

# 应用说明

## 32位APT32系列MCU

### ADC（模数转换器）参数与应用参考

---

2017.08

Author: Zheng Pan



**Revision History**

| NO | 日期     | 描述 | 作者 |
|----|--------|----|----|
| 1  | 2017-8 | 新建 | 郑攀 |

# 1. 综述

本文旨在介绍 APT32F 系列的 ADC 外设模块的性能参数点，以及提供软件编程和硬件设计参考。

## 1.1 模块结构与性能概述

APT32F 系列的 ADC 外设模块（Figure1-1），采用的是 SAR（逐次逼近电路）型电路结构，最大解析率为 12bit。

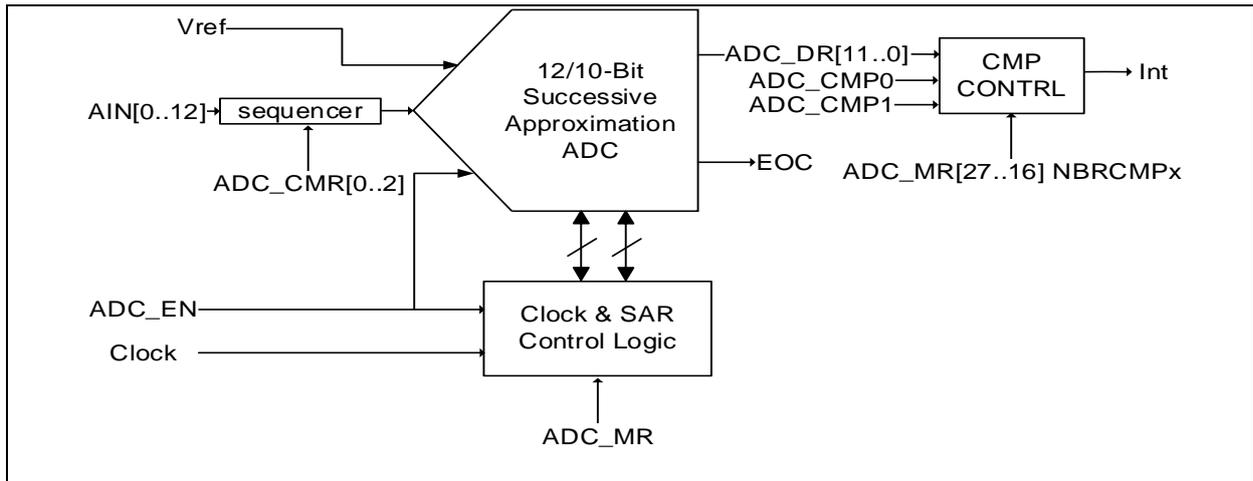


Figure 1-1 ADC模块框图

主要性能参数如下：

- 模拟输入范围：AVSS 到 AVREF (VDD)
- 最大12位的模拟输入分辨率。
- 5bit模块主时钟预分频控制。
- 采样时间（setup time）可配置，最大转换速度：12位500KSPS，10位1MSPS。
- ADC工作模式可配：单次转换，连续转换<sup>(1)</sup>。
- 支持多路输入AIN[12:0]，每路输入均可映射到实际芯片管脚。
- ADC启动转换触发源：软件启动，定时器1（TC1）脉冲匹配中断，外部管脚输入。
- 高级特性：

支持转换序列定义，序列长度可配，最长可达12次转换，每次转换可配置任意模拟通道作为输入。支持输入模拟量幅值自动监测功能，两个阈值寄存器，当采样值超出设定的监测阈值时产生超限中断，在设置了转换序列的情况下，可以通过寄存器配置，对转换序列中的哪次转换进行自动比较。

