 STM32 ADC 采样 频率的确定

   （4）16.7 可编程的通道采样时间

ADC 使用若干个ADC\_CLK 周期对输入电压采样，采样周期数目可以通过

ADC\_SMPR1 和ADC\_SMPR2 寄存器中的SMP[2:0]位而更改。每个通道可以以

不同的时间采样。

总转换时间如下计算：

TCONV = 采样时间+ 12.5 个周期

例如：

当ADCCLK=14MHz 和1.5 周期的采样时间

TCONV = 1.5 + 12.5 = 14 周期 = 1μs

SMPx[2:0]：选择通道x的采样时间

这些位用于独立地选择每个通道的采样时间。在采样周期中通道选择位必须保持不变。

000：1.5周期 100：41.5周期

001：7.5周期 101：55.5周期

010：13.5周期 110：71.5周期

011：28.5周期 111：239.5周期

注：

– ADC1的模拟输入通道16和通道17在芯片内部分别连到了温度传感器和VREFINT。

– ADC2的模拟输入通道16和通道17在芯片内部连到了VSS。

2.     具体分析如下：

（1）我们的输入信号是50Hz （周期为20ms），初步定为1周期200个采样点，（注：一周期最少采20个点，即采样率最少为1k） ，每2个 采样点间隔为 20ms /200 = 100 us

ADC可编程的通道采样时间 我们选最小的 1.5 周期，则 ADC采样周期一周期大小为

100us /1.5=66us 。 ADC 时钟频率为 1/66us =15 KHz。

  ADC可编程的通道采样时间 我们选71.5 周期，则 ADC采样周期一周期大小为

（100us /71.5） 。 ADC 时钟频率为 7.15MHz。

（2）接下来我们要确定系统时钟：我们 用的是 8M Hz 的外部晶振做时钟源（HSE），估计得 经过 PLL倍频 PLL 倍频系数分别为2的整数倍，最大72 MHz。为了 提高数据 计算效率，我们把系统时钟定为72MHz，(PLL 9倍频)。则 PCLK2=72MHz,PCLK1=36MHz；

我们通过设置时钟配置寄存器(RCC\_CFGR) 中 有 为ADC 时钟提供一个专用的可编程预分器，将PCLK2 8 分频后作为ADC 的时钟，则可知ADC 时钟频率为 9MHz

        从手册可知： ADC 转换时间：

STM32F103xx 增强型产品：ADC 时钟为56MHz 时为1μs(ADC 时钟为72MHz 为1.17μs)

 （3）由以上分析可知：不太对应，我们重新对以上中 内容调整，提出如下两套方案：

方案一：我们的输入信号是50Hz （周期为20ms），初步定为1周期2500个采样点，（注：一周期最少采20个点，即采样率最少为1k） ，每2个 采样点间隔为 20ms /2500 = 8 us

ADC可编程的通道采样时间 我们选71.5 周期，则 ADC采样周期一周期大小为

8us /71.5 。 ADC 时钟频率约为 9 MHz。

将PCLK2 8 分频后作为ADC 的时钟，则可知ADC 时钟频率为 9MHz

方案二：我们的输入信号是50Hz （周期为20ms），初步定为1周期1000个采样点，（注：一周期最少采20个点，即采样率最少为1k） ，每2个 采样点间隔为 20ms /1000= 20 us

ADC可编程的通道采样时间 我们选239.5周期，则 ADC采样周期一周期大小为

20us /239.5 。 ADC 时钟频率约为 12 MHz。

将PCLK2 6 分频后作为ADC 的时钟，则可知ADC 时钟频率为 12MHz

stm32f103最少有2个AD模数转换器，每个ADC都有18个通道，可以测量16个外部和2个内部模拟量。最大转换频率为1Mhz，也就是转换时间为1us（在 ADCCLK = 14Mhz,采样周期为1.5个时钟周期时）。最大时钟超过14Mhz，将导致ADC转换准确度降低。stm32的ADC是12位精度的。

