

```
/*
 * 文件名: ad_test.c
 * 功 能: 控制ad7606采集电压信 A
 * 说 明: 当ad7606转换结束, 触发中断, 对数据进行读 ;
 */

#include <stdio.h>
#include <string.h>
#include "system.h" //包含基本的硬件描述信息
#include "altera_avalon_timer_regs.h"
//定义内核寄存器的映射, 提供对底层硬件的符号化访问
#include "altera_avalon_pio_regs.h" //包含基本的I/O口信息
#include "alt_types.h" //Altera定义的数据类型
#include "sys/alt_irq.h"
#include "sys/alt_alarm.h" //系统时钟服务头文件
#include "priv/alt_busy_sleep.h" //延时函数alt_busy_sleep ()

//#define ad_busy *(volatile unsigned char *) AD_BUSY_BASE
alt_u8 led_state = 0;
alt_u16 ad_code[8];

/*
 * 名 称: uart_SendByte
 * 功 能: 向串口发送字 节
 */
void uart_SendByte (alt_u8 ucDat)
{
    while(! (IORD_ALTERA_AVALON_UART_STATUS (UART_BASE)&0x0040));
    //等待数据发送完毕
    IOWR_ALTERA_AVALON_UART_TXDATA (UART_BASE,ucDat);
}

/*
 * 名称: uart_SendStr
 * 功能: 向串口发送字符 ®
 */
void uart_SendStr (alt_u8 const *puiStr)
{
    while (1) {
        if (*puiStr == '\0') {
            break; /* 遇到结束符, 退出 */
        }
        uart_SendByte (*puiStr++);
    }
}

/*
 * 名 称: ad_data()
 * 功 能: 将读取的数字量转换成相应的电压 u
 */
float ad_data(alt_u16 code)
{
    float ad_data;
    if(code<=0x7FFF)
        ad_data=code*0.00030517578125;
}
```

```
        else if (code>=0x8000)

            ad_data=(code*0.00030517578125)-20;

        return ad_data;
    }
/*****

* 名    称: ad7606_irq_Init()

* 功    能: 初始化ad7606中断, 开中断, 清边沿捕获寄存 ÷

* 入口参数:  无

* 出口参数:  无

*****/
void ad7606_irq_Init()
{
    IOWR_ALTERA_AVALON_PIO_IRQ_MASK(AD_BUSY_BASE, 0x01); //开AD_BUSY中断
    IOWR_ALTERA_AVALON_PIO_EDGE_CAP(AD_BUSY_BASE, 0x00); //清中断捕获寄存器
    alt_irq_register(AD_BUSY_IRQ,0,AD7606_ISR);           //注册中断函数
}
/*****

* 名    称: void ad7606_Init()

* 功    能: 初始化ad7606 对ad7606进行复 »

* 入口参数:  无

* 出口参数:  无

*****/
void AD7606_Init()
{
    IOWR_ALTERA_AVALON_PIO_DATA(AD_REST_BASE,1);
    alt_busy_sleep(1);
    IOWR_ALTERA_AVALON_PIO_DATA(AD_REST_BASE,0);
    AD7606_irq_Init();
}

/*****

* 名    称: AD7606_ISR()

* 功    能: AD7606中断服务程 ò

*****/

void AD7606_ISR(void* context, alt_u32 id)
{
    alt_u8 i;
    char  chars[1];
    float voltage[8];

    IOWR_ALTERA_AVALON_PIO_DATA(AD_CONVEST_BASE,0);
    for(i=0;i<8;i++)
    {
```

```
IOWR_ALTERA_AVALON_PIO_DATA(AD_CS_BASE,0);    //使能ad7606
ad_code[i] = IORD_ALTERA_AVALON_PIO_DATA(AD_DATA_BASE);
//读取ad7606转换值
voltage[i] = ad_data(ad_code[i]);              //得到电压值
IOWR_ALTERA_AVALON_PIO_DATA(AD_CS_BASE,1);

}
for(i=0;i<8;i++)
{
    sprintf(chars,"%f",voltage[i]);
    uart_SendStr(chars);                        //通过串口发送转换数据
    uart_SendStr(" ");
}
uart_SendStr("\n");
IOWR_ALTERA_AVALON_PIO_EDGE_CAP(AD_BUSY_BASE, 0x00); //清中断捕获寄存器

IOWR_ALTERA_AVALON_PIO_DATA(LED_BASE,led_state);
led_state=~led_state;
alt_busy_sleep(1000000);
IOWR_ALTERA_AVALON_PIO_DATA(AD_CONVST_BASE,1);    //开启转换
}

/*****

* 名    称: main()

* 功    能: 等待ad7606中 ĩ

*****/
int main()
{
    ad7606_Init();
    while(1);
    return 0;
}
```