

32 位 微控制器

HC32F460 系列的模数转换器 ADC

适用对象

系列	产品型号
HC32F460	HC32F460JEU A
	HC32F460JETA
	HC32F460KEUA
	HC32F460KETA
	HC32F460PETB

目 录

1	摘要	4
2	ADC 简介	4
2.1	功能简介	4
2.2	主要特性	5
2.3	引脚配置	6
3	ADC 应用	7
3.1	模拟输入引脚与通道	7
3.2	模拟输入的采样时间和转换时间	8
3.3	模式和功能	9
3.4	序列 A 单次扫描模式	10
3.4.1	说明	10
3.4.2	应用	10
3.5	序列 A 连续扫描模式	11
3.5.1	说明	11
3.5.2	应用	11
3.6	双序列扫描模式	12
3.6.1	说明	12
3.6.2	应用	14
3.7	转换数据平均功能	14
3.7.1	说明	14
3.7.2	应用	14
3.8	模拟看门狗	15
3.8.1	说明	15
3.8.2	应用	15
3.9	内部模拟通道	16
3.9.1	说明	16
3.9.2	应用	16
3.10	可编程增益放大器 PGA	17
3.10.1	说明	17
3.10.2	应用	17
3.11	协同模式	18

3.11.1	单次并行触发模式	19
3.11.2	单次延迟触发模式	20
3.11.3	循环并行触发模式	22
3.11.4	循环延迟触发模式	23
4	例程讲解	25
4.1	基本流程	25
4.2	应用程序源码说明	26
4.2.1	应用程序基本结构	26
4.2.2	重要源码讲解	27
4.2.3	程序执行现象	30
5	总结	31
6	版本信息 & 联系方式	32

1 摘要

本应用笔记主要介绍 HC32F460 系列 MCU 的模数转换器（以下简称 ADC）的特点及使用方法，包括扫描模式、转换数据平均功能、模拟看门狗、可编程增益放大器和协同模式等。

2 ADC 简介

2.1 功能简介

HC32F460 系列 MCU 内部集成 ADC1 和 ADC2 两个 ADC 模块（系统框图如图 2-1），挂载于 AHB-APB（APB3）总线，可配置 12 位、10 位和 8 位分辨率，支持最多 16 个外部模拟输入通道和 1 个内部基准电压/8bitDAC 输出的检测通道。这些模拟输入通道可以任意组合成一个序列（序列 A 或序列 B），一个序列可以进行单次扫描（包括两个动作：采样和转换），或连续扫描。支持对任意指定通道进行连续多次扫描，并对转换结果进行平均。ADC 模块还搭载模拟看门狗（以下简称 AWD）功能，可对任意指定通道的转换结果进行监视，检测是否超出设定的阈值。

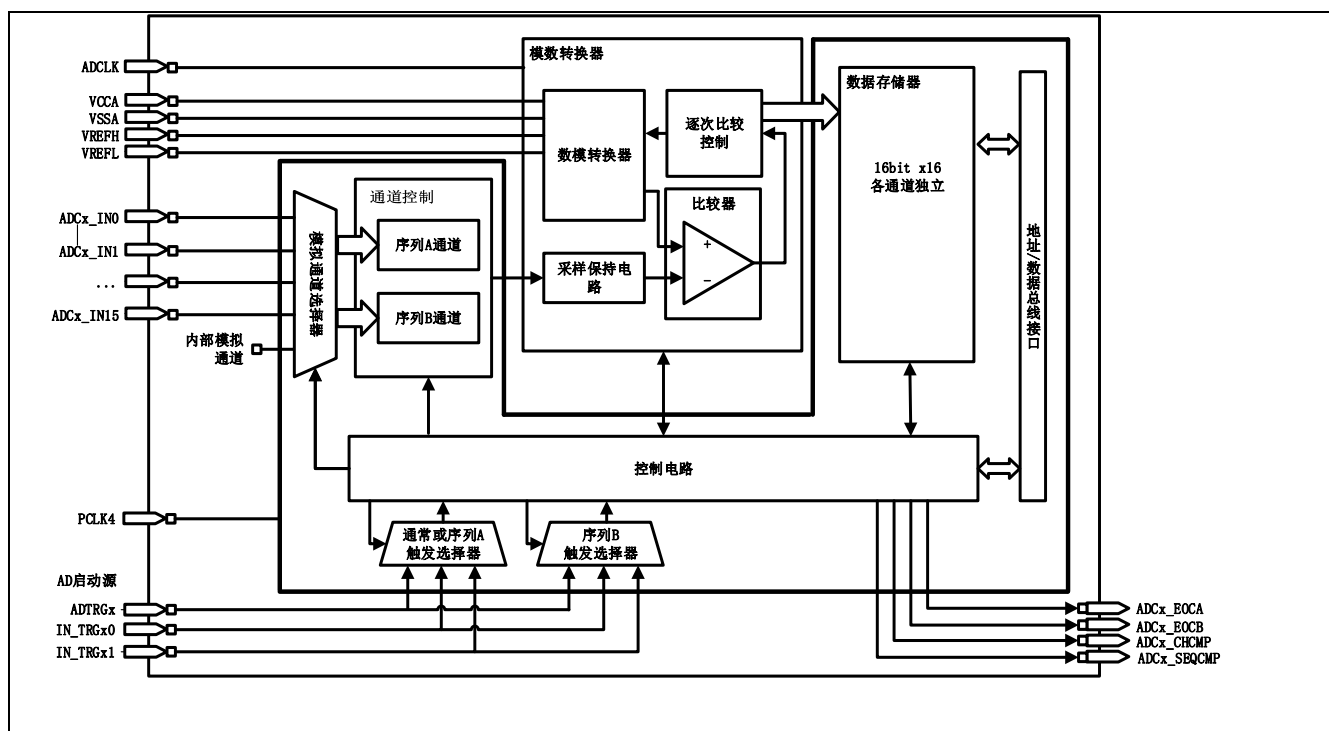


图 2-1 ADC 系统框图

2.2 主要特性

HC32F460 系列 MCU 的 ADC 具有如下主要特性:

