

KF8F3132——ADC 转换采样

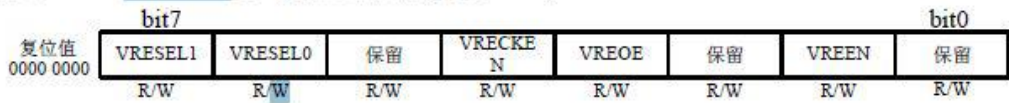
引言

本应用笔记提供了 KF8F3132—ADC 模块相关的配置信息以及如何能够快速的理解并上手使用该模块的一些配置方式。

本应用笔记须与 KF8F3132 数据手册结合使用。

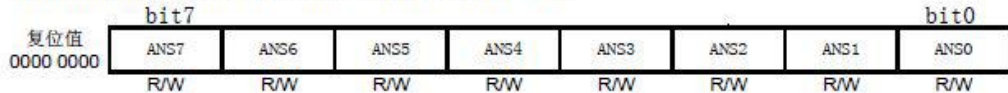
VRECTL (参考电压寄存器)

寄存器7.7: VRECTL: 参考电压寄存器1(地址:2BH)



ANSEL: 模拟/数字口选择寄存器

寄存器ANSEL: 模拟/数字口设置寄存器(地址: 31H)



位操作使用说明:

8 位单片机支持对寄存器的位进行直接的操作，因此在使用的过程中不仅可以通过给寄存器赋值来达到想要的配置，同时还可以直接对位进行操作来达到需要的配置。

以下是对程序中使用到的位进行说明:

- TR0<5:0> P0 口对应引脚方向控制
- TR1<5:0> P1 口对应引脚方向控制
- POLR<5:0> 控制 P0 口对应引脚输出高/低电平
- P1LR<5:0> 控制 P1 口对应引脚输出高/低电平
- ANS<7:0> 控制对应引脚配置为数字/模拟口
- VREEN 内部参考电压使能位
- START ADC 启动使能位