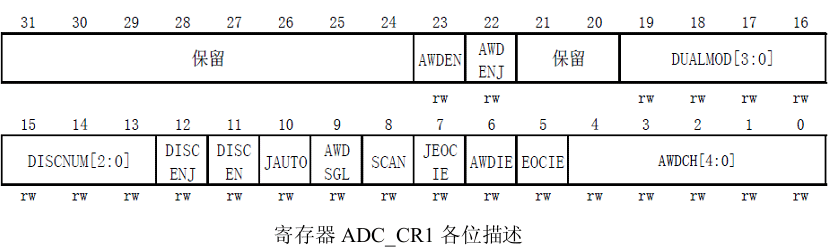
stm32的ADC转换有两种通道，规则通道和注入通道，注入通道可以抢占式地打断规则通道的采样，执行注入通道采样后，再执行之前的规则通道采样，和中断类似。本例只使用规则通道实现独立模式的中断采样，这里不再赘述两种通道区别。

stm32的ADC可以由外部事件触发(例如定时器捕获，EXTI线)和软件触发(即在配置相关寄存器时，直接开启采样）。

STM32的ADC在单次转换模式下，只执行一次转换，该模式可以通过ADC\_CR2 寄存器的ADON 位（只适用于规则通道）启动，也可以通过外部触发启动（适用于规则通道和注入通道），这是CONT 位为0 。 以规则通道为例，一旦所选择的通道转换完成，转换结果将被存在ADC\_DR 寄存器，EOC （转换结束）标志将被置位，如果设置了EOCIE ，则会产生中断。然后ADC将停止，直到下次启动。

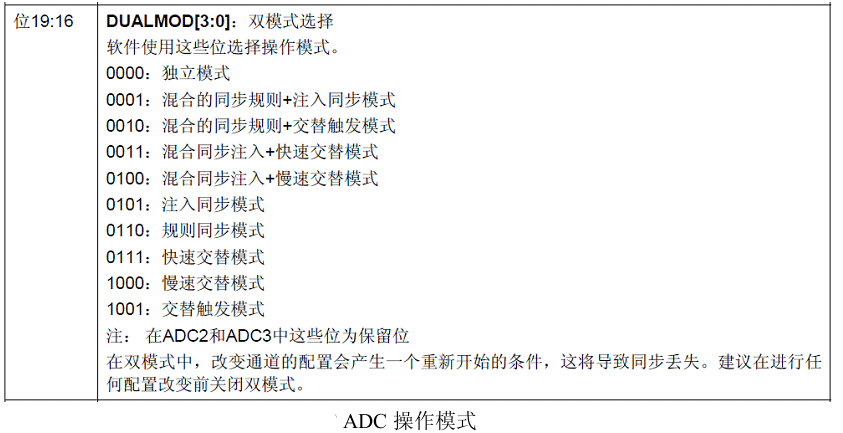
**寄存器简介**

**ADC控制寄存器（ADC\_CR1和ADC\_CR2）**

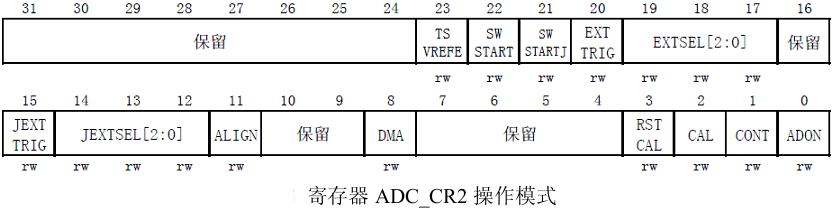
[](http://rmingwang.com/usr/uploads/2012/07/ADC_CR1.gif)

ADC\_CR1的**SCAN 位**，该位用于设置扫描模式，**由软件设置和清除**，**如果设置为1 ，则使用扫描模式，如果为 0，则关闭扫描模式。**在扫描模式下，由 ADC\_SQRx或ADC\_JSQRx寄存器选中的通道被转换。如果设置了 EOCIE 或JEOCIE，只在最后一个通道转换完毕后才会产生EOC 或JEOC 中断。

ADC\_CR1[19:16]用于设置 ADC的操作模式

[](http://rmingwang.com/usr/uploads/2012/07/ADC_DUALMOD.gif)

ADC\_CR2

[](http://rmingwang.com/usr/uploads/2012/07/ADC_CR2.gif)

