AVR352: Using the Coulomb Counting ADC

8-bit AVR® Microcontrollers

Application Note

Features

- **Modes of operation**
  - **Instantaneous and Accumulated conversions**
  - **Regular Current Condition**
- **Current measurement using shunt resistor**
- **Compensation and calibration**
  - **Sampling period variations**
  - **Offset-compensation**

1 Introduction

Several of the new AVR® battery monitoring devices features a Coulomb Counter ADC (CC-ADC). This is a highly accurate Sigma-Delta ADC that is designed for measuring charge and discharge currents in smart batteries.

This application note describes how to use the CC-ADC to get maximum accuracy and lowest possible current consumption.

Rev. 8078B-AVR-02/09

2 Theory of operation

The CC-ADC is an analog to digital converter designed with smart batteries especially in mind. It is used to measure charging and discharging currents with very high accuracy, to be able to provide highly accurate information about what current is flowing into or out of the battery, and thus provide state of charge and other battery information. This information is useful when predicting for how long the application can be expected to run with the remaining battery charge, and also when charging the battery. The information can also be used to assess the health of the battery and deciding if replacing it is needed.

2.1 CC-ADC modes of operation

The CC-ADC offers two different modes of operation:

- Instantaneous/Accumulated Current Conversion mode (ICC/ACC).
- Regular Current Condition (RCC) mode.

When the CC-ADC is enabled the ICC and ACC conversions start running. The ICC conversions are produced every 3.9ms, while the ACC results are generated in intervals depending on the CADAS bits setting in the CADCSRA register. The RCC mode is a mode where the CC-ADC is disabled in periods, and therefore samples the current less frequently to reduce the power consumption. The modes of operation are described below.

Please note that some of the registers used for changing settings operate in a different clock domain than the CPU and need to be synchronized to the CC-ADC clock domain. Subsequent writes to these registers will be blocked during this synchronization. A CC-ADC update busy bit, CADUB, can be used to check if an update is in progress or not.

2.1.1 Instantaneous Current Conversion mode (ICC-mode)

In ICC-mode the CC-ADC provides a 13 bit signed result in approximately 3.9ms. This mode can be used when it is desired to get information about the instantaneous current drawn. This can be used if it is desirable to measure battery voltage and discharge current at (approximately) the same time, for example to calculate impedance. The CC-ADC generates an interrupt when a conversion is completed and the result is ready in the CADICL and CADICH registers.

Note that the first three ICC conversions after enabling the CC-ADC, switching from RCC or changing polarity, are inaccurate and should possibly be discarded.

2.1.2 Accumulated Current Conversion mode (ACC-mode)

When a smart battery powered application is actively running and thereby draws significant current from the battery, the current can be monitored with very high accuracy by using the ACC mode. In ACC-mode the conversion takes between 128 ms and 1 second. However, the accuracy of the conversion is as high as 18-bit (signed). The ACC result is the average current drawn in the sampling period.

Though the ACC mode draws more current than the RCC-mode since the CC-ADC is always active, the additional current consumption of the CC-ADC is insignificant compared to the current drawn by the application itself and thus acceptable. The ACC conversions can run in both Idle and Power Save sleep mode, which should be used to reduce the overall current consumption.

Note that the first three ACC conversions after enabling the CC-ADC, switching from RCC or changing polarity, are inaccurate and should possibly be discarded. On the other hand there might run a significant current during this time that should be measured, even if the measurement is erroneous. This depends on the application.

2.1.2.1 Selecting the Accumulated Current Conversion time

Accumulation time used for ACC conversions can be set to 128, 256, 512 or 1024 milliseconds. The conversion time is not relevant for the resolution of the result, but is mainly for convenience so that it is possible to choose how often the current should be processed (how often the ACC interrupt should occur).

Note however that the noise component of the individual ACC conversions will increase with decreasing conversion time. As the noise component is stochastic the noise will average out and will not affect the calculations of the battery charge.

2.1.3 Regular Current Condition mode (RCC-mode)

When a battery-powered application is in stand-by mode, and thus draws little current from the battery, the current consumption of the microcontroller (MCU) in the smart battery should be very low to not affect the battery lifetime. The regular current mode provides the means to achieve this.

In RCC the CC-ADC repeatedly does one conversion before being turned off for a selectable interval, to minimize the current consumption. Further, the RCC mode can be used in conjunction with the AVR sleep modes, to stay in sleep until a current above a selectable level is encountered. An interrupt is then generated if the interrupt is enabled and the device wakes up.

RCC-mode has the same 13-bit signed resolution and the conversion time of 3.9 ms as ICC-mode, but the conversions are generated in intervals between 256 ms and 2 seconds as specified by the CADSI bits.