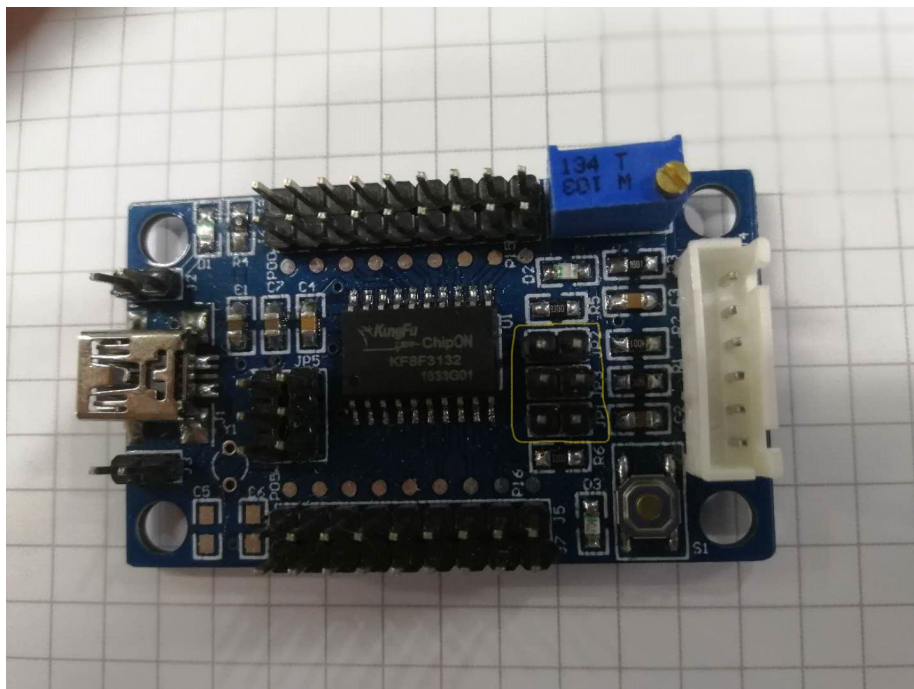
开发环境: chipon IDE

功能简述: 采样 ADC 电位器的输出电压。

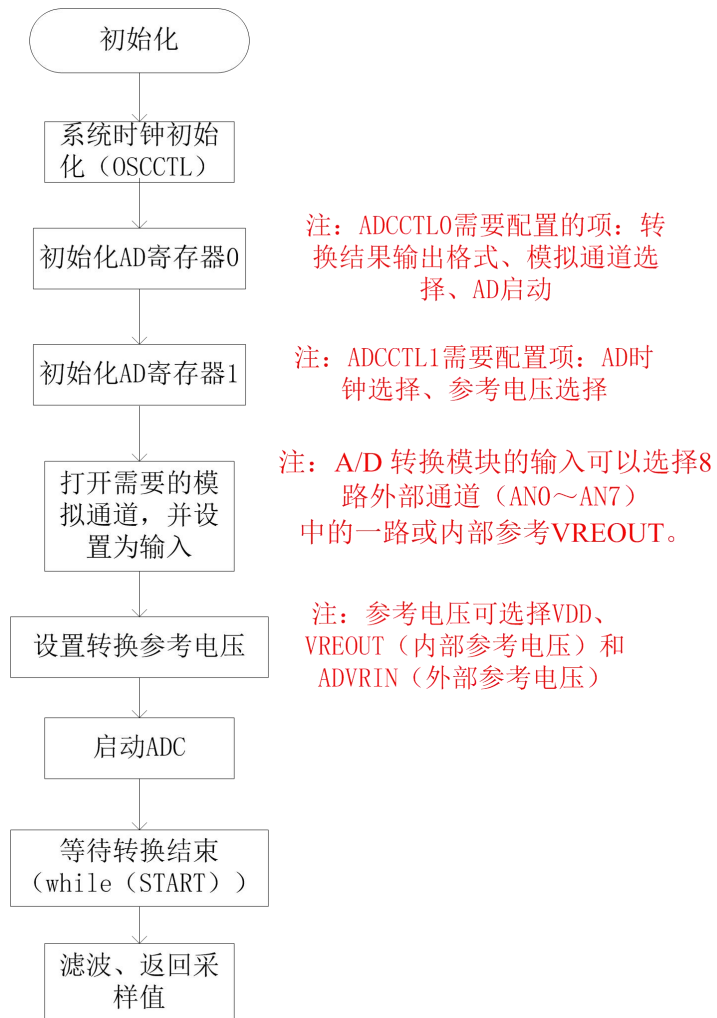
操作方法: 循环切换 ADCCTL1 的配置, 分别用 VDD 和 VREOUT 作为参考电压来采样, 在仿真调试界面查看 ADC 的采样值, 并根据参考电压计算其 ADC 采样电压值, 对比万用表实测的电压值, 观察 ADC 测量的精度和波动情况。将内部 3V 参考电压在 P04 脚输出, 用户可以用万用表测量其精度, 用示波器观察其波动。

实验现象: 电位器电压大于 4V 时, LED2、LED3 熄灭。电位器电压小于 3.9V 且大于 2.5V 时, LED2 熄灭, LED3 点亮。电位器电压小于 2.4V 时, LED2、LED3 点亮。

硬件说明: JP1、JP2、JP3 需带上跳线帽 (图中黄色框中的插针接跳线帽)



ADC 采样测试样例流程图：



样例程序如下：

时钟及 IO 口初始化部分：

```
void Init_fun()
{
    OSCCTL = 0x70; //设置为16M
    /*****端口初始化*****/

    TR0 = 0xFF;//P0口全部设置为输入模式，仿真调试时，需要将对应的IO口设置为输入模式
    TR12 =0;//LED2控制脚设置为输出模式
    LED2=0;
    TR11=0;
    LED3=0;

    /****初始化AD寄存器****/
    ADCCTL0 =0x81;//结果右对齐，禁止T2启动ADC，选择模拟通道0，清零AD转换状态位，使能ADC
    ADCCTL1 =0x45;//ADC时钟设为Fsys/4,ADC转换参考电压选择VDD

// ADCCTL1 =0x4D;//ADC时钟设为Fsys/4,ADC转换参考电压选择VREOUT

    ANS0 =1;//打开模拟通道0
    TR10 =1;//模拟通道0对应的IO口设为输入模式

    /****输出内部参考电压3V****/
    ANS13 =1;//使能模拟通道13
    TR04 =1;//P04 (VREOUT) 设为输入模式
    VRECTL =0x9A;//使能内部3V参考电压，打开AD时钟，内部参考电压输出
}
```