- 支持功耗管理，可只关闭ADC模拟部分，或者同时关闭ADC模拟部分和ADC主时钟。

## 1.2 ADC转换中的概念定义

ADC 的转换都是基于序列的转换，当序列中只有一次转换时，ADC 的转换表现为在触发后，只针对特定模拟输入通道执行一次数模转换。

ADC 模块中单次转换，连续转换，以及转换序列概念的定义如下：

#### **ADC 的单次采样：**

ADC 模块针对特定模拟输入通道，从采样到数字量化完成的一次操作。

#### **转换序列：**

由一个或者多个预定义的单次采样（定义了模拟输入通道的单次采样）组成的序列，序列的长度是可配置的，最大支持 12 个采样配置。

例如：我们可以定义一个长度为 3 的序列：ADC0，ADC1，ADC0，这个序列包含了三次转换，第一次转换 ADC0 通道，第二次转换 ADC1 通道，第三次转换 ADC0 通道。三次转换结束后，序列结束。

或者也可以定义长度为 1 的序列：ADC0。当序列长度为 1 时，ADC 只对指定通道进行一次采样。

#### **单次转换：**

转换序列被模块硬件完整执行一次的操作，执行完成后，模块硬件便停止自动操作，将停止标志置 1。当序列中只有一个对象时，单次转换即对指定对象进行一次转换。

#### **连续转换：**

转换序列被模块硬件完整执行一次后，模块硬件会自动从序列开头重新开始执行序列定义的操作，不会自动停止，连续转换模式下，不能保证在 ADC 被停止转换前转换序列被完整执行整数倍！

## 2. 软件编程参考

### 2.1 工作流程

当 ADC 转换被启动后，ADC 转换开始。当转换结束时，EOC 位(ADC\_SR[0])会自动被置 1，并且转换的结果被存入到 ADC\_DR 寄存器中以供读取。然后 ADC 进入等待状态。在开始另一个转换前，记住要先读取 ADC\_DR 寄存器的内容，否则下个转换结果将会覆盖前一个结果。