1) 高性能

- 可配置 12 位、10 位和 8 位分辨率
- 周边时钟（数字时钟）PCLK4 和 A/D 转换时钟 ADCLK 的频率比可选择:
- $PCLK4 : ADCLK = 1 : 1, 2 : 1, 4 : 1, 8 : 1, 1 : 2, 1 : 4$
- ADCLK 可选与系统时钟 HCLK 异步的 PLL 时钟，此时 $PCLK4 : ADCLK = 1 : 1$
- 采样率: 2.5MSPS ($PCLK4 = ADCLK = 60MHz$, 12 位分辨率, 采样 11 周期)
- 各通道可独立设置采样时间
- 各通道独立数据寄存器
- 数据寄存器可配置数据对齐方式
- 连续多次转换平均功能
- 模拟看门狗, 监视转换结果
- 不使用时可将 ADC 模块设定为停止状态

2) 模拟输入通道

- 最多有 16 个外部模拟输入采样通道
- 1 个内部基准电压/8bitDAC 输出的检测通道
- 最多有 16 个外部模拟输入引脚，外部模拟输入引脚可与采样通道自由映射

3) 转换开始条件

- 软件启动开始转换（只支持序列 A）
- 外设事件触发开始转换（支持序列 A 和序列 B）
- 外部引脚触发开始转换（支持序列 A 和序列 B）

4) 转换模式

- 序列 A 单次扫描
- 序列 A 连续扫描
- 双序列扫描
- 协同模式

5) 中断与事件信号输出

- 序列 A 扫描结束中断和事件 ADC_EOCA
- 序列 A 扫描结束中断和事件 ADC_EOCB
- 模拟看门狗通道比较中断和事件 ADC_CHCMP，序列比较中断和事件 ADC_SEQCMP
- 上述 4 个事件输出都可启动 DMA

2.3 引脚配置

HC32F460 系列 MCU 的 ADC1 有 17 个采样通道，最多支持 16 个外部模拟输入引脚，通道 0~15 可与外部模拟输入引脚自由映射，通道 16 用于内部基准电压/8bitDAC 的输出检测。

ADC2 有 9 个采样通道，最多支持 8 个外部模拟输入引脚，通道 0~7 可与外部模拟输入引脚自由映射，通道 8 用于内部基准电压/8bitDAC 的输出检测。

3 ADC 应用

3.1 模拟输入引脚与通道

HC32F460 系列 MCU 的 ADC 模块模拟输入引脚等配置如表 3-1。

项目		单元1 (ADC1)	单元2 (ADC2)
电源		VCCA	
		VSSA/VREFL	
基准电压		VREFH	
模拟通道	CH0	ADC1_IN0	ADC12_IN4
	CH1	ADC1_IN1	ADC12_IN5
	CH2	ADC1_IN2	ADC12_IN6
	CH3	ADC1_IN3	ADC12_IN7
	CH4	ADC12_IN4	ADC12_IN8
	CH5	ADC12_IN5	ADC12_IN9
	CH6	ADC12_IN6	ADC12_IN10
	CH7	ADC12_IN7	ADC12_IN11
	CH8	ADC12_IN8	内部模拟通道（基准电压/8bitDAC输出）
	CH9	ADC12_IN9	-
	CH10	ADC12_IN10	-
	CH11	ADC12_IN11	-
	CH12	ADC1_IN12	-
	CH13	ADC1_IN13	-
	CH14	ADC1_IN14	-
	CH15	ADC1_IN15	-
	CH16	内部模拟通道（基准电压/8bitDAC输出）	-
PGA		ADC1_IN0~3, ADC12_IN4~7, 8bitDAC_1输出中任意1通道	-

项目		单元1（ADC1）	单元2（ADC2）
硬件触发源	外部引脚	ADTRG1	ADTRG2
	片内周边	IN_TRG10	IN_TRG20
		IN_TRG11	IN_TRG21

表 3-1 ADC 引脚与通道

默认情况下，ADC1 的 CH0（通道 0）对应模拟输入引脚 ADC1_IN0，CH1 对应 ADC1_IN1.....，CH16 为内部模拟通道，只能用于检测内部基准电压、8bitDAC1 或 8bitDAC2。也就是，默认情况下，ADC1 序列 A 的通道选择寄存器 ADC1_CHSELRA0（或序列 B 的通道选择寄存器 ADC1_CHSELRB0）的 bit0 置 1，即选择了模拟输入引脚 ADC1_IN0，bit1 置 1，即选择模拟输入引脚 ADC1_IN1；序列 A 的通道选择寄存器 ADC1_CHSELRA1（或序列 B 的通道选择寄存器 ADC1_CHSELRB1）的 bit0 置 1，即选择了内部模拟输入，用于检测内部基准电压、8bitDAC1 或 8bitDAC2。

但是，HC32F460 系列的 ADC 模块具有模拟输入引脚与通道自由映射（除用于检测内部模拟输入的通道外）的功能，可满足用户不同的应用需求。例如，可将引脚 ADC12_IN10 映射到 ADC1 的一个通道（如 CH0，不能映射到 CH16），或同时映射到多个通道（如 CH0、CH2 和 CH3）。

