



... the world's most energy friendly microcontrollers

Analog to Digital Converter

AN0021 - Application Note

Introduction

This application note describes how to use the EFM32 Analog to Digital Converter to convert an analog input voltage to a digital value. Many aspects of the ADC, including inputs, references and the different operating modes are described. Calibration routines for offset and gain are also included.

This application note includes:

- This PDF document
- Source files (zip)
 - Example C-code
 - Multiple IDE projects



1 Analog to Digital Converter

1.1 Introduction

The EFM32 ADC is a Successive Approximation Register (SAR) architecture. The maximum resolution is 12 bits, and one million samples per second can be achieved at this resolution. The integrated input mux can select ADC-input from 8 external pins or 6 internal signals. With the help of PRS and DMA, the ADC can operate without CPU intervention, minimizing current consumption or allow the core to do other work. The ADC can be clocked at different speeds, and run at different warm-up modes to reduce the energy consumption even further.

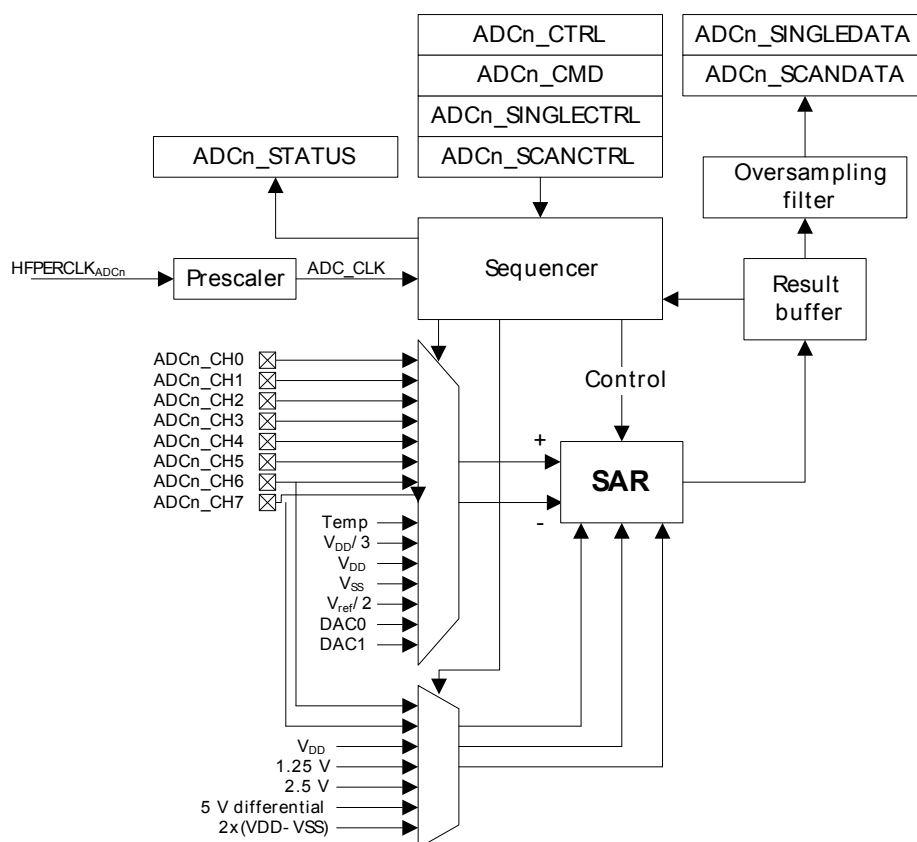
This application note shows general operation and usage of the ADC. In addition, advanced features and power saving techniques are described. Software examples of ADC operation both with DMA and PRS are included. Offset and Gain Calibration of the ADC is also described and included in the software examples.

For extremely low power periodic ADC sampling, a software example which enter EM2 between each ADC sample is also included. For sampling frequencies below a couple of kilohertz, this is the best way to do low power ADC sampling.

1.2 Overview

Figure 1.1 (p. 2) illustrates the internal connections of the inputs, reference selection and registers.

Figure 1.1. Analog to Digital Converter Overview



2 General Operation

2.1 Clock Selection

The ADC clock is supplied by an ADC-clock prescaler which divides the peripheral clock (HPPERCLK) by a selectable factor between 1 and 128. Any factor between 1 and 128 can be chosen by writing to a value between 0 and 127 the PRESC bits in ADCn_CTRL. The resulting ADC clock should be between 32 kHz and 13 MHz.