**ADCON** 位用于**开关AD转换器**。而**CONT 位**用于设置**是否进行连续转换**，我们使用单次转换，所以CONT 位必须为0。**CAL 和RSTCAL**用于**AD校准**。**ALIGN**用于设置数据对齐，我们使用**右对齐，该位设置为0** 。

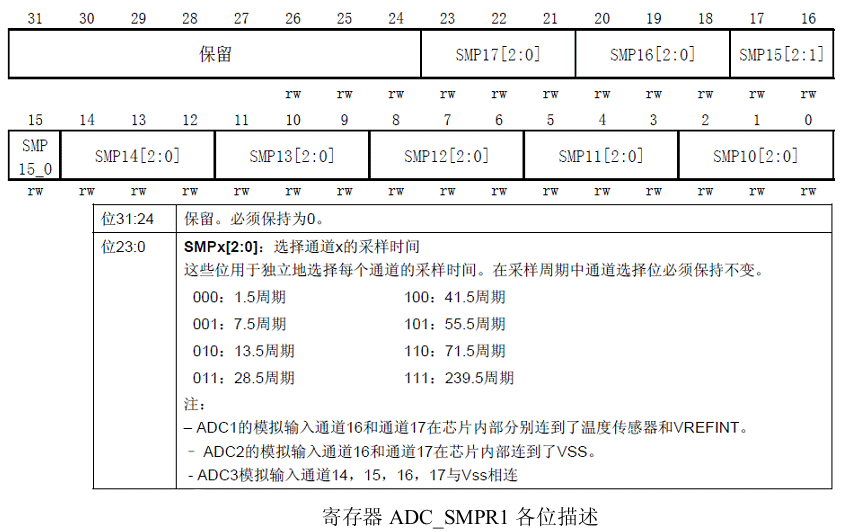
**EXTSEL[2:0]**用于**选择启动规则转换组转换的外部事件**，详细的设置关系如下：

[](http://rmingwang.com/usr/uploads/2012/07/ADC_EXTSEL.gif)

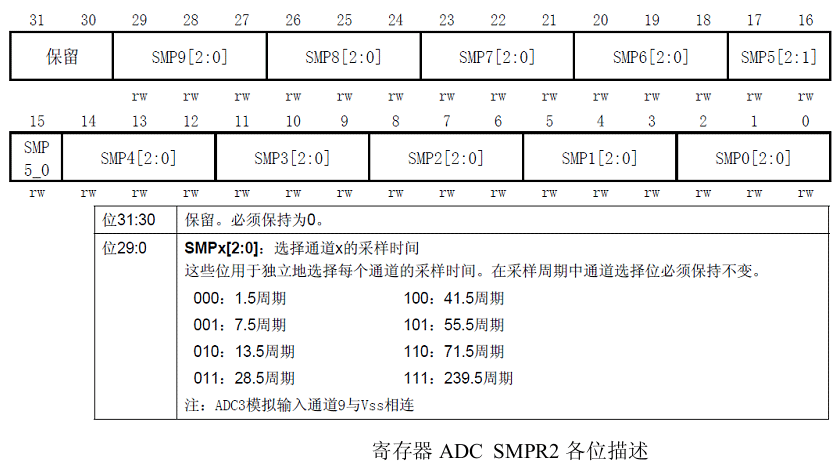
这里使用的是软件触发（SWSTART ），所以设置这3 个位为111 。ADC\_CR2 的SWSTART 位用于开始规则通道的转换，我们每次转换（单次转换模式下）都需要向该位写 1 。AWDEN 为用于使能温度传感器和Vrefint 。

**ADC采样事件寄存器（ADC\_SMPR1 和ADC\_SMPR2 ）**

这两个寄存器用于设置通道0~17的采样时间，每个通道占用 3 个位。

[](http://rmingwang.com/usr/uploads/2012/07/ADC_SMPR1.gif)

