

## ATxmega 演示代码之 ADC

——基于 ATxmega64A3-EK 和 JTAGICE mkII-CN

文档编号	MAN0010A_CH				
文档版本	Rev. A				
文档摘要	详细讲述了 XMEGA64A3 的 ADC 使用方法				
关键词	AVR、XMEGA、ATxmega64A3-EK、USB JTAGICE mkII-CN、ADC				
创建日期	2010-05-17	创建人员	Kiler	审核人员	<a href="#">Hotislandn</a>
文档类型	公开发布/仿真器配套文件				
版权信息	<a href="#">Mcuzone</a> 原创文档，转载请注明出处				

## 更新历史

版本	时间	更新	作者
Rev. A	2010-05-17	初始创建	Kiler

微控电子 乐微电子  
杭州市登云路 639 号 2B143  
销售 TEL: +86-571-89908193 13957118045  
支持 TEL: 13957118045 18913989166  
FAX: +86-571-88908193  
[www.mcuzone.com](http://www.mcuzone.com) [www.atarm.com](http://www.atarm.com)

# 1. 概述

ATxmega 是 ATMEL 推出的一款全新 MCU，与之前的 AVR 相比速度更快，ADC 和 DAC 性能更佳，功耗更低，而且有多达 8 个串口，因此在工控领域有较大用途。本系列文档以 ATxmega64A3-EK 开发板和本站的 USB AVR JTAGICE mkII-CN 仿真器为平台演示 ATxmega 的一些片上外设的操作。



本文演示 64A3 的 ADC 操作  
ATxmega64A3 具备两个 8 通道的 12bit 2Mps ADC。

# 2. 寄存器概述

## CTRLA - ADC Control Register A

Bit	7	6	5	4	3	2	1	0	
+0x00	DMASEL[1:0]		CH[3:0]START			FLUSH	ENABLE		CTRLA
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

控制寄存器 A，第 0 位用于使能 ADC，详情请参阅数据手册第 25 章。

## CTRLB - ADC Control Register B

Bit	7	6	5	4	3	2	1	0	
+0x01	-	-	-	CONVMODE	FREERUN	RESOLUTION[1:0]		-	CTRLB
Read/Write	R	R	R	R/W	R/W	R/W	R/W	R	
Initial Value	0	0	0	0	0	0	0	0	

控制寄存器 B，第 4 位用于决定转换结果是否有符号，1—有符号，0—无符号；第 3 位置 1，则运行于自由转换模式；第 2 至第 1 位，00—结果 12 位右对齐，10—结果 8 位右对齐，11—结果 12 位左对齐。

## Mcuzone Application Notes

### REFCTRL - ADC Reference Control register

Bit	7	6	5	4	3	2	1	0	
+0x02	-	-	REFSEL[1:0]		-	-	BANDGAP	TEMPREF	REFCTRL
Read/Write	R	R	R/W	R/W	R	R	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

基准源控制寄存器，第 5 至第 4 位，选择基准源，00—内部 1V，01—内部 Vcc/1.6，10—外部输入 AREF(PORTA)，11—外部输入 AREF(PORTB)。

### PRESCALER - ADC Clock Prescaler register

Bit	7	6	5	4	3	2	1	0	
+0x04	-	-	-	-	-	PRESCALER[2:0]			PRESCALER
Read/Write	R	R	R	R	R	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

时钟分频寄存器，第 2 至第 0 位，用于选择时钟的分频，详情请参阅数据手册第 25 章。

### CALL - ADC Calibration value registers

The CALL and CALH register pair hold the 12-bit value ADC calibration value CAL. The ADC is calibrated during production programming, the calibration value must be read from the signature row and written to the CAL register from software.