2.2 Input Selection

The 8 external inputs can either be selected as single ended inputs, or combined to allow for differential inputs. Two neighbouring inputs are used in differential mode, for instance channel 0 and channel 1 is one differential pair, channel 2 and channel 3 another. The lowest channel number is the positive differential input. The DIFF bit in ADCn_SINGLECTRL or ADCn_SCANCTRL enables differential mode.

The ADC input signals are shielded fairly well against other noisy signals within the EFM32. If high ADC accuracy is needed, it is advisable to not use any of the unused ADC input pins for noise inducing activities such as serial communication.

2.2.1 Single Ended Mode

In single ended mode the input signal is measured with ground as the lower reference. The voltage span between 0 V and the selected reference is divided in small steps according to the selected resolution. The result will be an unsigned number between 0 and $2^{\text{resolution}}$, indicating where the input voltage is located in the span between ground and the reference voltage.

2.2.2 Differential Mode

In differential mode the measured value is the difference between two inputs. Since one input is defined as the positive input, and the other is defined as the negative input, the difference can be positive or negative depending on which input is higher. As a result, the conversion result is a signed number, represented in two's complement form. If the negative input is higher than the positive input, the converted value will be negative. The ADC can not measure negative voltages or voltages larger than the reference voltage, and it is important to keep both inputs within the electrical limits of the device.

2.3 Reference Selection

In order to convert an analog voltage to a digital value the ADC needs a reference voltage to which it compares the incoming analog voltage. Since the ADC cannot measure voltages larger than the reference voltage, the reference voltage should be above the maximum expected measured voltage.

The reference can be selected from the following sources:

- 1.25 V internal bandgap
- 2.5 V internal bandgap
- V_{DD}
- 5 V internal differential bandgap
- External single ended input from pin 6
- Differential input, 2x(pin 6 - pin 7)
- Unbuffered $2 \times V_{DD}$

The internal bandgap references keeps a stable voltage regardless of the device supply voltage, as long as the supply voltage is higher than the reference voltage.

2.4 Single Conversion

In single conversion mode the ADC will convert one input either one time or continuously if the REP bit in ADCn_SINGLECTRL is set. If the result is not read before the next result is ready, the first result will be overwritten and the overflow flag SINGLEOF in ADCn_IF will be set.

2.5 Scan Conversion

In scan conversion mode the ADC can be configured to convert a sequence of different inputs, either one time or continuously if the REP bit in ADCn_SCANCTRL is set. The result after one conversion needs to be read out before the next conversion is finished. DMA can be used to transfer results to RAM after each conversion. All the results can then be read from RAM after the sequence has finished.

2.6 Periodic Measurements

Often the ADC needs to sample a voltage with fixed intervals, and preferably without cpu intervention. Periodic measurements can be achieved by using the Peripheral Reflex System and a timer that runs continuously and sends a PRS-pulse to the ADC with specific intervals. When the ADC receives a PRS-pulse it triggers a conversion start. A conversion finished interrupt can then handle the result. The conversion result can also be directly transferred to RAM using DMA.

2.7 Warm-Up Modes

The ADC and bandgap references, if selected, needs to be warmed up before a conversion is started. If fast sampling in scan mode is needed, the ADC with bandgap references can be kept warm between samples. There are four available warm-up modes:

- NORMAL: ADC and references are shut off when there are no samples waiting.
- FASTBG: Bandgap warm-up is eliminated, but with reduced reference accuracy.
- KEEPSCANREFWARM: The reference selected for scan mode is kept warm. The ADC will still need to be warmed up before conversion.
- KEEPADCWARM: The ADC and the reference selected for scan mode are kept warm.

It is only the reference selected for scan mode that is kept warm. To avoid warm-up time in single conversion mode, the single conversion reference needs to be the same as the scan conversion reference.

The minimum warm-up times for the ADC and bandgap reference is 1 μ s and 5 μ s respectively. The timing is done automatically by the ADC, given that a proper time base is given in the TIMEBASE bits in ADCn_CTRL. The TIMEBASE must be set to the number of HFPERCLK which corresponds to at least 1 μ s. The TIMEBASE only affects the timing of the warm-up sequence and not the ADC_CLK.