ADC\_SMPR2 的各位描述如下

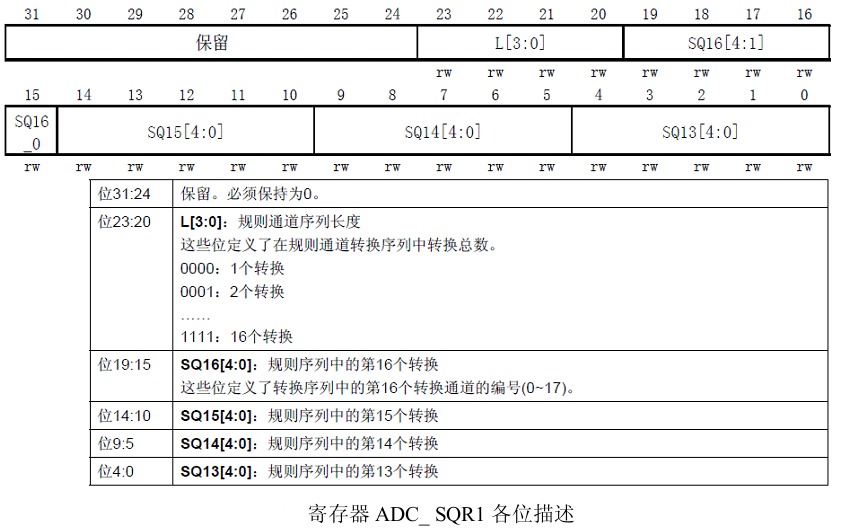
[](http://rmingwang.com/usr/uploads/2012/07/ADC_SMPR2.gif)

对于每个要转换的通道，采样时间建议尽量长一点，以获得较高的准确度，但是这样会降低ADC的转换速率。ADC的转换时间可以由下式计算：

**Tcovn= 采样时间+12.5 个周期**

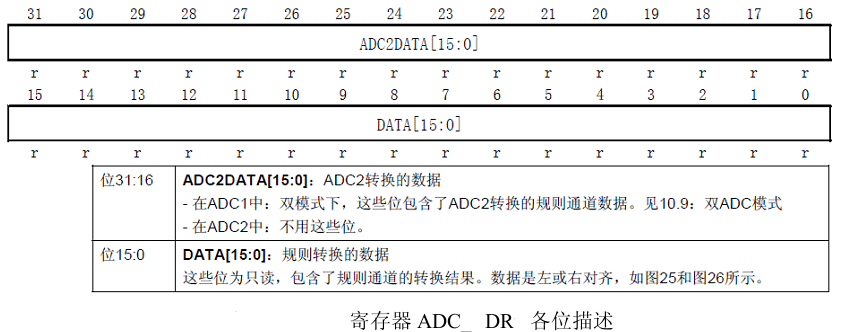
其中：Tcovn 为总转换时间，采样时间是根据每个通道的SMP位的设置来决定的。例如，当ADCCLK=14Mhz 的时候，并设置 1.5个周期的采样时间，则得到:Tcovn=1.5+12.5=14 个周期=1us 。

**ADC规则序列寄存器（ADC\_SQR1~3）**

[](http://rmingwang.com/usr/uploads/2012/07/ADC_SQR1.gif)

