Freescale Semiconductor

ADC example in Kinetis Design Studio (KDS) with FDRM-K64f

By:

Paul Garate
Augusto Panecatl

Description

In this document you will find a detailed step by step guide of how to configure the ADC on Kinetis K devices using Kinetis Design Studio, you will be reading a value from the microcontroller's internal temperature sensor.

1. Clock Gating

As in the previous examples, the first step is to enable the clock gate corresponding to the module we will use, in this case ADC0

12.2.13 System Clock Gating Control Register 6 (SIM_SCGC6)

DAC0, FTM2, and RNGA can be accessed through both AIPS0 and AIPS1. When accessing through AIPS1, define the clock gate control bits in the SCGC2 and SCGC3. When accessing through AIPS0, define the clock gate control bits in SCGC6.

Address: 4004_7000h base + 103Ch offset = 4004_803Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	DAC0	1	RTC	0	ADC0	FTM2	FTM1	FTM0	PIT	PDB	USBDCD	0		CRC	0	
W																
Reset	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	I2S	0	SPI1	SPI0	0		RNGA	0				FLEXCAN0	0		DMAMUX	FTF
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

SIM_SCGC6_PORTn_MASK is defined as mask to enable the module's clock, where "n" corresponds to the specific port we wish to activate, i.e:

SIM_SCGC6 = SIM_SCGC5_ADC0_MASK;

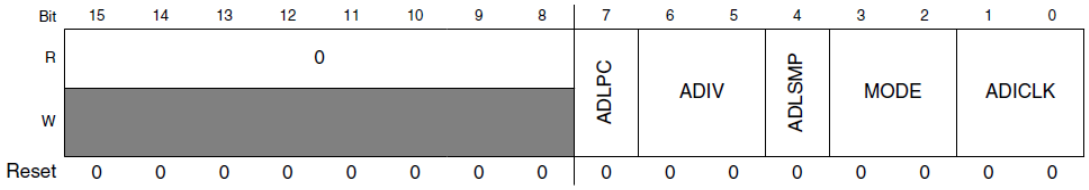
By declaring the mask we are writing `0x8000000` to the ***SIM_SCGC6*** register, setting up the 27th bit of the System Clock Gating Control Register 6 which enables ADC0 ;

```
SIM_SCGC6 |= SIM_SCGC6_ADC0_MASK;            /*Enable the ADC0 Clock*/
```

2. ADC Configuration

The K64F's ADC contains a lot of registers; you can see the section 35.3 in the Reference Manual (page 829). In this particular case we will just use 2 registers which are ***ADCx_CFG1*** and ***ADC0_SC1A***.

The first one (***ADCx_CGF1***) selects the operation mode, clock source, clock divide, and configuration for low power and long sample time.

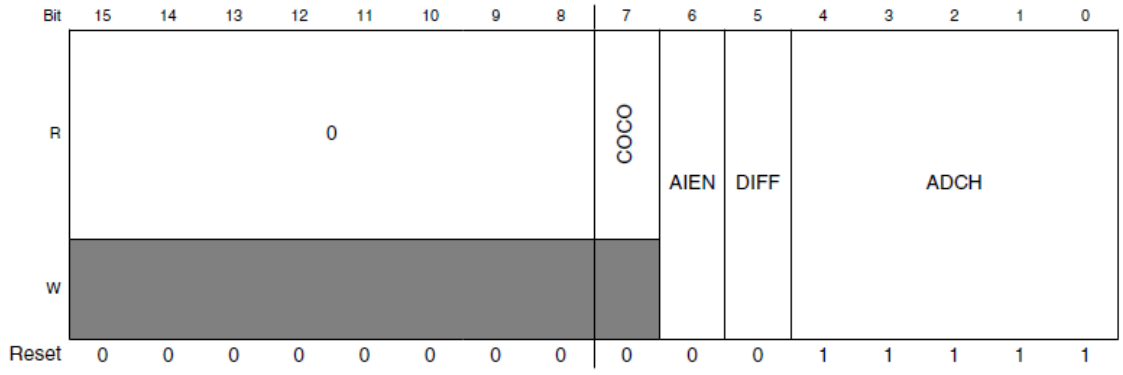


We will first configure the conversion mode resolution.

3–2 MODE	Conversion mode selection Selects the ADC resolution mode. 00 When DIFF=0:It is single-ended 8-bit conversion; when DIFF=1, it is differential 9-bit conversion with 2's complement output. 01 When DIFF=0:It is single-ended 12-bit conversion ; when DIFF=1, it is differential 13-bit conversion with 2's complement output. 10 When DIFF=0:It is single-ended 10-bit conversion. ; when DIFF=1, it is differential 11-bit conversion with 2's complement output 11 When DIFF=0:It is single-ended 16-bit conversion..; when DIFF=1, it is differential 16-bit conversion with 2's complement output

The MODE register allows you to select the ADC's resolution, in this example we will select a 16 bits resolution, thus; we need to write ***11*** to the MODE field.