Note that the ICC interrupt is also generated upon conversion complete if the ICC interrupt is enabled. When the current is above the Regular Current Level the Accumulated Current conversion mode should be selected.

The Regular Current Level is specified in the CADRC register. Each step in the register is equivalent to 26.9 µV. Equation 2-1 shows how to specify a 16mA limit assuming a 10mOhm shunt resistor.

Equation 2-1. Regular current level.

$$CADRC = \frac{\mathrm{I}_{\mathrm{reg\_level}}}{\dfrac{26.86\,\mu\mathrm{V/bit}}{10\,\mathrm{m}\Omega}} = \frac{16\mathrm{mA}}{2.686\,\mathrm{mA/bit}} \approx 0\mathrm{x}06$$

Note that the RCC mode provides no info about the current flowing, except that the current consumption is below the Regular Current Level. So when using RCC mode in conjunction with sleep modes the current must be measured in other ways. An alternative is enabling the ICC interrupt, but a better alternative is often to estimate the current consumption of the application when in stand-by mode. The standby current can often be fairly accurately estimated, and one thus needs a way to keep track of time. This can either be done by enabling the ICC interrupt or using the watchdog timer.

2.2 Measuring current by measuring voltage

When measuring the current drawn by an application from a battery, the most common method is to measure the voltage over a shunt resistor. The measurement is compared to the internal CC-ADC voltage reference (CCVref) of the AVR, which for e.g. the Atmega16HVA is 110 mV. In IC and RCC-mode the step size of the CC-ADC is thus (110 mV/2^{12}) = 26.86 µV, and in ACC-mode (110 mV /2^{17}) = 0.839 µV.

Equation 2-2. Voltage step resolution of the CC-ADC with given bit resolution, (Note that unsigned resolution must be used in computation).

$$\text{Voltage step size} = \frac{V_{CCADC\_ref}}{2^{bit\_resolution}}$$

Note however that since the CC-ADC is a Sigma-Delta ADC, results become un-linear related to the input if the input exceeds 90% or the reference, which for Atmega16HVA means if the input exceeds 100mV. This should therefore be avoided.

Consider a smart battery using a 10 mΩ shunt resistor; In ACC-mode it will be possible to measure currents up to 10 A (10 mΩ * 10 A = 100 mV) and still operate in the linear region of the CC-ADC range. In general the shunt resistor should be selected large enough to provide a sufficiently accurate result for low currents and be small enough to not exceed the 100 mV limit for CC-ADC input. The power dissipation must also be considered, as 10A through a 10mΩ sense resistor will give 1W of heat. To compute the current Equation 2-3 can be used:

Equation 2-3. Conversion between CC-ADC result (Volt) and current.

$$I_{\text{bat}} = \frac{\text{CCADC}_{\text{result}}}{R_{\text{shunt}}}$$

2.3 CC-ADC Clock source

The CC-ADC, depending on device, can be clocked by the internal Slow RC oscillator. The frequency of the Slow RC is typically 131 kHz, but the frequencies vary due to process variations and die temperature. Please refer to the datasheet and application note AVR351 for detailed information.

The Slow RC frequency is predictable over temperature to give an accurate time reference. For maximum accuracy the measurements should be compensated for the clock variation, regardless of clock source.

2.4 Calibration and compensation

To get maximum accuracy from the CC-ADC the voltage reference (VREF) needs to be accurately calibrated, offset needs to be deducted and the sampling period needs to be known.

2.4.1 VREF calibration

To maximize accuracy of the CC-ADC, the reference needs to be accurate. The CCVref is based on the internal voltage reference Vref, which thus needs to be accurately calibrated. Vref can be calibrated to have less than 90ppm/°C drift from -10°C to +70°C and absolute accuracy of typically +/-1mV. The methods for calibrating the Vref can be found in the application note AVR353: Voltage Reference Calibration and Voltage ADC Usage.

2.4.2 CC-ADC conversion time compensation

The CCADC results should be compensated for actual clock period, regardless of clock source. The Slow RC oscillator frequency can be predicted directly from chip temperature. In other word, if ACC-mode is used and the result after 1024 ms is found to be 1 A, it cannot be stated that 1024 mA-seconds has been drawn from the battery. It is necessary to compensate for the actual period before the charge drawn from the battery can be determined. Methods for determining the oscillator frequencies and periods are described in the application note AVR351: Runtime calibration and compensation of RC oscillators. When the frequency (or period time) is known the CC-ADC conversion period is also known.

