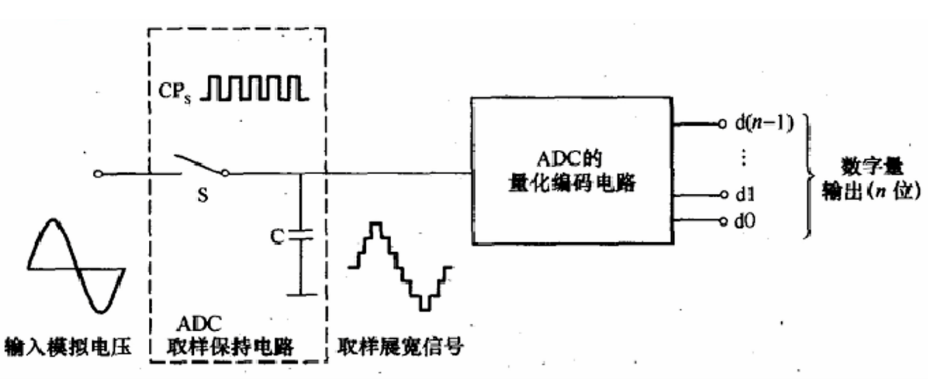
ADC:Analog-to-Digital Converter,模数转换器。它将连续的模拟量转化为离散的数字量，AID转换一般要经过采样、保持、量化和编码4个过程。



最高精度14位，P0端口的引脚可以配置为ADC输入端，依次为AIN0~AIN78路单端输入，4路差分输入，1路片内温度传感器。

两种转换方式:

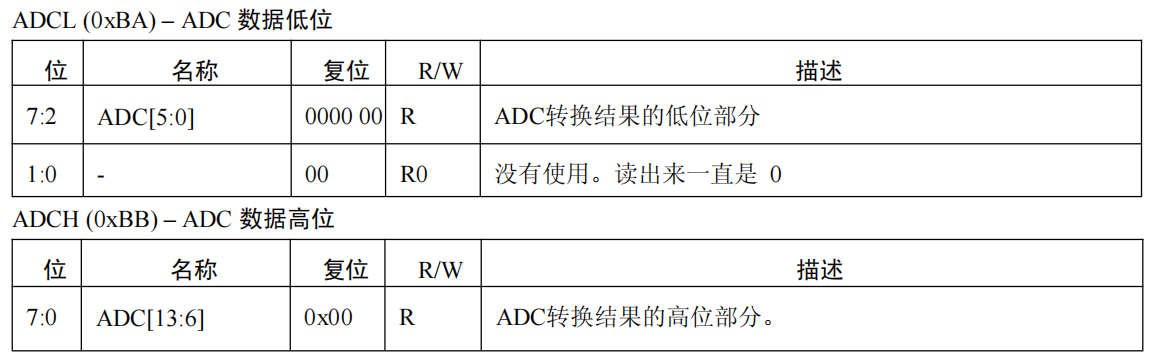
序列转换:多个通道，触发DMA。

单次转换:单个通道，产生一个中断请求，ADCIF标志硬件清除

在单次转换中，通过写ADCCON3寄存器触发指定的通道进行A/D转换。一旦ADCCON3寄存器被写入，A/D转换立即开始。

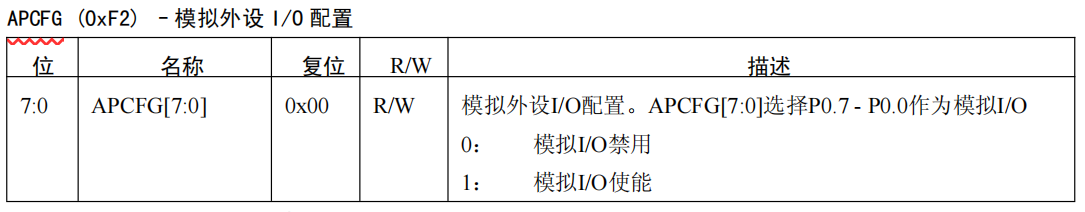
转换结果:转换结果以2的补码形式表示，有符号位，保存到寄存器ADCH和ADCL中。