ADC2 通道与引脚的默认对应关系和通道重映射与 ADC1 的类似。

关于通道重映射，固件例程 `adc_11_channel_remap` 给出了其具体用法。

3.2 模拟输入的采样时间和转换时间

关于 ADC 时间的详细说明，请参考用户手册“模拟输入的采样时间和转换时间”一节中，寄存器 `ADC_SSTR` 和 ADC 电气特性部分，请严格按照手册要求设置采样时间。在满足应用需求的情况下，请尽量将采样时间设置得大一些，尤其在对多个通道采样时，如果通道的采样时间设置偏小，可能会导致相邻通道（如序列 A 配置了 CH0、CH5、CH7，那么 CH0 和 CH5、CH5 和 CH7 都是相邻通道）之间透过采样电容发生耦合，而使转换结果不准确。

3.3 模式和功能

ADC 的通道可配置为序列 A 或序列 B，序列 A 和序列 B 可单独设置不同的触发源。两个序列共有四种扫描方式：

- 序列 A 单次扫描；
- 序列 A 连续扫描；
- 序列 A 单次扫描，序列 B 单次扫描；
- 序列 A 连续扫描，序列 B 单次扫描。

各通道还可设置平均功能，可连续扫描设定次数后，计算转换的平均值，并将平均值保存到数据寄存器中；模拟看门狗 AWD 在通道转换结束后对转换结果进行比较，可生成通道比较中断和事件 ADC_CHCMP，在整个序列扫描结束后，根据各通道比较结果生成序列比较中断和事件 ADC_SEQCMP；可编程增益放大器 PGA，可对模拟输入信号放大后再转换；协同工作模式下，ADC1 和 ADC2 可同时转换或连续交替转换；模拟输入引脚可与 ADC 通道的自由映射，再结合协同模式，可实现对指定模拟输入的高频扫描。

3.4 序列 A 单次扫描模式

3.4.1 说明

在此模式下，ADC 执行单个或多个通道的单次扫描，并在转换完成后停止，示意如图 3-1。

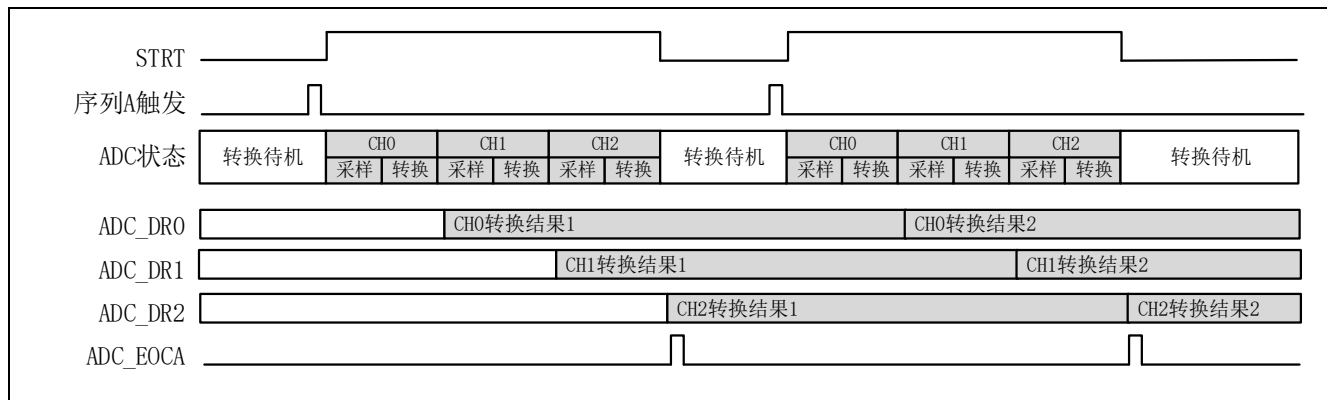


图 3-1 ADC 单次扫描模式

3.4.2 应用

该模式可以设置单个或多个通道。设置多个通道时，可实现对多个通道进行依次扫描，这些通道可设置不同的采样时间，用户不必在扫描过程中停止 ADC，即可以不同的采样时间重新扫描下一个通道，可避免额外的 CPU 负载以及繁重的软件开发。

该模式是最简单的 ADC 模式，应用方式灵活。如在系统启动前，可以用这种模式检测系统的一些状态信息，如电压、压力、温度等，以确定系统是否可以正常启动；在系统运行中，可用这种模式，按需检测系统状态，以获取系统实时状态。

应用例程 `adc_01_sa_base` 给出了该模式的具体用法。

3.5 序列 A 连续扫描模式

3.5.1 说明

连续扫描模式可对单个通道或多个通道进行连续不断的扫描，如图 3-2。连续扫描模式允许 ADC 在后台工作。因此，ADC 可在没有任何 CPU 干预的情况下对通道进行连续（循环）扫描通道。此外，还可以在连续扫描模式下使用 DMA，从而降低 CPU 负载。