The second one (***ADCx_SC1n***) is used to select the ADC channel input, select the single ended or differential mode, enable or disable the interrupt after conversion completion and it also contains the conversion complete flag (***COCO***).



The ADCH field has 5 bits to fill, and a lot of options, but we will only use it just to disable the module.

Field	Description
	01100 When DIFF=0, AD12 is selected as input; when DIFF=1, it is reserved. 01101 When DIFF=0, AD13 is selected as input; when DIFF=1, it is reserved. 01110 When DIFF=0, AD14 is selected as input; when DIFF=1, it is reserved. 01111 When DIFF=0, AD15 is selected as input; when DIFF=1, it is reserved. 10000 When DIFF=0, AD16 is selected as input; when DIFF=1, it is reserved. 10001 When DIFF=0, AD17 is selected as input; when DIFF=1, it is reserved. 10010 When DIFF=0, AD18 is selected as input; when DIFF=1, it is reserved. 10011 When DIFF=0, AD19 is selected as input; when DIFF=1, it is reserved. 10100 When DIFF=0, AD20 is selected as input; when DIFF=1, it is reserved. 10101 When DIFF=0, AD21 is selected as input; when DIFF=1, it is reserved. 10110 When DIFF=0, AD22 is selected as input; when DIFF=1, it is reserved. 10111 When DIFF=0, AD23 is selected as input; when DIFF=1, it is reserved. 11000 Reserved. 11001 Reserved. 11010 When DIFF=0, Temp Sensor (single-ended) is selected as input; when DIFF=1, Temp Sensor (differential) is selected as input. 11011 When DIFF=0,Bandgap (single-ended) is selected as input; when DIFF=1, Bandgap (differential) is selected as input. 11100 Reserved. 11101 When DIFF=0,V_{REFSH} is selected as input; when DIFF=1, -V_{REFSH} (differential) is selected as input. Voltage reference selected is determined by SC2[REFSEL]. 11110 When DIFF=0,V_{REFSL} is selected as input; when DIFF=1, it is reserved. Voltage reference selected is determined by SC2[REFSEL]. 11111 Module is disabled.

Taking into consideration the table values we need to write ***0x1F*** to the ***ADCH*** field n order to disable the module.

Here are the two lines used to configure the ADC0_CFG1 and ADC0_SC1A registers:

```
ADC0_CFG1 |= ADC_CFG1_MODE(3);                /*16bits ADC*/
ADC0_SC1A |= ADC_SC1_ADCH(31);                /*Disable the module, ADCH = 1111 */
```

*ADC_CFG1_MODE(3) = 0xC
*ADC_SC1_ADCH(31) = 0x1F

3. Measurement function

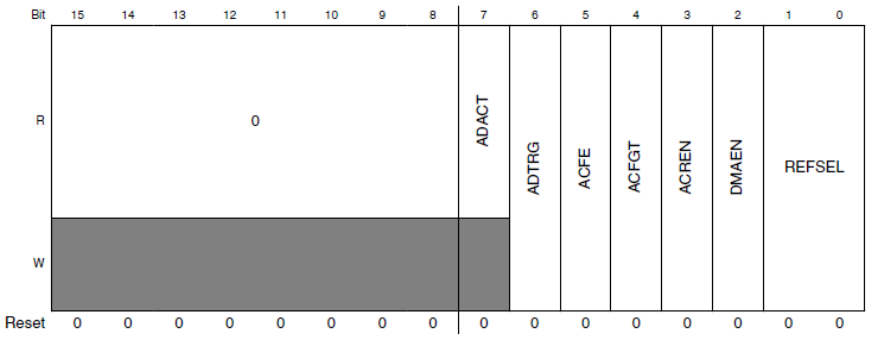
Once the module is configured we need create a function to perform the ADC readings. The function used in the code example is shown below:

```
unsigned short ADC_read16b(void)
{
      ADC0_SC1A = 26 & ADC_SC1_ADCH_MASK;          //Write to SC1A to start conversion
      while(ADC0_SC2 & ADC_SC2_ADACT_MASK);        //Conversion in progress
      while(!(ADC0_SC1A & ADC_SC1_COCO_MASK));     //Wait until conversion complete
      return ADC0_RA;
}
```

The first line assigns selects the ADC channel assigned to the internal temperature sensor (channel 26) by writing ***0x1A*** to the ***ADCH*** field.