ADC主要相关寄存器:APCFG寄存器、ADCH寄存器和ADCL寄存器。ADCCON1寄存器、ADCCON2寄存器、ADCCON3寄存器

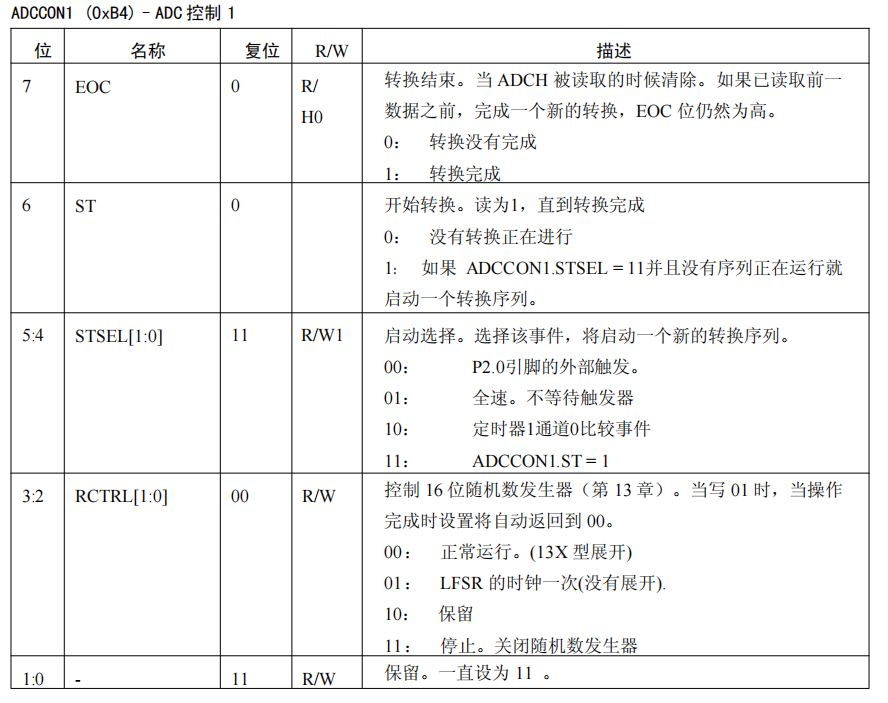


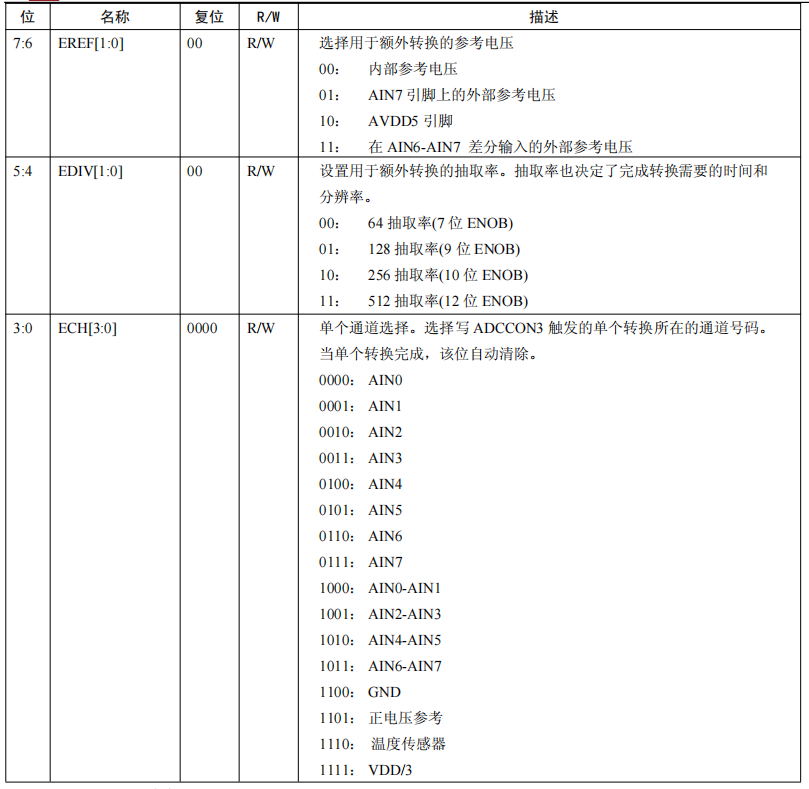
其中ADCH的最高位为符号位用来表示正负 0表示正数和零 1表示负数

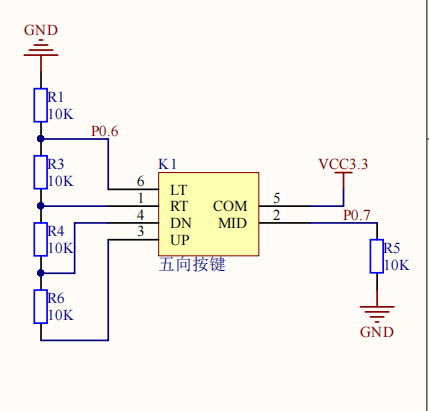
ADCL的低两位没有使用如果读出来的话一直是0



要使用哪个引脚作为输入通道就将指定的位置1即可







Key.c

|  |
| --- |
| #include "key.h"  #include "delay.h"  void Key\_Init(void)  {  //设置P0.1为通用I/O模式  P0SEL &=~ 0x02;  //设置方向为输入方向  P0DIR &=~ 0x02;  //设置输入为上下拉输入  P0INP &=~ 0x02;  //选择下拉输入  P2INP |= 0x20;      //配置P0.7通用IO  P0SEL &=~ (0x80);  //配置P0.7方向为输入  P0DIR &=~ (0x80);  //配置P0.7为上下拉  P0INP &=~ (0x80);  //配置P0.7为下拉输入  P2INP |= (0x20);    //开始P0.6的ADC  APCFG |= (0x40);    }  unsigned int Adc\_Value=0;  unsigned int ADC\_GetValue(void)  {  //以AVDD5为参考电压  //抽取率为512  //使用AIN6通道  ADCCON3 |= (0xB6);  //等待转换完成  while((ADCCON1 & 0x80)!=0x80){}  //读取结果  Adc\_Value = ADCH;  Adc\_Value = Adc\_Value << 8;  Adc\_Value = Adc\_Value | ADCL;  Adc\_Value = Adc\_Value >> 2;  return Adc\_Value;  }  int Key\_Read(void)  {  //P0.1引脚电平是否为高电平（按键是否按下）  if(P0\_1==1)  {  //延时消抖  Delay\_xms(30);  //再次判断  if(P0\_1==1)  {  //等待松手  while(P0\_1==1){}  return 1;  }  }      //P0.7引脚MID  if(P0\_7==1)  {  //延时消抖  Delay\_xms(30);  //再次判断  if(P0\_7==1)  {  //等待松手  while(P0\_7==1){}  return 2;  }  }      //UP  if(2000<ADC\_GetValue() && ADC\_GetValue()<2400)  {  if(2000<ADC\_GetValue() && ADC\_GetValue()<2400)  {  while(2000<ADC\_GetValue() && ADC\_GetValue()<2400){}  return 3;  }  }    //DOWN  if(2700<ADC\_GetValue() && ADC\_GetValue()<3100)  {  Delay\_xms(50);  if(2700<ADC\_GetValue() && ADC\_GetValue()<3100)  {  while(2700<ADC\_GetValue() && ADC\_GetValue()<3100){}  return 4;  }  }      //RIGHT  if(4200<ADC\_GetValue() && ADC\_GetValue()<4600)  {  Delay\_xms(50);  if(4200<ADC\_GetValue() && ADC\_GetValue()<4600)  {  while(4200<ADC\_GetValue() && ADC\_GetValue()<4600){}  return 5;  }  }    //LEFT  if(8000<ADC\_GetValue())  {  Delay\_xms(50);  if(8000<ADC\_GetValue())  {  while(8000<ADC\_GetValue()){}  return 6;  }  }      return 0;  } |

Key.h

|  |
| --- |
| #ifndef \_\_KEY\_H  #define \_\_KEY\_H  #include "iocc2530.h"  void Key\_Init(void);  int Key\_Read(void);  unsigned int ADC\_GetValue(void);  #endif |

Main.c