AD 模数转换部分：

```
uint Adc_fun(void)
{
    uint Adc_num = 0 ; //ADC转换缓冲变量
    START = 1; //启动ADC
    while(START); //等待转换结束
    Adc_num = ADCDATA0H; //将高位加进去
    Adc_num <<=8;
    Adc_num += ADCDATA0L; //将低位加进去
    return Adc_num; //返回转换值
}
```

滤波函数（算数平均滤波法）：

```
uint Adc_read(void)
{
    uint Adc_sum = 0 ; //adc采样累加变量
    uchar i = 0;
    for(i = 0; i < 8; i++)
    {
        Adc_sum += Adc_fun(); //累加八次采样值
    }
    Adc_sum >>= 3; //右移动3位 除8求均值
    return Adc_sum;
}
```

主函数:

```
void main()
{
    Dis_adc1 = 0 ;           //采样接收变量
    Dis_adc2 = 0 ;           //采样接收变量
    Init_fun();             //初始化
    while(1)
    {
        ADCCTL1 =0x45;//ADC时钟设为Fsys/4,ADC转换参考电压选择VDD
        Dis_adc1 = Adc_read(); //将ADC采样返回值赋予Dis_adc
        if(Dis_adc1<3195)//小于3.9V, VDD假设为5V
            LED3=0;
        if(Dis_adc1>3208)//大于4V, VDD假设为5V
            LED3=1;

        ADCCTL1 =0x4D;//ADC时钟设为Fsys/4,ADC转换参考电压选择VREOUT
        Dis_adc2 = Adc_read(); //将ADC采样返回值赋予Dis_adc
        if(Dis_adc2<3208)//小于2.4V
            LED2=0;
        if(Dis_adc2>3413)//大于2.5V
            LED2=1;
    }
}
```

模块使用注意事项：

1、初始化函数中所编写的输出内部参考电压 1.7v 为用户自主选择程序，用户可选择查看或不查看内部参考电压，如需查看则需要进行如下配置：

使能模拟通道 3、设置 P04 为输入模式、使能内部参考电压(VREEN 位)、使能内部参考电压输出 (VREOE 位)。

2、在引脚配置中 ，如果某引脚被配置为模拟输入口，将自动禁止有效地数字 I/O、上拉电阻、电平变化中断。

3、程序中未使用任何相关的中断，用户需要使用都中断函数时可以进行下边的配置：

在 MCU 初始化中添加如下配置：

```
{  
  
    ADIF = 0;           //清 0 中断标志位  
  
    PUIE = 0;          //使能外设中断(AD 模块是外设模块)  
  
    ADEN = 1;          //使能 AD 转换中断  
  
    AIE = 1;           //使能总中断  
  
}
```

在中断函数中进行如下配置：

```
Void int_fun0 _interrupt(0)      //设置中断函数  
  
{  
  
    if(ADIF == 1)  
  
    {
```



```

        ADIF = 0;

        Adc_num = ADCDATAH;    //将高位加进去
        Adc_num <<= 8;
        Adc_num += ADCDATAL;    //将低位加进去
    }
}

```

当转换完成时触发中断，程序进入中断函数执行中断，读取 AD 转换的数据，并返回转换函数。

4、当需要 AD 在休眠模式下继续工作时，要把 AD 时钟源设置为 AD 专用的内部振荡器。如果 AD 时钟源不为 AD 专用内部振荡器，执行 IDLE 指令将中断当前的转换操作，并关闭 AD 模块。

5、ADC 输入电压范围设定为 0—5V，ADC 为 12 位，满量程对应 5V，12 位满量程对应的数字值是 2^{12} ，数值 0 对应 0V，转换后数值为 X，X 对应的模拟电压为 Y，则 $2^{12}/5 = X/Y \Rightarrow Y = (5*X)/2^{12}$ 。此公式可用来计算电压的转换值。

6、AD 采样软件滤波方法：限幅滤波法、中位值滤波法、算数平均滤波法、逆推平均滤波法、中位值平均滤波法（防脉冲干扰平均滤波法）、1 阶滞后滤波法、消抖滤波法等。本样例中采用的是算数平均滤波法。

7、ADC 的输入阻抗不得大于 100K。

8、要进行采样时选用的采样电压 VDD 需要保持尽可能的稳定，避免采样值波动较大。

9、为保证 AD 转换的正确进行，所选择的 AD 转换时钟周期 (T_{ad})

典型值应在 1us 左右。