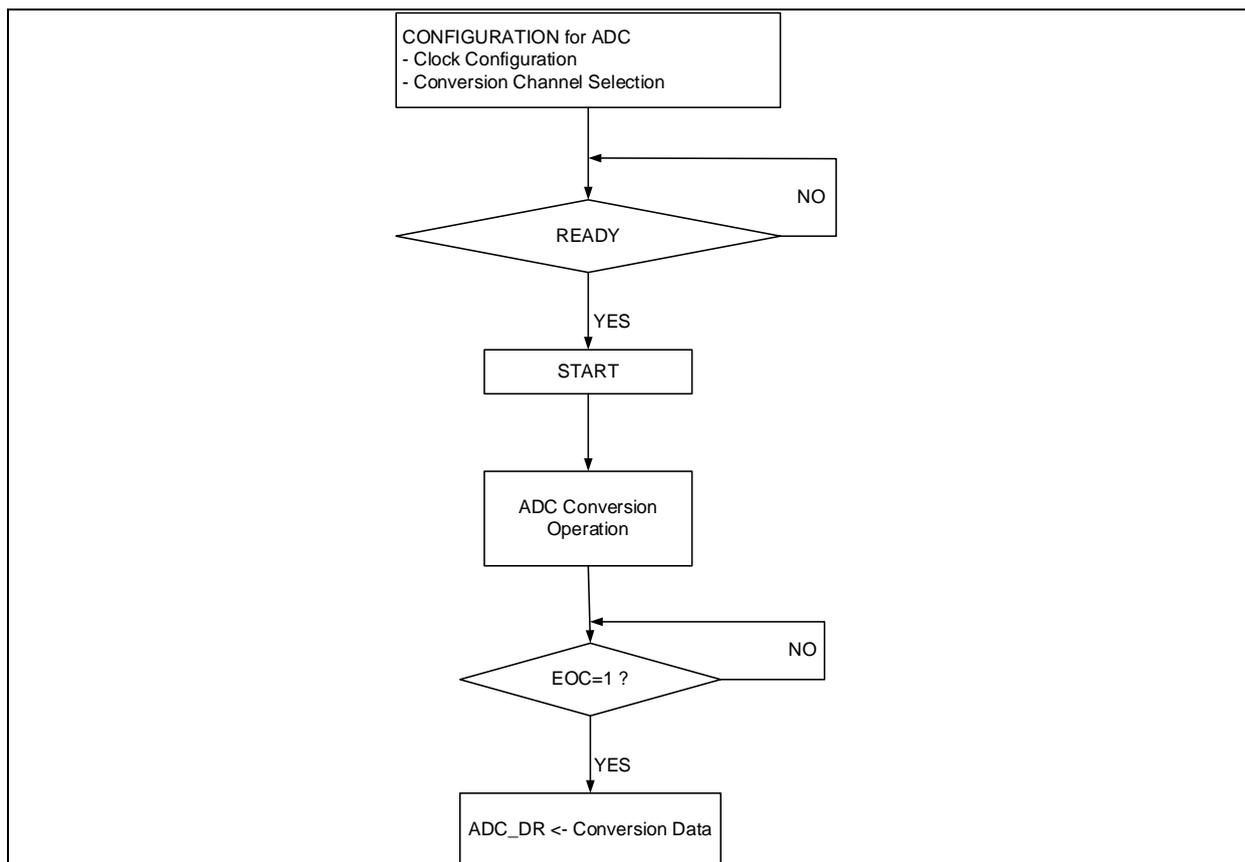


Figure 2-1 ADC工作流程图

### 2.2 软件操作流程

下面描述了在复位后使用 ADC 模块的基本操作流程：

1. 在 ADC\_ECR 中使能时钟
2. 在 ADC\_MR 和 ADC\_CMRx 中设置
  - A. 设置 ADC 的主时钟分频，PRLVAL 的值不能让模拟模块的工作时钟频率超过 10MHz。
  - B. 定义转换序列：转换次数(NBRCH)和哪些输入通道需要被转换(ADC\_CMRx 中的 CVx)。
  - C. 定义 ADC 的工作模式，单次转换还是连续转换模式(CONTCV)。
  - D. 定义 ADC 启动转换的源 (IES) 与触发条件 (ETRG)。
3. 使能 ADC 模块(ADC\_CR 中的 ADC\_EN)

4. 等待 ADC\_SR 中的 READY 位。只有当这个标志位被置 1 后，ADC 才能正常的开始转换。如果 ADC\_IMR 中的相应中断被使能，那么当 READY 标志置起的时候，会产生一个中断
5. 启动 ADC 转换，根据 ADC\_MR 的 IES 位：
  - A. IES =0 软件触发，需通过写 ADC\_CR 中的 START 位，开始转换
  - B. IES =0 硬件触发，初始化作为触发源的计数器或外部管脚。
6. 读取 ADC\_SR 的 EOC 标志位，等待 ADC 单次采样完成（注意请区别单次转换模式！）。
7. 然后 CPU 就可以读取 ADC\_DR 中的数字值，读取 ADC\_DR 寄存器时硬件自动清除 EOC 标志。
8. 如果需要停止转换有以下几种情况
 

在连续转换模式中，如果 CPU 判断不需要更多的转换了，那么它可以写 ADC\_CR 的 STOP 位停止转换。这样 ADC 就会停止工作并且等待下一个开始转换的请求。

在单次转换模式，ADC 不可以通过写 ADC\_CR 的 STOP 位停止，只能等待 ADC 完成一次完成的序列转换操作后自动停止。

## 2.3 典型应用代码实例

注：代码依赖于 APT32F101\_Release\_V1\_11 版的 BSP 代码。

### 2.3.1 以最高分辨率12bit，主时钟2分频，两通道（AIN2，AIN3）输入，转换序列长度2，单次转换。

```

/*****/

//ADC12_Init

//EntryParameter:NONE

//ReturnValue:NONE

/*****/

void ADC12_CONFIG(void)
{
    ADC12_RESET_VALUE();

    ADC12_CLK_CMD(ADC_CLK_CR, ENABLE);
    //ENABLE ADC CLK

    ADC12_Software_Reset();
    //software reset

    ADC12_Configure_Mode(ADC12_12BIT, ADC12_8CYCLES, One_shot_mode, 2, 2);
    //select 12BIT ADC, SAMPLING CYCLE=3, one shot mode, ADC_CLK=PCLK/2*1, NumConver=2

    ADC12_ConversionChannel_Config(ADC12_ADCIN2, 1);
    //set ADC2, 1st scan channel

    ADC12_ConversionChannel_Config(ADC12_ADCIN3, 2);
    //set ADC3, 2nd scan channel

    ADC12_CMD(ENABLE);
    //ENABLE ADC
    
```

```

        ADC12_ready_wait();
    // Waiting for ADC Ready
}

Void main (void) {

    unsigned int adc_data;

    /*ADC12_Control(ADC12_START);
    // Start conversion

    ADC12_EOC_wait();
    //End of conversion wait

    adc_data=ADC12_DATA_OUPUT();
    //ADC =ADC1 data

    ADC12_Control(ADC12_STOP);
}
    
```

