[**STM32 ADC学习**](http://www.cnblogs.com/hnrainll/archive/2011/01/18/1937888.html)

1. **什么时候开启AFIO时钟。**
2. **ADC时钟，采样间隔设置。**

12位ADC是一种逐次逼近型模拟数字数字转换器。它有多达18个通道，可测量16个外部和2个内部信号源。

ADC的输入时钟不得超过14MHZ，它是由PCLK2经分频产生。

如果被ADC转换的模拟电压低于低阀值或高于高阀值，AWD模拟看门狗状态位被设置。

**ADC通常要与DMA一起使用 这里只是简单的用库配置ADC 不断扫描来实现ADC的应用。**

**首先配置GPIO与ADC的时钟：**

ADC\_InitTypeDef  ADC\_InitStructure;   
GPIO\_InitTypeDef GPIO\_InitStructure;

RCC\_APB2PeriphClockCmd(RCC\_APB2Periph\_ADC1,ENABLE);   
RCC\_APB2PeriphClockCmd(RCC\_APB2Periph\_GPIOB,ENABLE);

GPIO\_InitStructure.GPIO\_Pin  =GPIO\_Pin\_1;   
GPIO\_InitStructure.GPIO\_Mode =GPIO\_Mode\_AIN;   
GPIO\_Init(GPIOB,&GPIO\_InitStructure); //默认速度为两兆

**配置ADC的运行：**

ADC\_InitStructure.ADC\_Mode              = ADC\_Mode\_Independent;  //独立模式   
ADC\_InitStructure.ADC\_ScanConvMode      =DISABLE;      //连续多通道模式   
ADC\_InitStructure.ADC\_ContinuousConvMode =ENABLE;      //连续转换   
ADC\_InitStructure.ADC\_ExternalTrigConv  = ADC\_ExternalTrigConv\_None; //转换不受外界决定   
ADC\_InitStructure.ADC\_DataAlign         =ADC\_DataAlign\_Right;   //右对齐   
ADC\_InitStructure.ADC\_NbrOfChannel      =1;       //扫描通道数   
ADC\_Init(ADC1,&ADC\_InitStructure);   
ADC\_RegularChannelConfig(ADC1,ADC\_Channel\_9, 1,ADC\_SampleTime\_1Cycles5); //通道X,采样时间为1.5周期,1代表规则通道第1个这个1是啥意思我不太清楚只有是1的时候我的ADC才正常。   
ADC\_Cmd  (ADC1,ENABLE);             //使能或者失能指定的ADC   
ADC\_SoftwareStartConvCmd(ADC1,ENABLE);//使能或者失能指定的ADC的软件转换启动功能

[](http://images.cnblogs.com/cnblogs_com/hnrainll/201101/201101180042507152.gif)   
这里我用的是ADC1的9通道 PB1引脚。

也用一些默认的配置函数同GPIO 的一样例如： ADC\_StructInit

**ADC\_InitTypeDef structureADC\_InitTypeDef定义于文件“stm32f10x\_adc.h”：**

typedef struct

{

u32 ADC\_Mode; FunctionalState ADC\_ScanConvMode; FunctionalStateADC\_ContinuousConvMode; u32 ADC\_ExternalTrigConv; u32ADC\_DataAlign; u8 ADC\_NbrOfChannel;

} ADC\_InitTypeDef

**注意：为了能够正确地配置每一个ADC通道，用户在调用ADC\_Init()之后，必须调用ADC\_ChannelConfig()来配置每个所使用通道的转换次序和采样时间。**

**然后就是不停的读；**

u16 TestAdc(void)   
{   
u16 adc;   
while(ADC\_GetFlagStatus(ADC1, ADC\_FLAG\_EOC)==RESET); //检查制定ADC标志位置1与否 ADC\_FLAG\_EOC 转换结束标志位   
adc=ADC\_GetConversionValue(ADC1);

returnadc;//返回最近一次ADCx规则组的转换结果   
}

**这个程序的8位单片机风格很重，真正的ADC一定要放在DMA或是中断之中。**

**转自：**[**http://ntn314.blog.163.com/blog/static/16174358420103563814664/**](http://ntn314.blog.163.com/blog/static/16174358420103563814664/)