Bit	7	6	5	4	3	2	1	0	
+0x0C	CAL[7:0]								CAL
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

### CALH - ADC Calibration value registers

Bit	7	6	5	4	3	2	1	0	
+0x0D	CAL[11:8]								CAL
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

ADC 校准值寄存器，通过 AVRStudio 读取校准值，然后把值写入寄存器。

### CTRL - ADC Channel Control Register

Bit	7	6	5	4	3	2	1	0	
+0x00	START	-	-	GAIN[2:0]			INPUTMODE[1:0]		CTRL
Read/Write	R/W	R	R	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

ADC 通道控制寄存器，第 7 位置 1，则开启单次转换；第 1 至第 0 位，选择通道输入模式，在结果无符号模式下，00—内部正输入，01—单端输入，在结果有符号模式下，00—内部正输入，01—单端输入，10—差分输入，11—差分增益。详情请参阅数据手册第 25 章。

## Mcuzone Application Notes

### MUXCTRL - ADC Channel MUX Control registers

The MUX register defines the input source for the channel.

Bit	7	6	5	4	3	2	1	0	
+0x01	-	MUXPOS[3:0]				-	MUXNEG[1:0]		MUXCTRL
Read/Write	R	R/W	R/W	R/W	R/W	R	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

选择器寄存器，第 6 至第 3 位，正输入通道选择，当 INPUTMODE 在 01, 10, 11 三种模式下，000—PIN0, 001—PIN1, 010—PIN2, 以此类推；第 1 至第 0 位，负输入通道选择，当 INPUTMODE 在 10 模式下，00—PIN0, 01—PIN1, 10—PIN2, 11—PIN3, 当 INPUTMODE 在 11 模式下，00—PIN4, 01—PIN5, 10—PIN6, 11—PIN7。

### INTFLAG - ADC Channel Interrupt Flag registers

Bit	7	6	5	4	3	2	1	0	
+0x03	-	-	-	-	-	-	-	IF	INTFLAG
Read/Write	R	R	R	R	R	R	R	R/W	
Initial Value	0	0	0	0	0	0	0	0	

ADC 通道中断标志寄存器，产生中断后，IF 会置位，通过写 1 来清标志位。

### INTCTRL - ADC Channel Interrupt Control registers

Bit	7	6	5	4	3	2	1	0	
+0x02	-	-	-	-	INTMODE[1:0]		INTLVL[1:0]		INTCTRL
Read/Write	R	R	R	R	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

ADC 通道中断控制寄存器，第 3 至第 2 位，选择中断模式；第 1 至第 0 位，设计中断级并使能中断。

### RESH - ADC Channel n Result register High

Bit	7	6	5	4	3	2	1	0	
12-bit, left.	RES[11:4]								
12-bit, right	-	-	-	-	RES[11:8]				
8-bit	-	-	-	-	-	-	-	-	
Read/Write	R	R	R	R	R	R	R	R	R
Initial Value	0	0	0	0	0	0	0	0	0

### RESL - ADC Channel n Result register Low

Bit	7	6	5	4	3	2	1	0	
12-/8-	RES[7:0]								
12-bit, left.	RES[3:0]				-	-	-	-	
Read/Write	R	R	R	R	R	R	R	R	R
Initial Value	0	0	0	0	0	0	0	0	0

结果寄存器，用于存放转换结果。

还有一些未用到的寄存器，如下：EVCTRL、TEMP、CMPH、CMPL 等，详情请参阅数据手册第 25 章。

**Mcuzone Application Notes**

**25.18 Register Summary - ADC**

This is the register summary when the ADC is configured to give standard 12-bit results. The register summary for 8-bit and 12-bit left adjusted will be similar, but with some changes in the result registers CHnRESH and CHnRESL.

