

ATxmega 演示代码之 DAC

——基于 ATxmega64A3-EK 和 JTAGICE mkII-CN

文档编号	MAN0012A_CH				
文档版本	Rev. A				
文档摘要	详细讲述了 XMEGA64A3 的 DAC 使用方法				
关键词	AVR、XMEGA、ATxmega64A3-EK、USB JTAGICE mkII-CN、DAC				
创建日期	2010-08-09	创建人员	Kiler	审核人员	Hotislandn
文档类型	公开发布/仿真器配套文件				
版权信息	Mcuzone 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	2010-08-09	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 www.atarm.com

1.概述

ATxmega 是 ATMEEL 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 DAC 操作，利用 DAC 产生音频信号。
ATxmega64A3 具备两个通道的 12bit DAC。

2. 寄存器概述

CTRLA – DAC Control Register A

Bit	7	6	5	4	3	2	1	0	
+0x00	-	-	-	IDOEN	CH1EN	CH0EN	.	ENABLE	CTRLA
Read/Write	R	R	R	R/W	R	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

控制寄存器 A，第 4 位是 DAC 中断输出使能，第 3 位和第 2 位是通道输出使能位，第 0 位则是 DAC 使能位。

CTRLB – DAC Control Register B

Bit	7	6	5	4	3	2	1	0	
+0x01	-	CHSEL[1:0]		-	-	-	CH1TRIG	CH0TRIG	CTRLB
Read/Write	R	R/W	R	R	R	R	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

控制寄存器 B，第 6 位和第 5 位是通道选择位，00—通道 0，10—通道 0 和 1。详情请见数据手册第 26 章。

Mcuzone Application Notes

CTRLC – DAC Control Register C

Bit	7	6	5	4	3	2	1	0	
+0x02	-	-	-	REFSEL[1:0]		-	-	LEFTADJ	CTRLC
Read/Write	R	R	R	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 C，第 4 位和第 3 位用于选择基准源，00—内部 1V，01—AVcc，10—外部基准 PORTA，11—外部基准 PORTB。

第 0 位用于选择数据的对齐模式，0—右对齐，1—左对齐。

STATUS – DAC Status Register

Bit	7	6	5	4	3	2	1	0	
+0x05	-	-	-	-	-	-	CH1DRE	CH0DRE	STATUS
Read/Write	R	R	R	R	R	R	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

状态寄存器，第 1 位是通道 1 数据寄存器空标志，第 0 位是通道 0 数据寄存器空标志。

CH0DATAH – DAC Channel 0 Data Register High

Bit	7	6	5	4	3	2	1	0
Right-adjust	-	-	-	-	CHDATA[11:8]			
Left-adjust	CHDATA[11:4]							
Right-adjust	Read/Write	R	R	R	R	R/W	R/W	R/W
Left-adjust	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Right-adjust	Initial Value	0	0	0	0	0	0	0
Left-adjust	Initial Value	0	0	0	0	0	0	0

CH0DATAH – DAC Channel 0 Data Register Low

Bit	7	6	5	4	3	2	1	0	
Right-adjust	CHDATA[7:0]								
Left-adjust	CHDATA[3:0]				-	-	-	-	
Right-adjust	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Left-adjust	Read/Write	R/W	R/W	R/W	R/W	R	R	R	R
Right-adjust	Initial Value	0	0	0	0	0	0	0	0
Left-adjust	Initial Value	0	0	0	0	0	0	0	0

通道 0 数据寄存器，用于存放 DAC 的数值。

CH1DATAH – DAC Channel 1 Data Register High

Bit	7	6	5	4	3	2	1	0	
Right-adjust	-	-	-	-	CHDATA[11:8]				
Left-adjust	CHDATA[11:4]								
Right-adjust	Read/Write	R	R	R	R	R/W	R/W	R/W	R/W
Left-adjust	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Right-adjust	Initial Value	0	0	0	0	0	0	0	0
Left-adjust	Initial Value	0	0	0	0	0	0	0	0

Mcuzone Application Notes

CH1DATAL – DAC Channel 1 Data Register Low byte

Bit	7	6	5	4	3	2	1	0
Right-adjust	CHDATA[7:0]							
Left-adjust	CHDATA[3:0]				-	-	-	-
Right-adjust	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Left-adjust	Read/Write	R/W	R/W	R/W	R/W	R	R	R
Right-adjust	Initial Value	0	0	0	0	0	0	0
Left-adjust	Initial Value	0	0	0	0	0	0	0