Before entering Energy Modes 2 or 3, the ADC should be stopped and WARMUPMODE in ADCn_CTRL written to 0 to reduce current consumption.

3 Advanced Features

3.1 Comparator and Bandgap Bias

The current consumption of the bandgap reference and the ADC comparator can be adjusted through the BIASPROG, HALFBIAS and COMPBIAS registers. But this will also affect the performance and bandwidth of the bandgap reference and comparator. The default register values should be used to ensure correct operation of the ADC within the specified clock speed range and with all configurations.

3.2 Oversampling

If higher than 12-bit accuracy is needed, the ADC can automatically sample many times and average the result in hardware. The number of samples for each averaged result can be selected as 2^n for $n = [1..12]$. If oversampling is enabled, the ADC result will not be ready until all the samples are converted and averaged.

The result is averaged by accumulating samples and right shifting the result. For 2x, 4x, 8x and 16x oversampling the result is not right shifted, instead the result consists of more than 12 bits. See Table 3.1 (p. 5) for oversampling result representation.

Table 3.1. Oversampling Result Shifting and Resolution

Oversampling setting	# right shifts	Result Resolution # bits
2x	0	13
4x	0	14
8x	0	15
16x	0	16
32x	1	16
64x	2	16
128x	3	16
256x	4	16
512x	5	16
1024x	6	16
2048x	7	16
4096x	8	16

If the samples of the measured signal are affected by uncorrelated random noise added to each sample, oversampling and averaging can be used to increase the signal to noise ratio. The number of additional samples needed to get n additional bits of meaningful data in the result is given by:

Note

$$\text{samples} = 2^{2n}$$

In practice this can be used to get for example 16 bit accuracy at the expense of using more time per sample because the signal must be sampled many times. To get 4 additional bits of meaningful data for the 12-bit ADC in the EFM32, the number of samples needed equals $2^{2 \cdot 4}$ which is 256 samples. With a maximum 12-bit sampling frequency of 1 MSamples/s, the maximum sampling frequency at 16 bit averaged accuracy is around 3.9 kSamples/s.

3.3 Peripheral Reflex System

The ADC can be configured as both a consumer and producer of PRS signals. Both scan and single conversions can be triggered by a PRS signal, PRS-signals can also be produced when a conversion is finished.

A common setup that is using the Peripheral Reflex System of the EFM32 is a timer-triggered ADC conversion. The timer would be the producer and the ADC would be the consumer, starting a conversion when the PRS signal arrives.

3.4 Direct Memory Access

After the DMA module is configured the ADC can send requests to the DMA module when either a single conversion or scan conversion is finished. The DMA module can then transfer the ADC result from the ADC register to memory without the CPU participating in the transaction.

3.5 Calibration

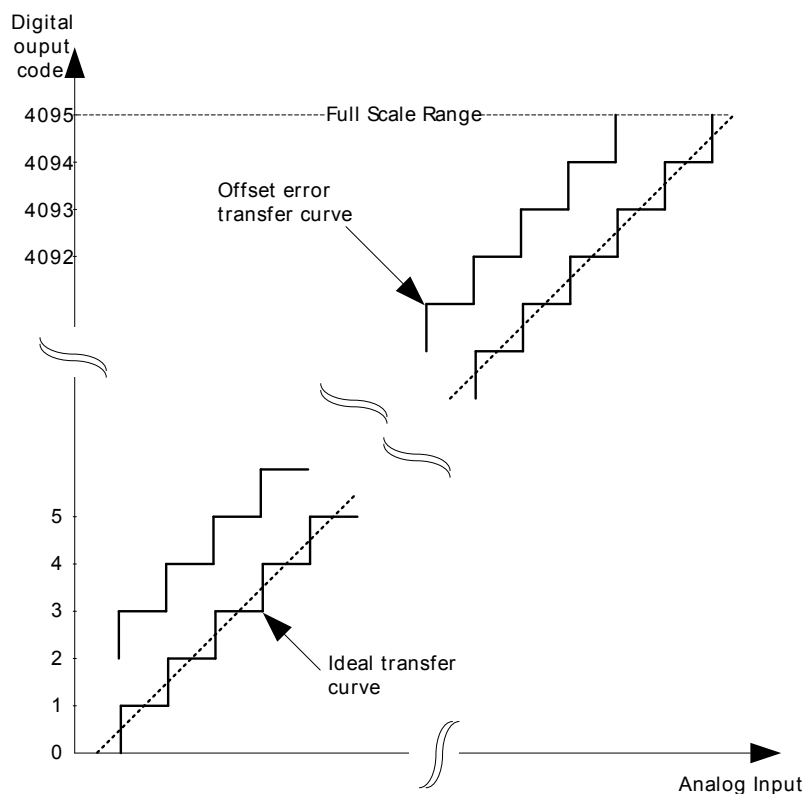
To get the highest possible performance from the ADC it should be calibrated for offset and gain errors for each reference voltage. Differences between devices due to production variations makes the calibration functionality necessary. During production, calibration values for offset and gain at 25 degrees celsius for the internal references are programmed in the device. The values are accessible to the user, and the calibration values for the 1.25V reference are written to the ADC calibration register after reset. If the EFM32 is operated at different temperatures, or with external references, calibration should be performed at runtime to find the correct values for that temperature and reference.