代码说明：

ADC12\_CONFIG 函数用于实现 ADC 模块的完整配置但不启动 ADC 模块。

其中 ADC12\_RESET\_VALUE() -----用于将 ADC 的寄存器复位到上电复位值。  
 ADC12\_CLK\_CMD() -----用于开启和关闭 ADC 主时钟  
 ADC12\_Software\_Reset () -----用于软复位 ADC 模块  
 ADC12\_Configure\_Mode () -----用于软复位 ADC 模块  
 ADC12\_ConversionChannel\_Config () -----用于配置 GPIO 口以及映射模拟输入与转换序列  
 ADC12\_CMD() -----用于开启和关闭 ADC 模块

ADC12\_Configure\_Mode 函数参数说明

|   |                         |         |            |        |
|---|-------------------------|---------|------------|--------|
| ADC位数   | ADC转换时间<br>(setup time) | ADC工作模式 | 主时钟<br>预分频 | 转换序列长度 |
| ↓   | ↓                       | ↓       | ↓          | ↓      |
| <code>ADC12_Configure_Mode(ADC12_12BIT , ADC12_8CYCLES , <u>One shot mode</u>, 2 , 2);</code> |                         |         |            |        |

实际应用中，有以下几点需要注意：

#### 1、 时钟频率和转换时间

理论的转换时间计算过程如下，实际应用中由于有 ADC 数据取出和处理的代码执行时间，所以最终转换速率请以实际代码为准。

ADC 工作的时钟是从 PCLK 获得的。AD 转换的过程需要总共(setup+12/10)个时钟周期。setup 时间可以通过 ADC\_CR 寄存器里的“SAMPLE”位(ADC\_CR[6:5])设置。ADC 模块提供一个时钟分频器，该分频器是一个 6 位计数器，由模式寄存器里的 PRLVAL 控制。下面的表达式给出了系统频率和 ADC 模拟模块时钟频率之间的关系。

如果 PRLVAL 是 0, 那么  $F\_ANA = PCLK$

否则 PRLVAL 是其它任何值的话,  $F\_ANA = PCLK / (2 * PRLVAL)$

PRLVAL 的值必须保证采样速度不超过手册规定的最大值(12 位 500KSPS, 10 位 1MSPS)。如果 PCLK/2 被选择位转换时钟, 并且 PCLK 频率是 21MHz, 那么一个时钟周期就是 95.2ns。转换速度计算如下(假设 setup 时间为默认值 6 个周期):

12 位 - (6 个 setup 时钟周期) + (每位 1 个时钟转换周期 x 12 位) + (3 个同步和结果处理时钟周期) = 21 个周期

$21 \times 95.2ns = 2us$  (500ksps)

10 位 - (6 个 setup 时钟周期) + (每位 1 个时钟转换周期 x 10 位) + (3 个同步和结果处理时钟周期) = 19 个周期

$19 \times 95.2ns = 1.7us$  (588ksps)