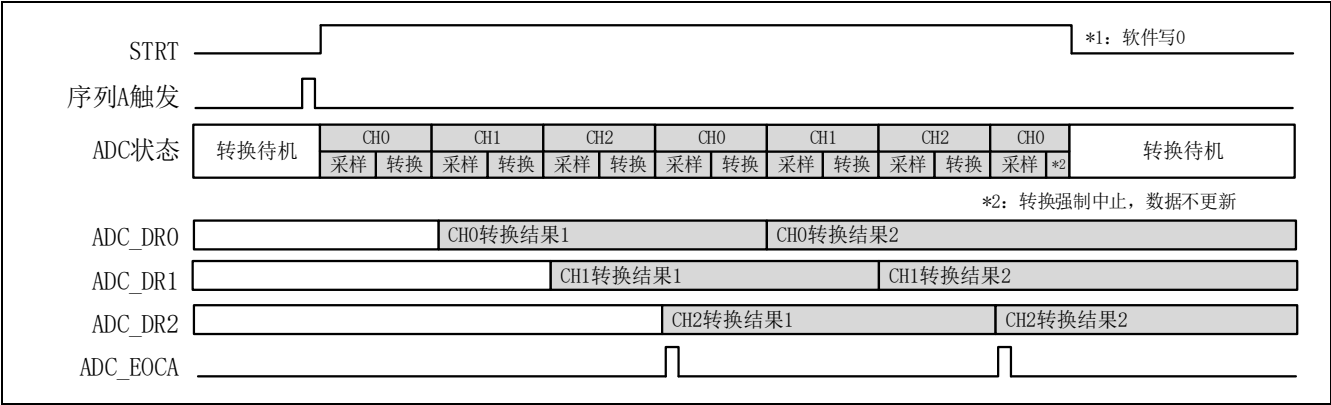


图 3-2 ADC 连续扫描模式

3.5.2 应用

此模式设置单个通道时，可用于监视电池电压、测量和调节烤箱温度等应用。在用于调节烤箱温度时，系统将读取温度并与用户设置的温度进行比较。当烤箱温度达到所需温度时，关闭加热电阻器的电源。

设置多个通道时，与多通道单次扫描模式类似，只是在完成序列的最后一个通道后不会停止扫描，而是从第一个通道重新开始扫描并无限循环下去。多通道连续扫描模式，可用于监视多电池充电器中的多个电压和温度。系统在充电过程中读取每节电池的电压和温度。当电压或温度达到最大值时，将切断相应电池与充电器的连接。

应用例程 `adc_01_sa_base` 中有该模式的设置以及简单的应用方法。

3.6 双序列扫描模式

3.6.1 说明

此处将“序列 A 单次扫描、序列 B 单次扫描”和“序列 A 连续扫描、序列 B 单次扫描”两种模式整合为双序列扫描模式进行介绍。双序列扫描模式，只是在前两种模式中增加了序列 B 的扫描。双序列扫描模式下，序列 B 必须由外部引脚或内部事件触发转换，软件启动对序列 B 无效，序列 A 可由软件启动扫描，也可由外部引脚或内部事件触发扫描。序列 B 的优先级高于序列 A，其与序列 A 的竞争关系如表 3-2。

A/D转换	触发信号发生	处理方式	
		ADC_CR1.RSCHSEL=0	ADC_CR1.RSCHSEL=1
序列A转换过程中	序列A触发	触发信号无效	
	序列B触发	1) 序列A的转换被中断，开始序列B转换 2) 序列B的转换完成后，序列A从被中断的通道开始继续转换	1) 序列A的转换被中断，开始序列B转换 2) 序列B的转换完成后，序列A从第一个通道开始重新转换
序列B转换过程中	序列A触发	序列B全部通道转换完成后，开始序列A转换	
	序列B触发	触发信号无效	
AD空闲中，序列A，B同时触发		序列B先启动，全部通道转换完成后，开始序列A转换	