通道 1 数据寄存器，用于存放 DAC 的数值。

还有一些未用到的寄存器，如下：

EVCTRL、TIMCTRL、GAINCAL、OFFSETCAL，详情请参阅数据手册第 26 章。

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Page
+0x00	CTRLA	-	-	-	-	CH1EN	CH0EN	LPMODE	ENABLE	320
+0x01	CTRLB	-	-	CHSEL[1:0]	-	-	-	CH1TRIG	CH0TRIG	320
+0x02	CTRLC	-	-	-	-	REFSEL[1:0]	-	-	LEFTADJ	321
+0x03	EVCTRL	-	-	-	-	-	-	EVSEL[2:0]	-	322
+0x04	TIMCTRL	-	-	CONINTVAL[2:0]	-	-	-	REFRESH[3:0]	-	322
+0x05	STATUS	-	-	-	-	-	-	CH1DRE	CH0DRE	324
+0x06	Reserved	-	-	-	-	-	-	-	-	
+0x07	Reserved	-	-	-	-	-	-	-	-	
+0x08	GAINCAL	-	-	-	-	-	-	-	-	326
+0x09	OFFSETCAL	-	-	-	-	-	-	-	-	327
+0x10	Reserved	-	-	-	-	-	-	-	-	
+0x11	Reserved	-	-	-	-	-	-	-	-	
+0x12	Reserved	-	-	-	-	-	-	-	-	
+0x13	Reserved	-	-	-	-	-	-	-	-	
+0x14	Reserved	-	-	-	-	-	-	-	-	
+0x15	Reserved	-	-	-	-	-	-	-	-	
+0x16	Reserved	-	-	-	-	-	-	-	-	
+0x17	Reserved	-	-	-	-	-	-	-	-	
+0x18	CH0DATAL	-	-	-	-	-	-	-	-	325
+0x19	CH0DATAH	-	-	-	-	-	-	-	-	324
+0x1A	CH1DATAL	-	-	-	-	-	-	-	-	326
+0x1B	CH1DATAH	-	-	-	-	-	-	-	-	325

3. 演示代码

C 调音符	1	1 [#]	2	2 [#]	3	4	4 [#]	5	5 [#]	6	6 [#]	7
频率 Hz	262	277	293	311	329	349	370	392	415	440	466	494

示例代码：

此代码实现音调输出的功能，按下 S1 就发出“哆”的声音，S2 是“咪”的声音，S3 是“咪”.....

```
void dac_init(void)
```

```
{
    PORTB_DIR |= 0x0c;
    DACB_CTRLA |= 0x1d;           //使能
    DACB_CTRLB |= 0x40;           //通道选择
    //DACB_CTRLC |= 0x08;         //AVCC 基准源，左对齐
    DACB_GAINCAL = 0x12;
    DACB_OFFSETCAL = 0x2d;
}
```

```
void delay(unsigned char music)
{
    unsigned int time;
    switch(music)
    {
        case 1:for(time=82;time>0;time--);break;           //262Hz,3916/2=1908us
        case 2:for(time=73;time>0;time--);break;           //293Hz,3413/2=1706us
        case 3:for(time=65;time>0;time--);break;           //329Hz,3039/2=1519us
        case 4:for(time=62;time>0;time--);break;           //349Hz,2865/2=1432us
        case 5:for(time=53;time>0;time--);break;           //392Hz,2551/2=1275us
        case 6:for(time=47;time>0;time--);break;           //440Hz,2272/2=1136us
        case 7:for(time=41;time>0;time--);break;           //494Hz,2024/2=1012us
        case 8:for(time=400;time>0;time--);break;           //10ms
    }
}
```

```
void io_init(void)
{
    PORTC_DIR &= 0xcf;           //按键方向
    PORTD_DIR &= 0xfc;
    PORTC_PIN4CTRL |= 0x18;
    PORTC_PIN5CTRL |= 0x18;
    PORTD_PIN0CTRL |= 0x18;
    PORTD_PIN1CTRL |= 0x18;
}
```

```
unsigned char scankey(void)
{
    unsigned char i1,j1,i2,j2,z;
    i1 = PORTC_IN & 0x30;
    j1 = PORTD_IN & 0x03;
    if(i1 == 0x30 && j1 == 0x03) return 0;
    delay(8);
    i2 = PORTC_IN & 0x30;
    j2 = PORTD_IN & 0x03;
    if(i1 != i2 || j1 != j2) return 0;           //对比数据
    if(i2 == 0x20) z = 1;
    if(i2 == 0x10) z = 2;
    if(j2 == 0x02) z = 3;
    if(j2 == 0x01) z = 4;
    return z;
}
```

Mcuzone Application Notes

```
int main(void)
{
    unsigned int i,j;
    io_init();
    dac_init();
    while(1)
    {
        j = scankey();
        if(j!=0)
            for(i=0;i<700;i++)
            {
                DACB_CH0DATA=0x09ff;
                delay(j);
                DACB_CH0DATA=0x0000;
                delay(j);
            }
    }
}
```