In addition to offset and gain errors, the ADC result will be affected by non-linear effects such as INL (Integral Non-Linearity) and DNL (Differential Non Linearity). INL and DNL are further described in the datasheet for the device, and are not subject to calibration.

3.5.1 Offset Calibration

An offset error is a constant offset of the real conversion result from the ideal result. This means that the output will be offset by a constant amount over the whole conversion range. Offset error is illustrated in Figure 3.1 (p. 7) .

The simplest way to calibrate the offset is to select differential mode with shorted inputs. The difference between to shorted lines is a simple way to get a true 0V voltage. The ADC has a built-in feature for shorting the differential inputs. This can be done in software without any external components. The calibration register is then adjusted until the converted output is as close to 0 as it can get. Adjusting offset at 0 V also simplifies the gain calibration.

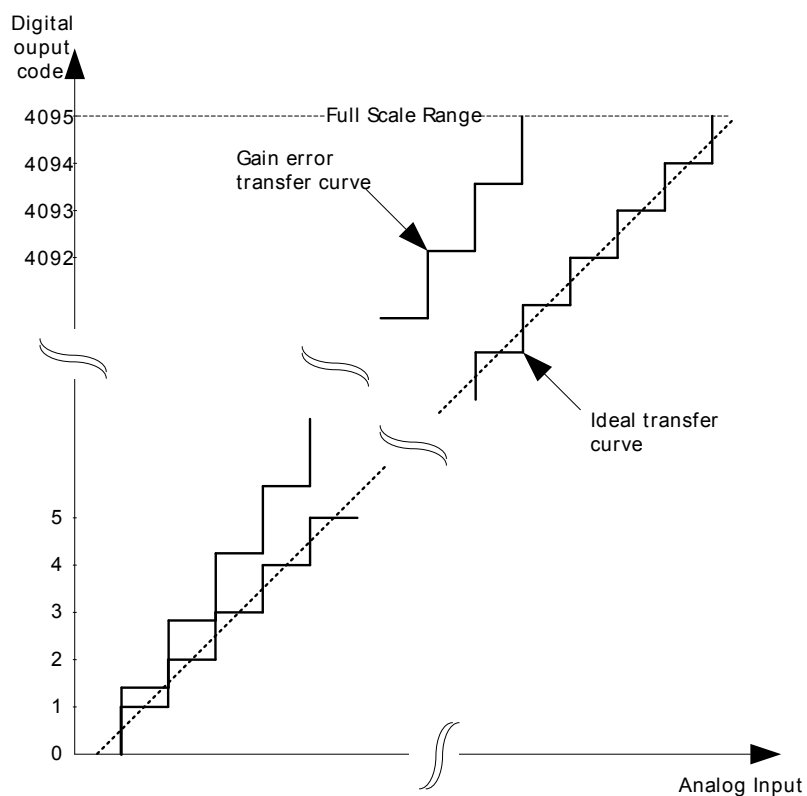
Figure 3.1. ADC Offset Error

ADC offset error gives a constant offset, either above or below the ideal output.