表 3-2 序列 A 和序列 B 的竞争关系

配置 ADC_CR1.RSCHSEL 为 0 时，当序列 A 被中断后，恢复时，从被中断通道继续扫描，如图 3-3。

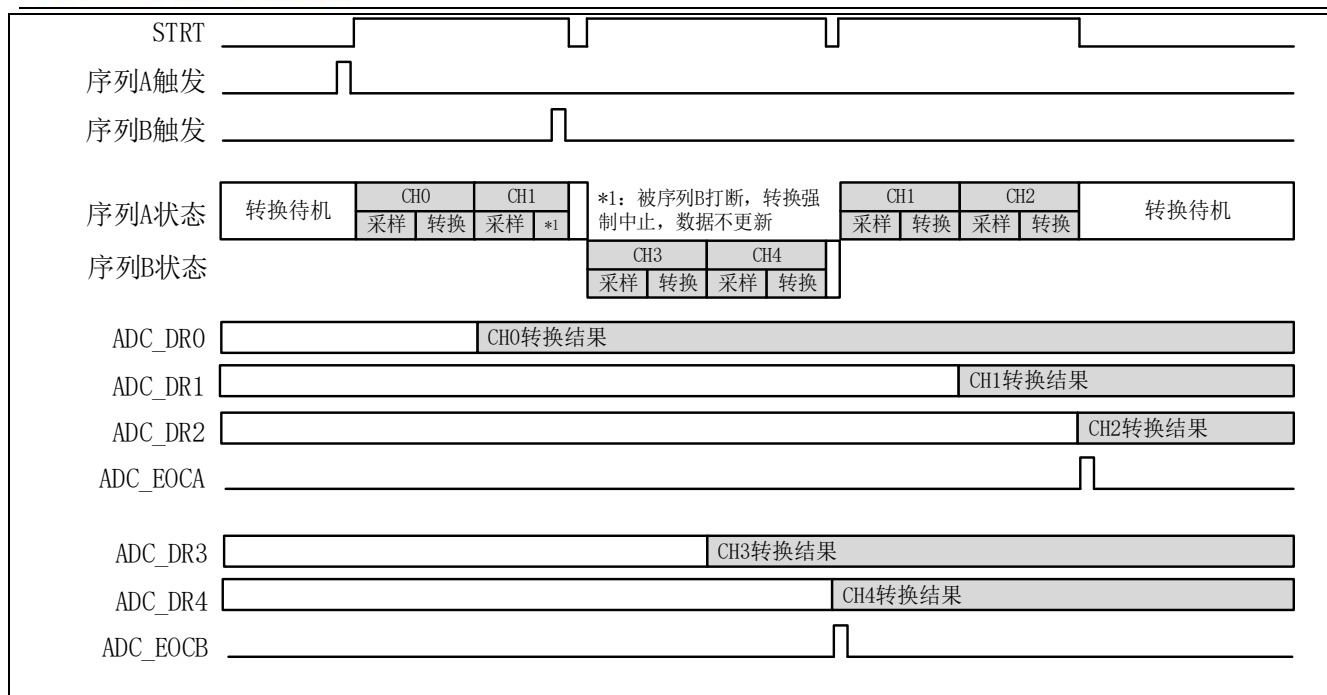


图 3-3 序列 A 从被中断通道恢复扫描

配置 **ADC_CR1.RSCHSEL** 为 1 时, 当序列 A 被中断后, 恢复时, 从序列第一个通道重新开始扫描, 如图 3-4。

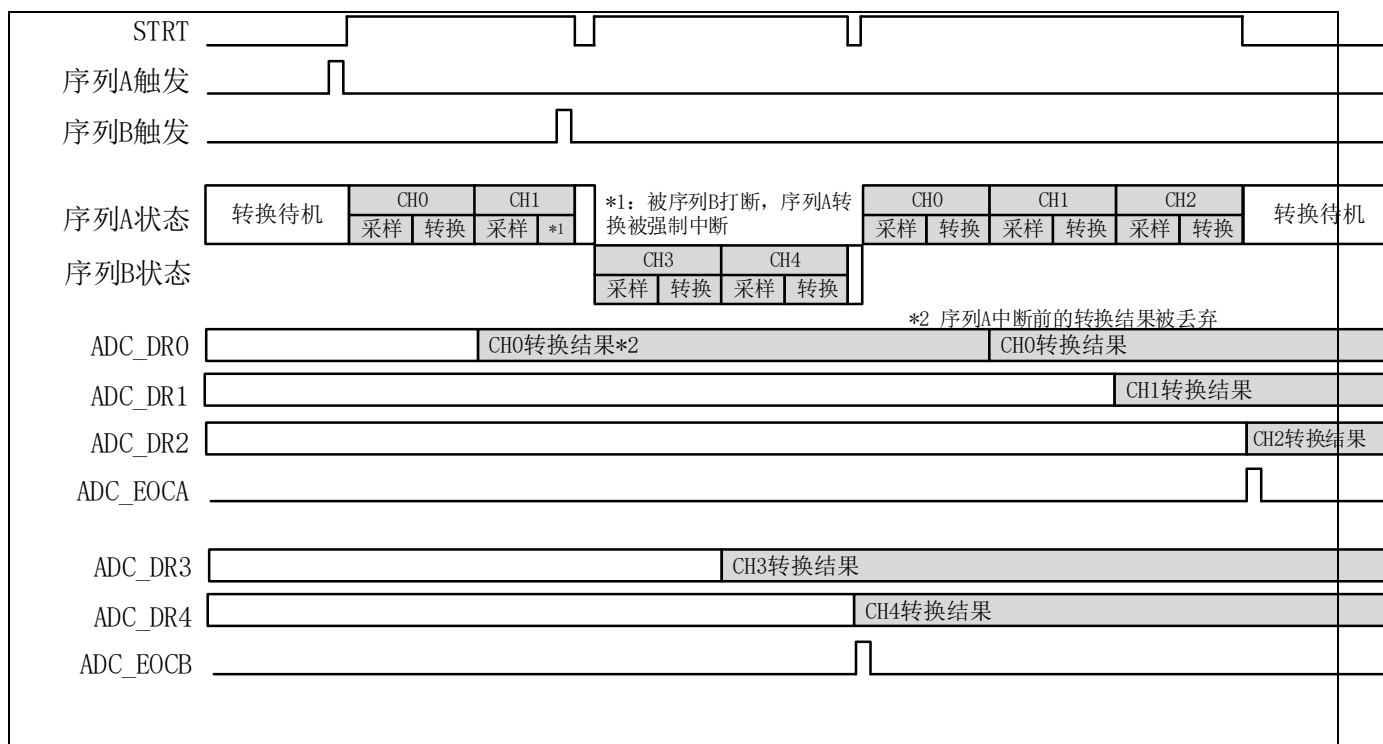


图 3-4 序列 A 从本序列第一个通道恢复扫描

注意:

- 序列 A 和序列 B 不能选择相同的通道，不能选择相同的触发源。

3.6.2 应用

可在前两种模式的应用中，加入需要实时响应（更高优先级）扫描的通道，将其配置为序列 B。例程 `adc_04_sa_sb_event_trigger` 实现了双序列扫描的基本用法。

3.7 转换数据平均功能

3.7.1 说明

转换数据平均功能，可设置连续扫描 2、4、8、16、32、64、128 或 256 次后，将转换结果平均后，保存到数据寄存器。该功能可去除一定的噪声成分，使结果更加准确。该功能的优势是在无任何硬件变更的情况下提高 ADC 的准确度，缺点是降低了转换速度和频率（相当于降低了有效采样率）。

3.7.2 应用

针对不同的应用，可设置不同的连续扫描次数，该次数取决于需要的精度、最低转换速度等。应用笔记没有单独为该功能提供例程，在例程 `adc_01_sa_base` 中有该功能的配置方式。