Field	Description
	01100 When DIFF=0, AD12 is selected as input; when DIFF=1, it is reserved.
	01101 When DIFF=0, AD13 is selected as input; when DIFF=1, it is reserved.
	01110 When DIFF=0, AD14 is selected as input; when DIFF=1, it is reserved.
	01111 When DIFF=0, AD15 is selected as input; when DIFF=1, it is reserved.
	10000 When DIFF=0, AD16 is selected as input; when DIFF=1, it is reserved.
	10001 When DIFF=0, AD17 is selected as input; when DIFF=1, it is reserved.
	10010 When DIFF=0, AD18 is selected as input; when DIFF=1, it is reserved.
	10011 When DIFF=0, AD19 is selected as input; when DIFF=1, it is reserved.
	10100 When DIFF=0, AD20 is selected as input; when DIFF=1, it is reserved.
	10101 When DIFF=0, AD21 is selected as input; when DIFF=1, it is reserved.
	10110 When DIFF=0, AD22 is selected as input; when DIFF=1, it is reserved.
	10111 When DIFF=0, AD23 is selected as input; when DIFF=1, it is reserved.
	11000 Reserved.
	11001 Reserved.
	11010 When DIFF=0, Temp Sensor (single-ended) is selected as input; when DIFF=1, Temp Sensor (differential) is selected as input.
	11011 When DIFF=0,Bandgap (single-ended) is selected as input; when DIFF=1, Bandgap (differential) is selected as input.
	11100 Reserved.
	11101 When DIFF=0,V_{REFSH} is selected as input; when DIFF=1, -V_{REFSH} (differential) is selected as input. Voltage reference selected is determined by SC2[REFSEL].
	11110 When DIFF=0,V_{REFSL} is selected as input; when DIFF=1, it is reserved. Voltage reference selected is determined by SC2[REFSEL].
	11111 Module is disabled.

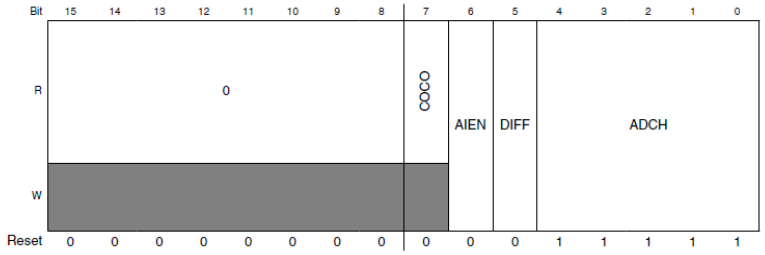
The second line is used to check if the conversion is still in progress. The code will remain in the wait cycle until the ***ADACT*** flag is erased.



ADCx_SC2 field descriptions

Field	Description
31–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
7 ADACT	Conversion Active
	Indicates that a conversion or hardware averaging is in progress. ADACT is set when a conversion is initiated and cleared when a conversion is completed or aborted. 0 Conversion not in progress. 1 Conversion in progress.

The third line checks if the conversion has been completed by checking the status of the conversion complete flag ***COCO***.

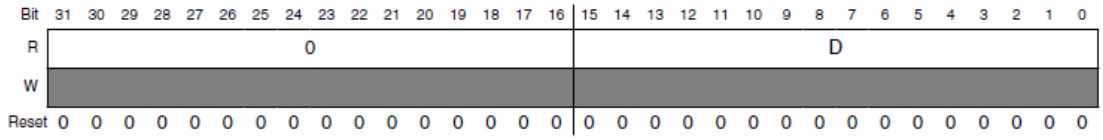


ADCx_SC1*n* field descriptions

Field	Description
31–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
7 COCO	Conversion Complete Flag This is a read-only field that is set each time a conversion is completed when the compare function is disabled, or SC2[ACFE]=0 and the hardware average function is disabled, or SC3[AVGE]=0. When the compare function is enabled, or SC2[ACFE]=1, COCO is set upon completion of a conversion only if the compare result is true. When the hardware average function is enabled, or SC3[AVGE]=1, COCO is set upon completion of the selected number of conversions (determined by AVGS). COCO in SC1A is also set at the completion of a calibration sequence. COCO is cleared when the respective SC1n register is written or when the respective Rn register is read. 0 Conversion is not completed. 1 Conversion is completed.

The last line reads the result register and returns the conversion data corresponding to the ADC measurement of the selected channel.

Address: Base address + 10h offset + (4d × i), where i=0d to 1d



ADCx_R*n* field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
15–0 D	Data result

4. Code

The main code implements a call to the ADC function, the bADCData variable stores the Data Result, and the Delay function controls the time between measurements.

```
    for (;;)
    {
       bADCData = ADC_read16b();
       DelayFunction();
    }



void DelayFunction (void)
{
       unsigned long Counter = 0xFFFFF;
       do
       {
              Counter--;
       }while(Counter);
}
```

*The code also includes some UART configurations, to understand the UART module refer to the ***"Serial (UART) example in Kinetis Development Software (KDS) with FDRM-K64f***" Document.