3.5.2 Gain Calibration

A gain error, if the offset is calibrated at 0 V, is a linearly increasing offset from the correct conversion result as the input voltage increases. This is illustrated in Figure 3.2 (p. 8). Gain calibration should be done after the offset is calibrated correctly at 0 V. This is because the gain calibration routine assumes that the measurement at 0 has 0 offset. The gain can then be calibrated by only looking at one other conversion point, preferably the top value to get the highest possible accuracy.

By applying a known and fixed voltage that corresponds to the top of the ADC range which is the ADC reference voltage. The gain can be corrected by adjusting the calibration register until the ADC output corresponds to the highest possible result value.

Figure 3.2. ADC Gain Error

If offset is calibrated at 0, an ADC gain error gives a linearly increasing offset either above or below the ideal output.

4 Software Examples

4.1 Single Conversion

The single conversion example configures the ADC to sample VDD/3 with the 1.25 V bandgap reference. This setup makes it possible to measure the supply voltage to the chip. The voltage is calculated and displayed on the segment lcd which is updated 10 times every second.

4.2 Scan Conversion with DMA Transfer

In this example the ADC is configured to use scan mode on channel 2, 3 and 4. The DMA is set up to transfer the ADC result from one conversion to RAM while the ADC continues with the next conversion. When a conversion of all three channels is complete, the result is printed on the segment lcd. The three channels are selected because they are connected to the accelerometer on the development kit, and the pins are easily accessible on the starter kits.

4.3 Oversampling

The single conversion example is now expanded to include oversampling. The result should be a bit more stable. The main loop starts an oversampled ADC conversion every 100 ms, waits for completion and prints the result in the same way as the single conversion example.

4.4 PRS Triggered Sampling

The PRS example configures TIMER0 to overflow roughly 3 times each second. On each overflow, a PRS signal is sent to the ADC which in turn triggers a single conversion. The ADC input selected for this example is channel 5. The EFM32 stays in EM1, and wakes up when the conversion is complete.

4.5 Enter EM2 Between ADC Samples

The two ADC + EM2 examples demonstrates the ultra low energy consumption achievable for low sampling speeds by utilizing deep sleep mode (EM2) between each ADC sample.

4.5.1 Optimized Loop ADC + EM2 Example

For the optimized example the time spent in EM0/EM1 during each ADC conversion is minimized. The sample loop is optimized with the "wait for event"-instruction instead of interrupts (see AN0039 EFM32 Interrupt Handling). Even though the ADC in 12 bit mode can finish each sample in 13 clock cycles + warmup time, the amount of clock cycles spent in EM0/EM1 is a bit higher. The overhead is caused by the delay until the first instruction is executed, the clearing of interrupt flags, storing the ADC sample and changing between energy modes. For the optimized loop example, the break-even sampling frequency compared to the autonomous EM1 approach is around 20 kHz. (The EM1 approach which utilizes PRS triggering and DMA is described in the DMA application note, AN0013.)

4.5.2 Interrupt Driven ADC + EM2 Example

This example start each ADC sample inside the RTC-interrupt routine. This approach allows other peripherals to issue interrupts while the ADC sampling sequence is active. The drawback is twice the average energy consumption compared to the optimized approach. For the interrupt driven example, the break-even sampling frequency compared to the standard EM1 (ADC + PRS + DMA) approach is approximately 5 kHz.

4.5.3 Typical ADC Current Consumption

Figure 4.1. Total current consumption when sampling at different frequencies.