3.8 模拟看门狗

3.8.1 说明

HC32F460 系列 MCU 的模拟看门狗，可配置为上下限比较或区间比较，如图 3-5。

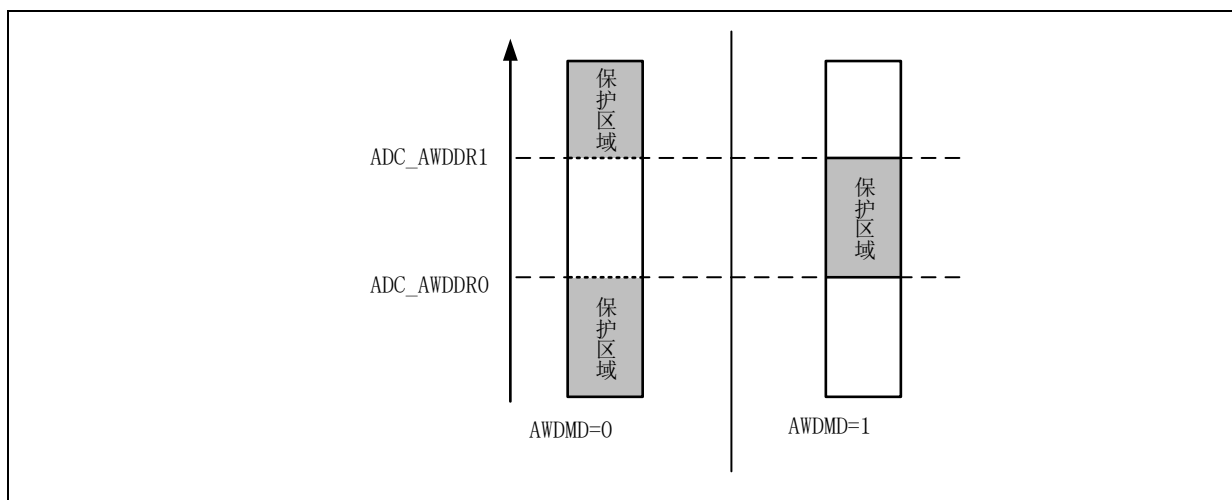


图 3-5 模拟看门狗比较条件

用户可预先设置比较条件和相应的上下限或区间。在通道转换结束后，模拟看门狗对转换结果进行比较，如果满足比较条件，则产生通道比较中断和事件 ADC_CHCMP，在整个序列扫描结束后，根据各通道比较结果生成序列比较中断和事件 ADC_SEQCMP。每个使能模拟看门狗的通道，只要其转换结果满足比较条件，都会产生一次中断和事件 ADC_CHCMP；每个序列，只要其中一个通道的转换结果满足比较条件，都会产生中断和事件 ADC_SEQCMP。在条件满足后，一个序列只产生一次中断和事件 ADC_SEQCMP。也就是，一个 ADC 模块，一轮扫描结束后，可产生多次中断和事件 ADC_SEQCMP，最多两次（因为最多只有两个序列）中断和事件 ADC_SEQCMP。

注意：

- 不推荐同时使用 ADC_CHCMP 中断和 ADC_SEQCMP 中断。

3.8.2 应用

在一些控制系统中，需要严格监测电压、压力、温度等信号的范围，使用模拟看门狗能够快速检测到这些信号的异常状况，并做出相应的应对措施，以确保设备安全。例程

adc_08_sa_sb_awd_base 给出了模拟看门狗的配置和基本应用方法；但通常，模拟看门狗的中断用法更为高效，应用也更为普遍，例程 adc_09_sa_sb_awd_interrupt 给出了模拟看门狗的中断配置和用法。

3.9 内部模拟通道

3.9.1 说明

ADC1 和 ADC2 都具有一个用于检测内部模拟输入的通道，分别为通道 16 和通道 8，用来检测三个可选择的内部模拟输入量，内部基准电压、8bitDAC1 输出或 8bitDAC2 输出，如图 3-6。

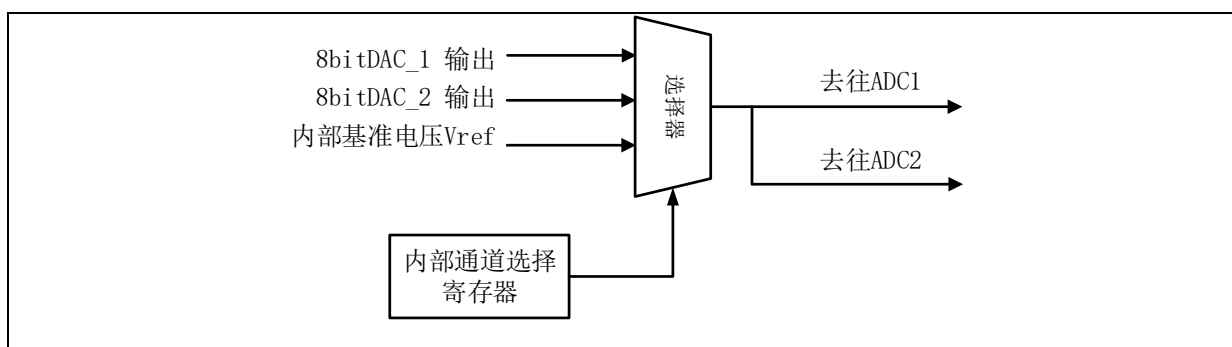


图 3-6 内部模拟通道选择

注意：

- 只能选择 ADC1 和 ADC2 其中之一的内部检测通道，来检测三个内部模拟输入的其中一个，不能同时使用 ADC1 和 ADC2 的内部检测通道。

3.9.2 应用

在一些系统中，可能由于某些原因导致 ADC 的参考电压不稳定，从而无法知道模拟输入的实际电压值，这时，可用 ADC1 或 ADC2 的内部检测通道，来检测内部基准电压（在系统工作电压正常时恒为 1.1V），来反推 ADC 当前的参考电压，从而得知模拟输入当前的实际电压值。具体实现如下：

1. 在某已知参考电压 VREF1 下测得内部基准电压的 ADC 值为 VAL1；
2. 随着系统的运行，参考电压可能随着供电源（如电池）的电压降低而下降，此时测得内部基准电压的 ADC 值为 VAL2，设此时参考电压为 VREF2；
3. 由于内部基准电压是恒定的，所以有：

$$VAL1 \times VREF1 = VAL2 \times VREF2;$$

$$VREF2 = (VAL1 \times VREF1) / VAL2;$$

由此已知当前参考电压为 VREF2，就不难得到模拟输入的实际电压了。

例程 adc_10_internal_channel 展示了内部通道的各种配置和简单用法。

3.10 可编程增益放大器 PGA

3.10.1 说明

HC32F460 系列 MCU 集成了可编程增益放大器 PGA，能对模拟信号进行放大处理，可节省 MCU 外接运算放大器的硬件成本。PGA 电路先将模拟信号进行放大，然后再将放大后的模拟信号输出至 ADC 模块进行采样转换，其工作示意图如图 3-7。