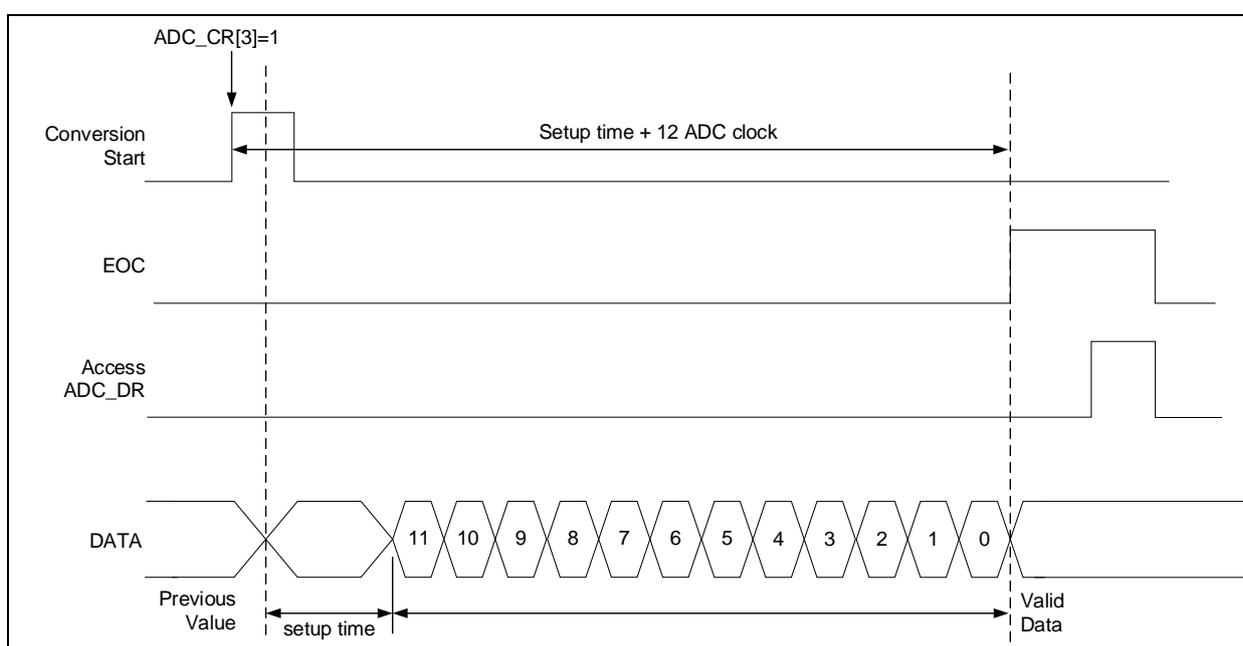


Figure 2-2 ADC工作时序图

## 2、转换序列长度与数据寄存器的值

当转换序列长度大于 1 时, 需要适当降低 ADC 的采样频率, 以保证数据寄存器不至于溢出。可在读取数据寄存器前检查 ADC\_SR 的 OVR 标志位是否为 1, 以此为依据调整 ADC 的采样频率到合适的值。

## 3、转换序列与工作模式

即使在单次转换模式, 如果 NBRCH[5:0]的值大于 0, ADC 也会进行多次采样。

此时如果转换序列与通道的映射寄存器 ADC\_CM Rx 配置了两个以上的模拟输入, 转换结果寄存器的值将是令人迷惑的形式。

所以, 在只需要一个模拟输入时, 请确保 NBRCH[5:0]的值等于 0。

在有多个模拟输入时, 请确保 ADC\_CM Rx 配置与序列长度以及输入能一一对应起来。

下面为实际需求的配置实例:

A. ADC2 为唯一的模拟输入通道, 每次触发启动转换后只需要一个转换结果。

序列长度 1 单次转换模式 ADC2 作为转换序列首个待转换输入。

NBRCH = 0, CONTCV = 0, CV0 = 0x2

- B. ADC2 为唯一的模拟输入通道，每次触发启动转换后需要多个转换结果。  
序列长度 1 连续转换模式 ADC2 作为转换序列首个待转换输入。  
NBRCH = 0, CONTCV = 1, CV0 = 0x2,  
此外应适当增加 PRLVAL 的值，以降低 ADC 的采样频率，保证数据不会覆盖溢出
- C. ADC2 ADC3 ADC5 为模拟输入通道，组成转换序列 ADC3 ADC2 ADC5 ADC3，且每次触发启动转换后，序列中每次采样都只需要一个转换结果。  
序列长度 4 单次转换模式 序列映射配置  
NBRCH = 3, CONTCV = 0, CV0 = 0x3, CV1 = 0x2, CV2 = 0x5, CV3 = 0x3  
此外应适当增加 PRLVAL 的值，以降低 ADC 的采样频率，保证数据不会覆盖溢出
- D. ADC2 ADC3 ADC5 为模拟输入通道，组成转换序列 ADC3 ADC2 ADC5 ADC3，且每次触发启动转换后，序列中每次采样需要多个转换结果。  
序列长度 4 连续转换模式 序列映射配置  
NBRCH = 3, CONTCV = 1, CV0 = 0x3, CV1 = 0x2, CV2 = 0x5, CV3 = 0x3  
此外应适当增加 PRLVAL 的值，以降低 ADC 的采样频率，保证数据不会覆盖溢出