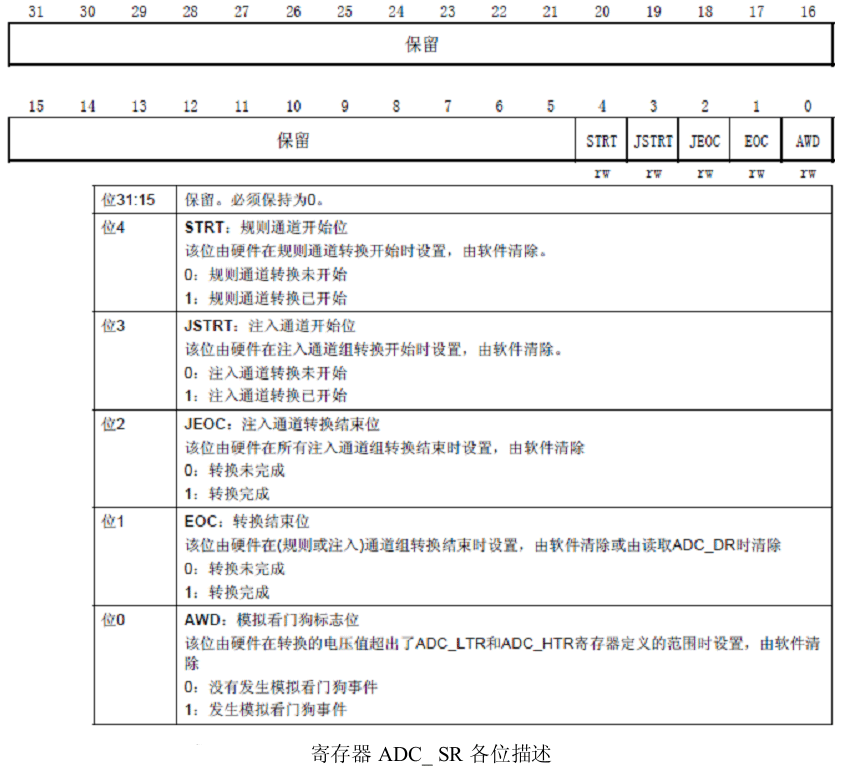
L[3：0] 用于存储规则序列的长度，我们这里只用了 1 个，所以设置这几个位的值为 0 。其他的SQ13~16 则存储了规则序列中第13~16 个通道的编号（0~17）。另外两个规则序列寄存器同ADC\_SQR1大同小异，我们这里就不再介绍了，要说明一点的是：**我们选择的是单次转换，所以只有一个通道在规则序列里面，这个序列就是SQ0 ，通过ADC\_SQR3的最低5 位设置。**

**ADC规则数据寄存器(ADC\_DR)**

[](http://rmingwang.com/usr/uploads/2012/07/ADC_DR.gif)

这里要提醒一点的是，该寄存器的数据可以通过ADC\_CR2 的ALIGN位设置左对齐还是右对齐。在读取数据的时候要注意。

**ADC状态寄存器（ADC\_SR ）**

[](http://rmingwang.com/usr/uploads/2012/07/ADC_SR.gif)

这里我们要用到的是EOC 位，我们通过判断该位来决定是否此次规则通道的AD转换已经完成，如果完成我们就从ADC\_DR 中读取转换结果，否则等待转换完成。

**寄存器操作步骤**

**1 、开启PA口时钟，设置PA0 为模拟输入。**

STM32F103RBT6的ADC通道0 在PA 0 上，所以，我们先要使能 PORTA的时钟，然后设置PA 0 为模拟输入。

**2 、使能ADC1 时钟，并设置分频因子。**

要使用ADC1，第一步就是要使能 ADC1 的时钟，在使能完时钟之后，进行一次 ADC1 的复位。接着我们就可以通过RCC\_CFGR设置ADC1 的分频因子。分频因子要确保 ADC1 的时钟（ADCCLK）不要超过14Mhz 。

**3 、设置ADC1 的工作模式。**

在设置完分频因子之后，我们就可以开始 ADC1 的模式配置了，设置单次转换模式、触发方式选择、数据对齐方式等都在这一步实现。

**4 、设置ADC1 规则序列的相关信息。**

接下来我们要设置规则序列的相关信息，我们这里只有一个通道，并且是单次转换的，所以设置规则序列中通道数为1 ，然后设置通道 0 的采样周期。

**5 、开启AD转换器，并校准。**

在设置完了以上信息后，我们就开启AD转换器，执行复位校准和AD校准，注意这两步是必须的！不校准将导致结果很不准确。

**6 ）读取ADC值。**

在上面的校准完成之后，ADC就算准备好了。接下来我们要做的就是设置规则序列 0 里面的通道，然后启动ADC转换。在转换结束后，读取ADC1\_DR 里面的值就是了。

**硬件设置：我们通过ADC1 的通道0 （PA 0 ）来读取外部电压值。**

注意：这里不能接到板上5V电源上去测试，这可能会烧坏 ADC!