If the temperature of the AVR is stable, the frequency of the oscillator is stable too, and a fixed compensation can be used. However, if the temperature varies the frequency will vary and the compensation value has to be updated when the temperature of the die has changed more than e.g. 2°C (refer to datasheet for information about frequency variation as a function of temperature). To determine the charge drawn from the battery current measurement should be multiplied by the conversion period CC-ADC.

Equation 2-4. Compensation of power drawn from battery.

$$P = I_{bat} \cdot T_{RC} \cdot T_{conv} \qquad ; T_{conv} = \{128, 256, 512, 1024\}\text{ms}$$

2.4.3 Offset compensation

The CC-ADC result has an offset error, which applies to both ICC and ACC results. If one measures this offset it can be subtracted from succeeding measurements, but e.g. the ATmega16HVA features the possibility to change the polarity of the CC-ADC input. As the ACC results are usually accumulated, this can be used to cancel the offset by changing the CADPOL bit with a fixed period.

Equation 2-5. Offset cancellation by inverting input polarity.

$$CCADC_{pos} = x[n] + E_{offset}$$

$$CCADC_{neg} = -x[n+1] + E_{offset}$$

$$CCADC_{pos} + (-CCADC_{neg}) = (x[n] + x[n+1]) + (E_{offset} - E_{offset})$$

$$CCADC_{pos} + (-CCADC_{neg}) = x[n] + x[n+1]$$

Note that if the polarity of the CC-ADC input is altered during a conversion, by clearing or setting the CADPOL bit, an error will be introduced. To minimize this error the polarity should also be altered as soon as possible after an ACC conversion is completed. Since the error introduced when changing polarity does not cancel out on accumulation, the polarity change should not be done more often than required. As a rule of thumb the error cause by a polarity change will be approximately –0.9% if polarity is changed every second, while it will be in the range –0.015% if the polarity is changed every minute.

In ICC mode the polarity change might not be used to correct the offset error since the instantaneous value can be desired used without accumulation. The alternative is to run a calibration routine that determines the offset in production and corrects ICC results at run-time based on this calibration value.

The removal of the offset based on a production calibration can be used for the ACC measurements as well, but should be expected to be less accurate. If good accuracy should be reached with a static offset removal, the production calibration of the ACC offset would easily take several seconds, which is not desirable in mass production.

3 Code examples

A code example is provided with this application note. The code is documented using the code documentation tool Doxygen to give comprehensive source documentation. Compiler information and device settings are also included. To access this documentation, please open the readme.html file found in the code directory, which will direct a web browser to the html code documentation.

Headquarters

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

International

Atmel Asia
Unit 1-5 & 16, 19/F
BEA Tower, Millennium City 5
418 Kwun Tong Road
Kwun Tong, Kowloon
Hong Kong
Tel: (852) 2245-6100
Fax: (852) 2722-1369

Atmel Europe
Le Krebs
8, Rue Jean-Pierre Timbaud
BP 309
78054 Saint-Quentin-en-Yvelines Cedex
France
Tel: (33) 1-30-60-70-00
Fax: (33) 1-30-60-71-11

Atmel 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

Product Contact

Web Site
www.atmel.com/

Technical Support
avr@atmel.com

Sales Contact
www.atmel.com/contacts

Literature Request
www.atmel.com/literature

Disclaimer: The information in this document is provided in connection with Atmel products. No license, express or implied, by estoppel or otherwise, to any intellectual property right is granted by this document or in connection with the sale of Atmel products. **EXCEPT AS SET FORTH IN ATMEL'S TERMS AND CONDITIONS OF SALE LOCATED ON ATMEL'S WEB SITE, ATMEL ASSUMES NO LIABILITY WHATSOEVER AND DISCLAIMS ANY EXPRESS, IMPLIED OR STATUTORY WARRANTY RELATING TO ITS PRODUCTS INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT. IN NO EVENT SHALL ATMEL BE LIABLE FOR ANY DIRECT, INDIRECT, CONSEQUENTIAL, PUNITIVE, SPECIAL OR INCIDENTAL DAMAGES (INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF PROFITS, BUSINESS INTERRUPTION, OR LOSS OF INFORMATION) ARISING OUT OF THE USE OR INABILITY TO USE THIS DOCUMENT, EVEN IF ATMEL HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.** Atmel makes no representations or warranties with respect to the accuracy or completeness of the contents of this document and reserves the right to make changes to specifications and product descriptions at any time without notice. Atmel does not make any commitment to update the information contained herein. Unless specifically provided otherwise, Atmel products are not suitable for, and shall not be used in, automotive applications. Atmel's products are not intended, authorized, or warranted for use as components in applications intended to support or sustain life.