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Page	
+0x00	CTRLA	DMASEL[1:0]		CH[3:0]START			FLUSH	ENABLE		302	
+0x01	CTRLB	-	-	-	CONVMODE	FREERUN	RESOLUTION[1:0]		-	302	
+0x02	REFCTRL	-	-	REFSEL[1:0]		-	-	BANDGAP	TEMPREF	303	
+0x03	EVCTRL	SWEEP[1:0]		EVSEL[2:0]		EVACT[2:0]				304	
+0x04	PRESCALER	-	-	-	-	-	PRESCALER[2:0]			305	
+0x05	Reserved	-	-	-	-	-	-	-	-		
+0x06	INTFLAGS	-	-	-	-	CH[3:0]IF				306	
+0x07	TEMP	TEMP[7:0]									306
+0x08	Reserved	-	-	-	-	-	-	-	-		
+0x09	Reserved	-	-	-	-	-	-	-	-		
+0x0A	Reserved	-	-	-	-	-	-	-	-		
+0x0B	Reserved	-	-	-	-	-	-	-	-		
+0x0C	CALL	CALL[7:0]									307
+0x0D	CALH	-	-	-	-	CAL[11:8]					
+0x0E	Reserved	-	-	-	-	-	-	-	-		
+0x0F	Reserved	-	-	-	-	-	-	-	-		
+0x10	CH0RESL	CH0RES[7:0]									308
+0x11	CH0RESH	CH0RES[15:8]									307
+0x12	CH1RESL	CH1RES[7:0]									308
+0x13	CH1RESH	CH1RES[15:8]									307
+0x14	CH2RESL	CH2RES[7:0]									308
+0x15	CH2RESH	CH2RES[15:8]									307
+0x16	CH3RESL	CH3RES[7:0]									308
+0x17	CH3RESH	CH3RES[15:8]									307
+0x18	CMPL	CMP[7:0]									308
+0x19	CMPH	CMP[15:8]									308
+0x1A	Reserved	-	-	-	-	-	-	-	-		
+0x1B	Reserved	-	-	-	-	-	-	-	-		
+0x1C	Reserved	-	-	-	-	-	-	-	-		
+0x1D	Reserved	-	-	-	-	-	-	-	-		
+0x1E	Reserved	-	-	-	-	-	-	-	-		
+0x1F	Reserved	-	-	-	-	-	-	-	-		
+0x20	CH0 Offset										
+0x28	CH1 Offset										
+0x30	CH2 Offset										
+0x38	CH3 Offset										

**25.19 Register Summary - ADC Channel**

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Page	
+0x00	CTRL	START	-	-	GAIN[2:0]			INPUTMODE[1:0]		309	
+0x01	MUXCTRL	-	MUXPOS[3:0]			-	MUXNEG[1:0]			310	
+0x02	INTCTRL	-	-	-	-	INTMODE[1:0]		INTLVL[1:0]		312	
+0x03	INTFLAG	-	-	-	-	-	-	-	IF	312	
+0x04	RESL	RES[7:0]									313
+0x05	RESH	RES[15:8]									312
+0x06	Reserved	-	-	-	-	-	-	-	-		
+0x07	Reserved	-	-	-	-	-	-	-	-		

### 3. 演示代码

示例代码:

此代码实现单次转换，并把转换的结果不做任何处理发送到电脑。

```
void Usart_Init(void); //USART 寄存器设置
```

```
void Usart_PutChar(unsigned char cTXData); //字节发送函数
```

```
void adc_init(void)
{
    ADCA_CAL = 0x0444;           //校准值
    ADCA_PRESCALER = 0x03;      //时钟分频

    ADCA_REFCTRL = 0x10;        //基准源
    ADCA_CTRLB = 0x10;          //转换模式-有符号
    ADCA_CH0_CTRL = 0x01;       //输入模式
    // ADCA_CH0_MUXCTRL = 0x0a; //通道选择

    ADCA_CTRLA = 0x01;          //ADC 使能
    ADCA_CH0_INTCTRL = 0x01;    //中断使能
}

int main(void)
{
    unsigned int i,j;
    Usart_Init();
    PORTC_DIR |= 0x08;
    adc_init();
    ADCA_CH0_CTRL |= 0x80;
    while(1)
    {
        while(!ADCA_CH0_INTFLAGS);
        ADCA_CH0_INTFLAGS = 0x01;
        i = ADCA_CH0_RES & 0xff00;
        i = i / 0x100;
        j = ADCA_CH0_RES & 0x00ff;
        Usart_PutChar(i);
        Usart_PutChar(j);
    }
}

void Usart_Init(void)
{
    USARTC0_CTRLA = 0x14;        //设置中断级，并使能中断
```

Mcuzone Application Notes

```
    USARTC0_CTRLB |= 0x18;           //使能
    USARTC0_BAUDCTRLA = 0x0c;       //设置波特率
}

void Usart_PutChar(unsigned char cTXData)
{
    while(!(USARTC0_STATUS & 0x20)); //只有数据寄存器为空时才能发送数据
    USARTC0_DATA = cTXData;
}
```

www.mcuzone.com