**程序设计**

**ADC采样得到的只是一个相对值，将 转换值/4096\*参考电压 即可得到采样电压 这里的4096是因为stm32的adc为12位精度，表示参考电压时即为 2^12=4096**

MAIN.C

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45 | #include <stm32f10x\_lib.h>  #include "sys.h"  #include "usart.h"  #include "delay.h"  #include "led.h"  #include "key.h"  #include "exti.h"  #include "wdg.h"  #include "timer.h"  #include "lcd.h"  #include "rtc.h"  #include "wkup.h"  #include "adc.h"  //ADC 实验  int main(void)  {      u16 adcx;      float temp;      Stm32\_Clock\_Init(9);//系统时钟设置      delay\_init(72);     //延时初始化      uart\_init(72,9600); //串口1初始化      led\_init();      LCD\_Init();      Adc\_Init();        POINT\_COLOR=RED;//设置字体为红色      LCD\_ShowString(60,110,"ADC TEST");      //显示提示信息      POINT\_COLOR=BLUE;//设置字体为蓝色      LCD\_ShowString(60,130,"ADC\_CH0\_VAL:");      LCD\_ShowString(60,150,"ADC\_CH0\_VOL:0.000V");      while(1)      {          adcx=Get\_Adc(ADC\_CH0);          LCD\_ShowNum(156,130,adcx,4,16);//显示ADC的值          temp=(float)adcx\*(3.3/4096);          adcx=temp;          LCD\_ShowNum(156,150,adcx,1,16);//显示电压值          temp-=adcx;          temp\*=1000;          LCD\_ShowNum(172,150,temp,3,16);          LED0=!LED0;          delay\_ms(250);      }  } |

ADC.C

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56 | #include <stm32f10x\_lib.h>  #include "adc.h"  //ADC 驱动代码  //初始化ADC  //这里我们仅以规则通道为例  //我们默认将开启通道0~3  void  Adc\_Init(void)  {      //先初始化IO口      RCC->APB2ENR|=1<<2;    //使能PORTA口时钟      GPIOA->CRL&=0XFFFF0000;//PA0 1 2 3 anolog输入      //通道10/11设置      RCC->APB2ENR|=1<<9;    //ADC1时钟使能      RCC->APB2RSTR|=1<<9;   //ADC1复位      RCC->APB2RSTR&=~(1<<9);//复位结束      RCC->CFGR&=~(3<<14);   //分频因子清零      //SYSCLK/DIV2=12M ADC时钟设置为12M,ADC最大时钟不能超过14M!      //否则将导致ADC准确度下降!      RCC->CFGR|=2<<14;        ADC1->CR1&=0XF0FFFF;   //工作模式清零      ADC1->CR1|=0<<16;      //独立工作模式      ADC1->CR1&=~(1<<8);    //非扫描模式      ADC1->CR2&=~(1<<1);    //单次转换模式      ADC1->CR2&=~(7<<17);      ADC1->CR2|=7<<17;     //软件控制转换      ADC1->CR2|=1<<20;      //使用用外部触发(SWSTART)!!!   必须使用一个事件来触发      ADC1->CR2&=~(1<<11);   //右对齐      ADC1->SQR1&=~(0XF<<20);      ADC1->SQR1&=0<<20;     //1个转换在规则序列中 也就是只转换规则序列1      //设置通道0~3的采样时间      ADC1->SMPR2&=0XFFFFF000;//通道0,1,2,3采样时间清空      ADC1->SMPR2|=7<<9;      //通道3  239.5周期,提高采样时间可以提高精确度      ADC1->SMPR2|=7<<6;      //通道2  239.5周期,提高采样时间可以提高精确度      ADC1->SMPR2|=7<<3;      //通道1  239.5周期,提高采样时间可以提高精确度      ADC1->SMPR2|=7<<0;      //通道0  239.5周期,提高采样时间可以提高精确度        ADC1->CR2|=1<<0;       //开启AD转换器      ADC1->CR2|=1<<3;        //使能复位校准      while(ADC1->CR2&1<<3);  //等待校准结束      //该位由软件设置并由硬件清除。在校准寄存器被初始化后该位将被清除。      ADC1->CR2|=1<<2;        //开启AD校准      while(ADC1->CR2&1<<2);  //等待校准结束      //该位由软件设置以开始校准，并在校准结束时由硬件清除  }  //获得ADC值  //ch:通道值 0~3  u16 Get\_Adc(u8 ch)  {      //设置转换序列      ADC1->SQR3&=0XFFFFFFE0;//规则序列1 通道ch      ADC1->SQR3|=ch;      ADC1->CR2|=1<<22;       //启动规则转换通道      while(!(ADC1->SR&1<<1));//等待转换结束      return ADC1->DR;     //返回adc值  } |

