39 Fax: 43-7242-2244-393 Denmark - Copenhagen Tel: 45-4450-2828 Fax: 45-4485-2829 Finland - Espoo Tel: 358-9-4520-820 France - Paris Tel: 33-1-69-53-63-20 Fax: 33-1-69-30-90-79 Germany - Garching Tel: 49-8931-9700 Germany - Haan Tel: 49-2129-3766400 Germany - Heilbronn Tel: 49-7131-67-3636 Germany - Karlsruhe Tel: 49-721-625370 Germany - Munich Tel: 49-89-627-144-0 Fax: 49-89-627-144-44 Germany - Rosenheim Tel: 49-8031-354-560 Israel - Ra'anana Tel: 972-9-744-7705 Italy - Milan Tel: 39-0331-742611 Fax: 39-0331-466781 Italy - Padova Tel: 39-049-7625286 Netherlands - Drunen Tel: 31-416-690399 Fax: 31-416-690340 Norway - Trondheim Tel: 47-72884388 Poland - Warsaw Tel: 48-22-3325737 Romania - Bucharest Tel: 40-21-407-87-50 Spain - Madrid Tel: 34-91-708-08-90 Fax: 34-91-708-08-91 Sweden - Gothenberg Tel: 46-31-704-60-40 Sweden - Stockholm Tel: 46-8-5090-4654 UK - Wokingham Tel: 44-118-921-5800 Fax: 44-118-921-5820