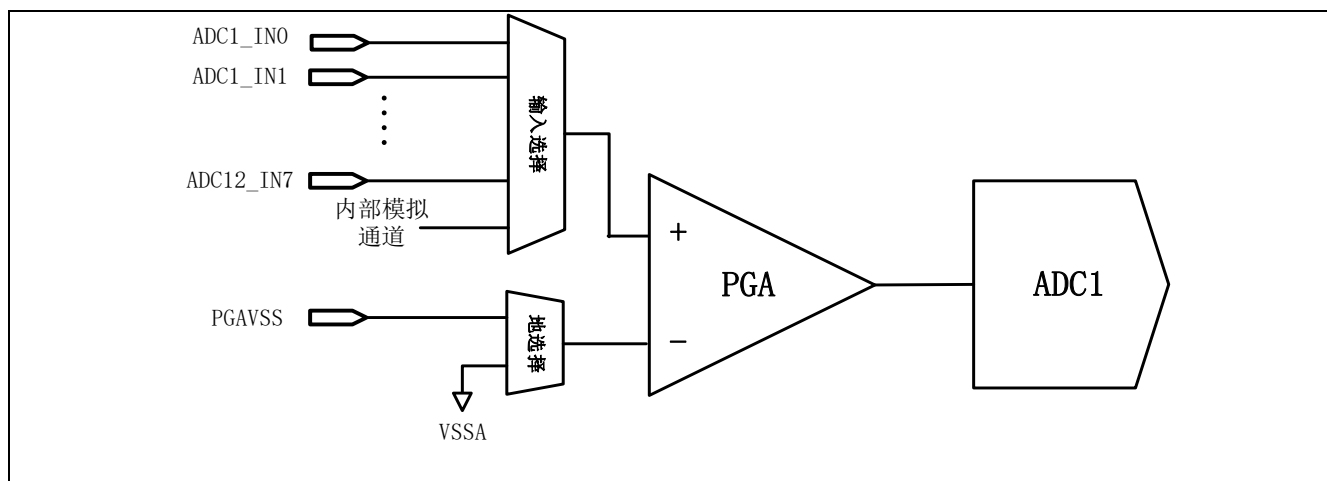


图 3-7 PGA 工作示意图

注意：

- 只有 ADC1 支持 PGA；
- PGA 通道直接与模拟输入引脚对应，其对应引脚映射的通道必须被 ADC1 序列 A 或序列 B 的通道选择寄存器选中，才可对该模拟输入进行放大。

3.10.2 应用

用户可根据实际的应用场景，选择合适的放大倍数，对模拟输入进行放大。模拟输入电压和放大倍数必须满足如下条件：

$$0.1 * V_{CCA} / \text{Gain} \leq V_I \leq 0.9 * V_{CCA} / \text{Gain}$$

其中， V_{CCA} 是模拟电源电压，Gain 是放大倍数， V_I 是模拟输入电压，具体请参考用户手册 PGA 相关部分。关于 PGA 的配置和用法，请参考例程 `adc_12_adc1_pga`。

3.11 协同模式

HC32F460 系列 MCU 具有两个 ADC 模块，可使用 ADC 协同工作模式。在协同工作模式下，转换启动只能由 ADC1 的触发源触发启动，且软件启动无效。

协同模式可配置为以下四种协同模式：

- 单次并行触发模式
- 单次延迟触发模式
- 循环并行触发模式
- 循环延迟触发模式

注意：

- 协同模式只能由 ADC1 配置；
- 设置为协同工作模式的 ADC，其配置（扫描模式、分辨率和数据对齐方式等）应尽量相同；具体通道无需相同，但是通道数量及对应采样时间应相同；
- 使用单次触发时，请将协同工作模式的 ADC 设置为序列 A 单次扫描或循环扫描；使用循环触发模式时，请设置为序列 A 单次扫描；
- 禁止多个 ADC 模块同时对一个模拟输入进行采样；
- 请严格按照用户手册协同模式相关部分的说明，来设置采样时间（ADC_SSTR 寄存器）和协同模式控制寄存器 ADC_SYNCCR 的 SYNCPLY。
- 请禁止序列 B，以免打乱同步。

例程 adc_13_adc1_adc2_sync 定义了这四种模式，实现了这四种模式的基本配置和应用。

```
/*
 * SYNC_SINGLE_SERIAL:
 * ADC1 and ADC2 only work once after being triggered.
 * Mode AdcMode_SAOnc and AdcMode_SAContinuous are valid.
 * ADC2 start after ADC1 N PCLK4 cycles.
 */
#define SYNC_SINGLE_SERIAL      (0u)

/*
 * SYNC_SINGLE_PARALLEL:
 * ADC1 and ADC2 only work once after being triggered.
 * Mode AdcMode_SAOnc and AdcMode_SAContinuous are valid.
 * ADC1 and ADC2 start at the same time.
 * ADC1 and ADC2 CAN NOT select the same ADC pin.
 */
#define SYNC_SINGLE_PARALLEL    (2u)

/*
```

```

* SYNC_CONTINUOUS_SERIAL:
* ADC1 and ADC2 are always working after being triggered.
* Mode AdcMode_SAOnc is valid.
* ADC2 start after ADC1 N PCLK4 cycles.
* ADC1 and ADC2 CAN NOT select the same ADC pin.
*/
#define SYNC_CONTINUOUS_SERIAL    (4u)

/*
* SYNC_CONTINUOUS_PARALLEL:
* ADC1 and ADC2 are always working after being triggered.
* Mode AdcMode_SAOnc is valid.
* ADC1 and ADC2 start at the same time.
*/
#define SYNC_CONTINUOUS_PARALLEL  (6u)

