ATxmega 演示代码之 IO 口

——基于 ATxmega64A3-EK 和 JTAGICE mkII-CN

文档编号	MAN0006A_CH				
文档版本	Rev. A				
文档摘要	详细讲述了 XMEGA64A3 的 IO 口相关寄存器及 IO 操作演示				
关键词	AVR、XMEGA、IO、ATxmega64A3-EK、JTAGICE mkII-CN				
创建日期	2010-05-17	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	2010-05-17	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 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。

本文演示代码为 64A3 的 IO 操作。
XMEGA64A3 的 IO 口是标准的 IO 口，内部都有弱上拉。

2. 寄存器概述

DIR - Data Direction Register

Bit	7	6	5	4	3	2	1	0	
+0x00	DIR[7:0]								DIR
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	

数据方向寄存器，高电平为输出模式，低电平为输入模式。

OUT - Data Output Value

Bit	7	6	5	4	3	2	1	0	
+0x04	OUT[7:0]								OUT
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	

输出数据寄存器，控制引脚输出的电平。

IN - Data Input Value Register

Bit	7	6	5	4	3	2	1	0	
+0x08	IN[7:0]								IN
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	

输入数据寄存器，用于读取引脚当前的电平。

PINnCTRL - Pin n Configuration Register

Bit	7	6	5	4	3	2	1	0	
	SRLEN	INVEN	OPC[2:0]			ISC[2:0]			PINnCTRL
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	

引脚控制寄存器，第 3 位到第 5 位可以使能内部弱上拉和下拉，设置为 011 则上拉，010 则下拉。

还有一些未用的寄存器，如下：DIRSET、DIRCLR、DIRTGL、OUTSET、OUTCLR、OUTTGL、INTCTRL、INT0MASK、INT1MASK、INTFLAS，详情请参阅数据手册第 13 章。

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Page
+0x00	DIR	DIR[7:0]								138
+0x01	DIRSET	DIRSET[7:0]								138
+0x02	DIRCLR	DIRCLR[7:0]								139
+0x03	DIRTGL	DIRTGL[7:0]								139
+0x04	OUT	OUT[7:0]								139
+0x05	OUTSET	OUTSET[7:0]								139
+0x06	OUTCLR	OUTCLR[7:0]								140
+0x07	OUTTGL	OUTTGL[7:0]								140
+0x08	IN	IN[7:0]								140
+0x09	INTCTRL	-	-	-	-	INT1LVL[1:0]		INT0LVL[1:0]		141
+0x0A	INT0MASK	INT0MSK[7:0]								141
+0x0B	INT1MASK	INT1MSK[7:0]								141
+0x0C	INTFLAGS	-	-	-	-	-	-	INT1IF	INT0IF	142
+0x0D	Reserved	-	-	-	-	-	-	-	-	
+0x0E	Reserved	-	-	-	-	-	-	-	-	
+0x0F	Reserved	-	-	-	-	-	-	-	-	
+0x10	PIN0CTRL	SRLEN	INVEN	OPC[2:0]			ISC[2:0]			142
+0x11	PIN1CTRL	SRLEN	INVEN	OPC[2:0]			ISC[2:0]			142
+0x12	PIN2CTRL	SRLEN	INVEN	OPC[2:0]			ISC[2:0]			142
+0x13	PIN3CTRL	SRLEN	INVEN	OPC[2:0]			ISC[2:0]			142
+0x14	PIN4CTRL	SRLEN	INVEN	OPC[2:0]			ISC[2:0]			142
+0x15	PIN5CTRL	SRLEN	INVEN	OPC[2:0]			ISC[2:0]			142
+0x16	PIN6CTRL	SRLEN	INVEN	OPC[2:0]			ISC[2:0]			142
+0x17	PIN7CTRL	SRLEN	INVEN	OPC[2:0]			ISC[2:0]			142
+0x18	Reserved	-	-	-	-	-	-	-	-	
+0x19	Reserved	-	-	-	-	-	-	-	-	
+0x1A	Reserved	-	-	-	-	-	-	-	-	
+0x1B	Reserved	-	-	-	-	-	-	-	-	
+0x1C	Reserved	-	-	-	-	-	-	-	-	
+0x1D	Reserved	-	-	-	-	-	-	-	-	
+0x1E	Reserved	-	-	-	-	-	-	-	-	
+0x1F	Reserved	-	-	-	-	-	-	-	-	

3. 演示代码

此代码实现按键点亮 LED 功能。

```
    int main(void)
{
     unsigned char j;
     io_init();
     while(1)
     {
          j = scankey();
          switch(j)
          {
               case 1:PORTE.OUT |= 0x01;break;              //D1
               case 2:PORTE.OUT |= 0x02;break;              //D2
               case 3:PORTF.OUT |= 0x10;break;              //D4
               case 4:PORTF.OUT |= 0x20;break;              //D6
          }
     }
}

void io_init(void)
{
    PORTC_DIR &= 0xcf;              //按键方向
    PORTD_DIR &= 0xfc;
    PORTC_PIN4CTRL |= 0x18;       //开启弱上拉
    PORTC_PIN5CTRL |= 0x18;
    PORTD_PIN0CTRL |= 0x18;
    PORTD_PIN1CTRL |= 0x18;
    PORTE_DIR |= 0x03;              //LED 方向
    PORTF_DIR |= 0x30;
}

void Delayus(unsigned int lus)
{
     while(lus--)
     {
          _delay_loop_2(1);

     }
}

void Delayms(unsigned int lms)
{
```

```
    while(lms--)
    {
        Delayus(1000);
    }
}

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;
    Delayms(5);
    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;
    do{
        i2 = PORTC_IN & 0x30;
        j2 = PORTD_IN & 0x03;
      }while(i2 != 0x30 || j2 != 0x03);                    //判断是否释放
    return    z;
}
```