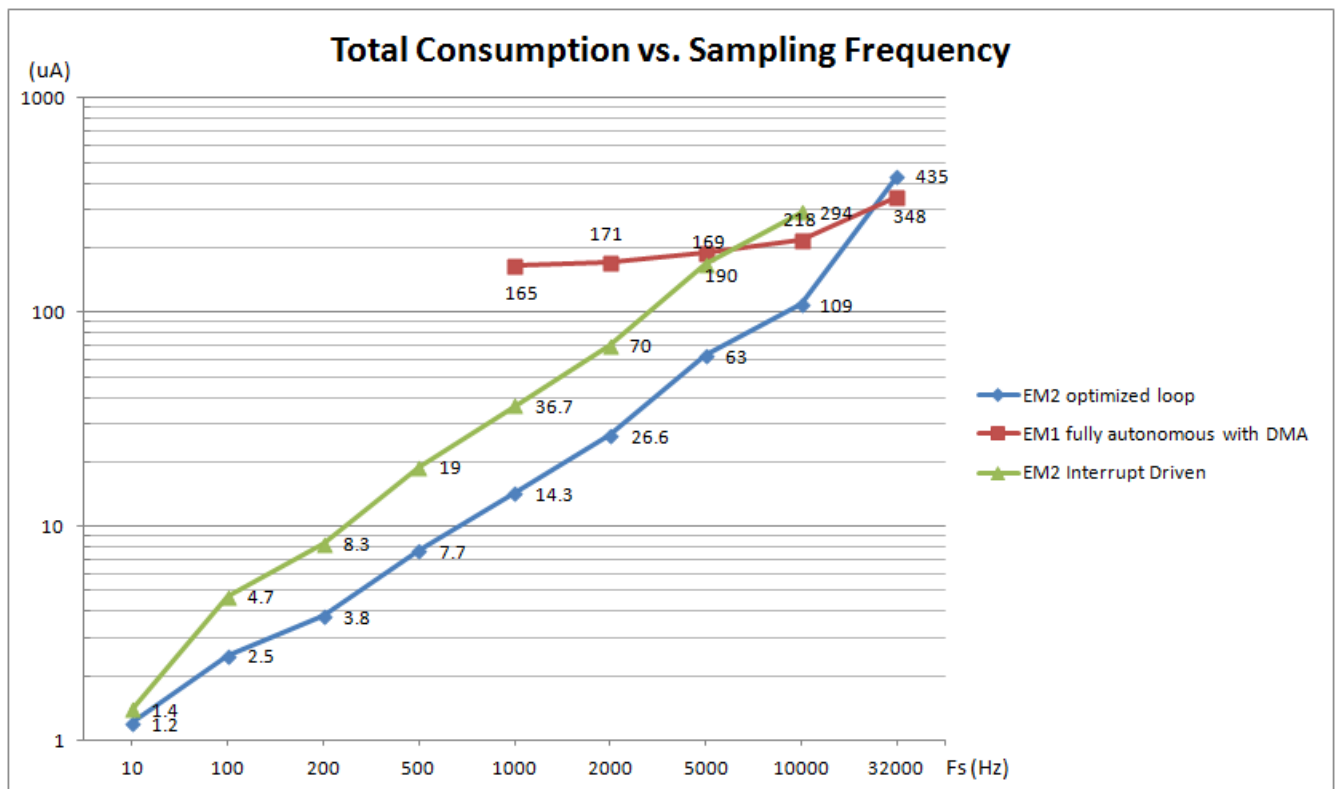


Figure 4.1 (p. 10) indicates the average current consumption for 12 bit samples at different sampling frequencies, conditions are 3.3V and room temperature for EFM32TG840F32. The three different graphs shows the consumption for the optimized EM2 example, the interrupt driven EM2 example and the EM1 only example included with the AN0013 EFM32 DMA application note.

4.6 Calibration

4.6.1 Offset Calibration

The offset calibration routine must be performed before gain calibration and roughly follows these steps:

1. Selects the wanted reference.
2. Set acquisition time to 16 cycles.
3. Selects DIFF0 as input and enable differential input mode.
4. A binary search is used to find the offset calibration value. The result of the binary search is written to the ADC calibration register.

4.6.2 Gain Calibration

The gain calibration routine roughly follows these steps:

1. Selects external ADC channel number 4.
2. A stable voltage equal to the reference voltage selected for calibration should be applied to the ADC channel 4 pin.
3. A binary search is used to find the gain calibration value. The search terminates on a value a few LSB's lower than the maximum ADC value to avoid overshoot. The result of the binary search is written to the ADC calibration register.

The code example combines both the offset and gain calibration routines in one function. It resets and configures the ADC before performing the offset and gain calibrations.

5 Revision History

5.1 Revision 1.09

2013-09-03

New cover layout

5.2 Revision 1.08

2013-05-08

Added software projects for ARM-GCC and Atollic TrueStudio.

Added comment about noise coupled between ADC pins if doing serial communication on ADC pins.

5.3 Revision 1.07

2012-11-12

Adapted software projects to new kit-driver and bsp structure.

5.4 Revision 1.06

Updated EM2 ADC sampling graph with values for EFM32 TG.

Added EM2 ADC sampling examples for Tiny Gecko.