### 3. 硬件设计参考

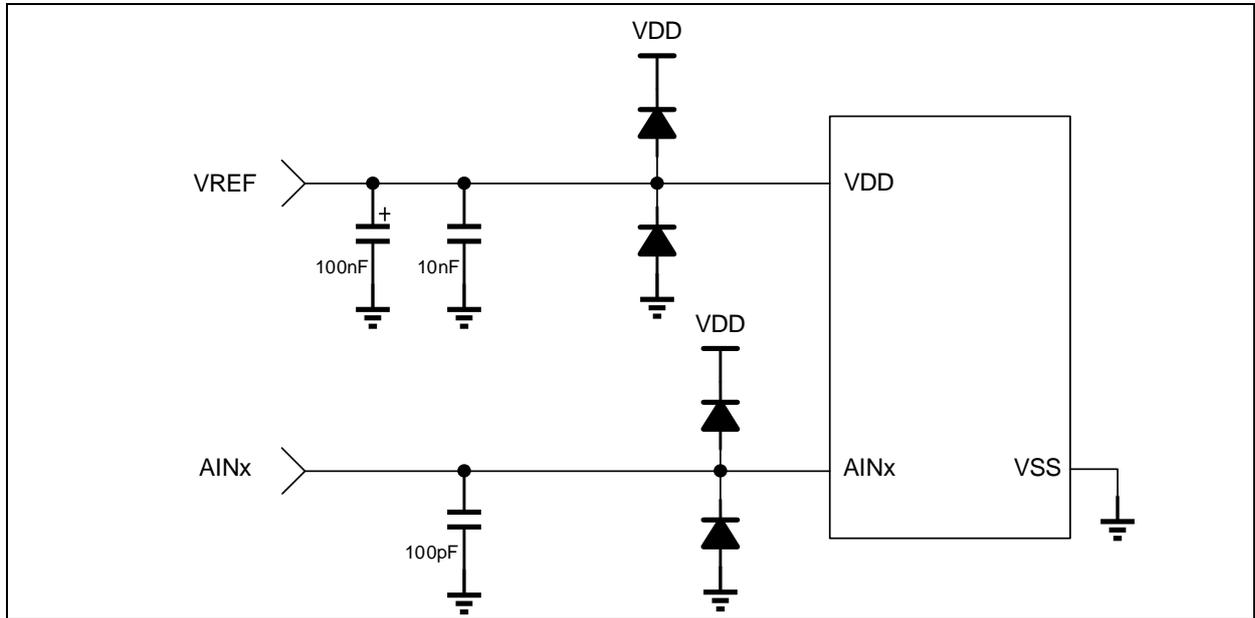


Figure 3-1 参考电路

为改善 ADC 模块的转换精度，请您根据以下参考原则设计您的硬件应用电路。

- 1、ADC 模块的参考电压与 VDD 为同一个外部管脚，可按如图 Figure 3-1 的参考电路添加 VDD 去耦电容，去耦电容应紧靠 VDD 管脚放置，以减小电源耦合噪声对 ADC 转化性能的劣化。
- 2、对于旋转为 ADC 输入的管脚，可按图 Figure 3-1 添加输入管脚滤波电容，以改善来自空间或线路的串扰噪声对 ADC 转化性能的劣化。
- 3、ADC 的输入信号源与芯片间的输入路径上，ADC 输入的 PCB 迹线周围需做铺地处理，与迹线相邻的布线层也不要做强干扰发射的其他信号并行。
- 4、与 ADC 输入管脚相邻的 mcu 管脚不要分配强干扰的数字信号，未使用的管脚置成输出或下拉。
- 5、添加精确外部时钟晶振，可改善 ADC 的 SNR 参数。