ADC.H

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12 | #ifndef \_\_ADC\_H  #define \_\_ADC\_H    #define ADC\_CH0  0 //通道0  #define ADC\_CH1  1 //通道1  #define ADC\_CH2  2 //通道2  #define ADC\_CH3  3 //通道3    void Adc\_Init(void);  u16  Get\_Adc(u8 ch);    #endif |

## [转载]STM32的ADC的采样时间及模拟信号的最大带宽

 (2011-08-08 16:48:56)

[[http://simg.sinajs.cn/blog7style/images/common/sg_trans.gif](javascript:;)转载▼](javascript:;)

|  |  |
| --- | --- |
| 标签：  [转载](http://search.sina.com.cn/?c=blog&q=%D7%AA%D4%D8&by=tag) | 分类： [MCU](http://blog.sina.com.cn/s/articlelist_2101947072_1_1.html) |

**原文地址：**[STM32的ADC的采样时间及模拟信号的最大带宽](http://blog.sina.com.cn/s/blog_644f3cb40100sjt8.html)**作者：**[邵道君](http://blog.sina.com.cn/u/1682914484)

请看STM32技术参考手册的16.2节，和STM32F103xx数据手册的5.3.17节表44。可以在ST的中文网站下载到上述2个手册： [url=http://www.stmicroelectronics.com.cn/stonline/mcu/MCU\_Pages.htm]http://www.stmicroelectronics.com.cn/stonline/mcu/MCU\_Pages.htm[/url]

前面所说“STM32的ADC的采样及转换时间最小为1us”，实际上STM32的ADC采样及转换时间可以通过程序编程进行调整，共有8种选择，按 ADC模块的驱动时钟算分别为：

1.5 ADC时钟周期

7.5 ADC时钟周期

13.5 ADC时钟周期

28.5 ADC时钟周期

41.5 ADC时钟周期

55.5 ADC时钟周期

71.5 ADC时钟周期

239.5 ADC时钟周期

采样及转换时间最小的1us是在CPU时钟为56MHz(STM32F101xx为28MHz)时达到。

可能提出的问题

tS(fADC = 14 MHz)最小 = 0.107?s.实际上就是 =1.5T/14.还有几件事：

1.采样及转换时间最小的1us是在CPU时钟为56MHz(STM32F101xx为28MHz)时达到。"这句话出自何处？

我倒是看到fADC的最大值是14MHz.

2.这句话是自身矛盾的。为什么为了达到1Mhz的采样率，好芯片要更高的时钟？

3.采样时间和采样周期是两个概念。采样时间是整个ADC性能的重要参数。（请参考采样示波器的“采样”的概念）

4.您在上面提到的采样时间的选择，AD转换周期(TCONV) = 采样时间+ 12.5个周期。如何选择？依据是什么？

5.我上面提到的“模拟信号的最大带宽”这个概念，我在坛上关注了大半年，发现大家从未讨论过。而这对用好ADC是很重要的。

问题1：采样及转换时间最小的1us是在CPU时钟为56MHz(STM32F101xx为28MHz)时达到。"这句话出自何处？我倒是看到fADC的最大值是14MHz

答：这句话出自《STM32技术参考手册》第16.2节

ADC conversion time:

– STM32F103xx performance line devices: 1 us at 56 MHz (1.17 us at 72 MHz)

– STM32F101xx access line devices: 1 us at 28 MHz (1.55 us at 36 MHz)

关于fADC的最大值是14MHz，请看《STM32F103xx数据手册》第5.3.5节，表18下面的注释：

Specific conditions for ADC: fHCLK = 56 MHz, fAPB1 = fHCLK/2, fAPB2 = fHCLK, fADCCLK = fAPB2/4, ADON bit in the ADC\_CR2 register is set to 1.