5.5 Revision 1.05

2012-08-13

Added projects for the Tiny and Giant Gecko STKs.

Added EM2 ADC sampling software examples.

5.6 Revision 1.04

2012-04-20

Adapted software projects to new peripheral library naming and CMSIS_V3.

5.7 Revision 1.03

2011-11-17

Updated IDE project paths with new kits directory.

5.8 Revision 1.02

2011-05-18

Updated projects to align with new bsp version

5.9 Revision 1.01

2010-11-16

Changed example folder structure, removed build and src folders.

Updated chip init function to newest efm32lib version.

Updated register defines in code to match newest efm32lib release.

5.10 Revision 1.00

2010-10-14

Initial revision.

A Disclaimer and Trademarks

A.1 Disclaimer

Silicon Laboratories intends to provide customers with the latest, accurate, and in-depth documentation of all peripherals and modules available for system and software implementers using or intending to use the Silicon Laboratories products. Characterization data, available modules and peripherals, memory sizes and memory addresses refer to each specific device, and "Typical" parameters provided can and do vary in different applications. Application examples described herein are for illustrative purposes only. Silicon Laboratories reserves the right to make changes without further notice and limitation to product information, specifications, and descriptions herein, and does not give warranties as to the accuracy or completeness of the included information. Silicon Laboratories shall have no liability for the consequences of use of the information supplied herein. This document does not imply or express copyright licenses granted hereunder to design or fabricate any integrated circuits. The products must not be used within any Life Support System without the specific written consent of Silicon Laboratories. A "Life Support System" is any product or system intended to support or sustain life and/or health, which, if it fails, can be reasonably expected to result in significant personal injury or death. Silicon Laboratories products are generally not intended for military applications. Silicon Laboratories products shall under no circumstances be used in weapons of mass destruction including (but not limited to) nuclear, biological or chemical weapons, or missiles capable of delivering such weapons.

A.2 Trademark Information

Silicon Laboratories Inc., Silicon Laboratories, the Silicon Labs logo, Energy Micro, EFM, EFM32, EFR, logo and combinations thereof, and others are the registered trademarks or trademarks of Silicon Laboratories Inc. ARM, CORTEX, Cortex-M3 and THUMB are trademarks or registered trademarks of ARM Holdings. Keil is a registered trademark of ARM Limited. All other products or brand names mentioned herein are trademarks of their respective holders.

B Contact Information

Silicon Laboratories Inc.

400 West Cesar Chavez

Austin, TX 78701

Please visit the Silicon Labs Technical Support web page:

<http://www.silabs.com/support/pages/contacttechnicalsupport.aspx>

and register to submit a technical support request.

Table of Contents

1. Analog to Digital Converter	2
1.1. Introduction	2
1.2. Overview	2
2. General Operation	3
2.1. Clock Selection	3
2.2. Input Selection	3
2.3. Reference Selection	3
2.4. Single Conversion	4
2.5. Scan Conversion	4
2.6. Periodic Measurements	4
2.7. Warm-Up Modes	4
3. Advanced Features	5
3.1. Comparator and Bandgap Bias	5
3.2. Oversampling	5
3.3. Peripheral Reflex System	6
3.4. Direct Memory Access	6
3.5. Calibration	6
4. Software Examples	9
4.1. Single Conversion	9
4.2. Scan Conversion with DMA Transfer	9
4.3. Oversampling	9
4.4. PRS Triggered Sampling	9
4.5. Enter EM2 Between ADC Samples	9
4.6. Calibration	10
5. Revision History	12
5.1. Revision 1.09	12
5.2. Revision 1.08	12
5.3. Revision 1.07	12
5.4. Revision 1.06	12
5.5. Revision 1.05	12
5.6. Revision 1.04	12
5.7. Revision 1.03	12
5.8. Revision 1.02	12
5.9. Revision 1.01	12
5.10. Revision 1.00	13
A. Disclaimer and Trademarks	14
A.1. Disclaimer	14
A.2. Trademark Information	14
B. Contact Information	15
B.1.	15

List of Figures

1.1. Analog to Digital Converter Overview 2

3.1. ADC Offset Error 7

3.2. ADC Gain Error 8

4.1. Total current consumption when sampling at different frequencies. 10

List of Tables

3.1. Oversampling Result Shifting and Resolution 5

silabs.com