即fADC在fHCLK=fAPB2=56MHz时达到14MHz。

2.这句话是自身矛盾的。为什么为了达到1Mhz的采样率，好芯片要更高的时钟？

103当然可以用和101同样的时钟并达到1Mhz的采样率。

但因为103的APB2可达72MHz,而101的APB2只可达36MHz，如果用户想使用APB2的其他外设，用户会不希望为了配合ADC的14M而采用低的时钟。所以这句话可以这样说：“在不影响ADC最高采样率的情况下，APB2最高可跑到56M"

3.采样时间和采样周期是两个概念。采样时间是整个ADC性能的重要参数。（请参考采样示波器的“采样”的概念）

103的DATASHEET有的，"ADC characteristics"

tS Sampling time， fADC = 14 MHz， 0.107 μs ，就是1.5×1/fADC

4.您在上面提到的采样时间的选择，AD转换周期(TCONV) = 采样时间+ 12.5个周期。如何选择？依据是什么？

要看外接的等效输入电阻及电容。103的DATASHEET上有一个公式

R(AIN)<ts/(f\* C\*ln(2 exp(N+2)))

还有一个图表

Ts (cycles) tS (μs) RAIN max (kΩ)

1.5 0.11 1.2

7.5 0.54 10

13.5 0.96 19

28.5 2.04 41

41.5 2.96 60

55.5 3.96 80

71.5 5.11 104

239.5 17.1 350

5.我上面提到的“模拟信号的最大带宽”这个概念，我在坛上关注了大半年，发现大家从未讨论过。而这对用好ADC是很重要的。

同样拿示波器举例，示波器前端有运放，再接ADC。对于运放，一般才提”模拟信号的最大带宽“，但对于后面的ADC，重要的指标是采样、保持、转换时间，根据Nyquist采样定律，至少2倍采样率才能重现波形，而通常做法是4~10倍，所以对于103，一定要提“模拟信号的最大带宽”这个概念的话，有工程意义上的值是1M/4~10 约为250k～100k。

1. STM32F的ADC最高采样率是1MHz。

2. 原翻译错了，根据aes\_sea的建议“采样及转换时间最小的1us是在CPU时钟为56MHz(STM32F101xx为28MHz)时达到”应改成 “在不影响ADC最高采样率的情况下，STM32F101xx的APB2最高可跑到56M..."

我的理解是：

STM32F10X的ADC的最高工作频率为14MHz，不能达到18MHz，（这是ST的无奈，当然已经很不错了。）而芯片的fSYSCLK和fADC 又必须保持2N倍的关系。

所以，若用户考虑ADC的性能，fADC=14MHz，则选SYSCLK=14\*4=56MHz。

若用户考虑系统的性能，fSYSCLK=72MHzf，则选fADC=72/6=12MHz。

此时，ADC最高采样率是12/14=0.857MHz。

3. 最小采样时间就是1.5个ADC时钟周期。

4. 由于芯片的保持电容<5P,而其模拟开关的导通电阻<1K,本来要求AD源的输出阻抗<1.2K,为了方便用户,简化AD源的电路，采样时间是可选的。

故上述表格本应反过来：根据用户的各个AD源的输出阻抗，选择合适的采样时间。

5. “根据Nyquist采样定律，至少2倍采样率才能重现波形”是老的观念。

否则，采样示波器的的最高频率怎么会远高于其ADC的最高频率。软件无线电也不会出现了。

技术高手应该明白“通带采样”的概念。而这又和“最小采样时间”关联。

带宽是相对于模拟信号而言的，即能够采集到的最大的模拟信号的频率。采样率是相对于数字信号而言，即采样时间所决定。这两点在AD转换器上都有体现，一般来讲AD的采样率要大于带宽，但也有小于带宽的，这就是将来高带宽的示波器的思想，用多个AD，利用相位差，在一个时钟周期内多个AD完成协同工作，这样就可以降低单个AD的采样率，是未来的趋势。

分享： [[http://simg.sinajs.cn/blog7style/images/common/sg_trans.gif](javascript:;)分享到新浪Qing](javascript:;) [http://simg.sinajs.cn/blog7style/images/common/sg_trans.gif](javascript:;)