```

程序清单 3-1 协同模式定义

只需修改程序清单 3-1 中的宏定义 SYNC_MODE 即可指定具体的协同模式，如设置协同模式为循环延迟触发模式，见程序清单 3-2：

```

/* Select sync mode depending on your application. */
#define SYNC_MODE    SYNC_CONTINUOUS_SERIAL

```

程序清单 3-2 将协同模式配置为循环延迟触发

3.11.1 单次并行触发模式

3.11.1.1 说明

该模式下，ADC1 序列 A 的触发条件同时触发处于协同工作模式的所有 ADC 模块，如图 3-8。ADC1 序列 A 的触发条件只触发处于协同工作模式的 ADC 模块一次，这些 ADC 模块在采样转换一次后，是否停止，视其序列 A 扫描模式而定。

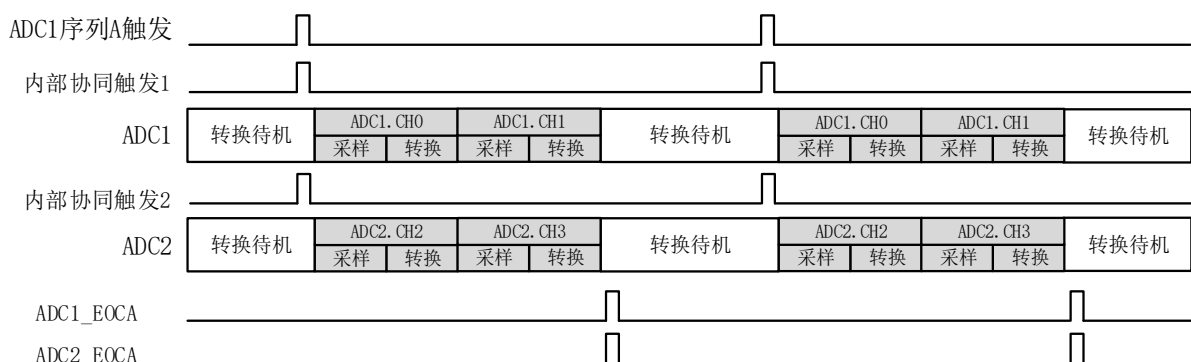


图 3-8 单次并行触发模式

3.11.1.2 应用

并行触发的特点是，处于协同工作模式的 ADC 能同时对各自的模拟输入进行采样转换。

例如，测量单相或三相瞬时电功率并绘制其曲线： $P_n(t) = U_n(t) \times I_n(t)$ 。在这种情况下，应同时测量电压和电流，然后计算瞬时功率，即电压 $U_n(t)$ 与电流 $I_n(t)$ 的乘积。

要测量单相电功率，可将 ADC1 的一个通道和 ADC2 的一个通道配合使用，一个通道测量电压，一个通道测量电流，如图 3-9；要测量三相电功率，可将 ADC1 的三个通道和 ADC2 的三个通道配合使用，三个通道测量电压，三个通道测量电流，如图 3-10。

单相情况：	采样	转换	
ADC1		CHx: U(t)	CHx: U(t) CHx: U(t)
ADC2		CHy: I(t)	CHy: I(t) CHy: I(t)
$P(t) = U(t) \times I(t) = CHx \times Chy$			

图 3-9 测量单相电功率

三相情况：	采样	转换	
ADC1		CHu: Uph1(t)	CHv: Uph2(t) CHw: Uph3(t)
ADC2		CHx: Iph1(t)	CHy: Iph2(t) CHz: Iph3(t)
$P_{ph1}(t) = U_{ph1}(t) \times I_{ph1}(t) = CHu \times CHx$ $P_{ph2}(t) = U_{ph2}(t) \times I_{ph2}(t) = CHv \times CHy$ $P_{ph3}(t) = U_{ph3}(t) \times I_{ph3}(t) = CHw \times CHz$			

图 3-10 测量三相电功率

如果只需要测量某一时刻的瞬时功率，使用单次并行触发模式即可；如果要连续测量，可结合定时器使用，用定时器定时触发该协同模式，或结合序列 A 连续扫描模式，或用循环并行触发模式，具体可根据需要的扫描频率等需求选择合适的解决方法。

3.11.2 单次延迟触发模式

3.11.2.1 说明

ADC1 序列 A 的触发条件触发 ADC1 后，经过设定的延迟后触发 ADC2 启动采样转换，如图 3-11。ADC1 序列 A 的触发条件只触发处于协同工作模式的 ADC 模块一次，这些 ADC 模块在采样转换一次后，是否停止，视其序列 A 扫描模式而定。

3.11.2.2 应用

结合序列 A 连续扫描模式，可实现对同一模拟输入的高频采样。例如，如果要转换的信号的最大频率为 2.5MHz，则采样率应该大于或等于该信号频率的二倍（符合香农采样定理）。由于一个 ADC 的最大采样率为 2.5MSPS，达不到采样定理的要求。此时，可通过这种方式将采样率提高到 5MSPS。关键设置如下：

1. 设置 ADC 时钟为 60MHz;
2. ADC1 和 ADC2 设置扫描模式为序列 A 连续扫描;
3. ADC1 和 ADC2 配置同一个模拟输入, 并设置相同的采样周期数 `ADC_SSTR = 11`;
4. 配置同步模式为单次延迟触发, 并设置同步延迟时间 `ADC_SYNCCCR.SYNCDLY = 12`;

由此可得工作示意图如图 3-12。



图 3-12 ADC 协同模式实现 5MSPS 采样率

由图 3-12 可知，采样率为： $60 / \text{ADC_SYNCCR.SYNCDLY} = 5\text{MSPS}$ ，该频率满足应用需求。用户还可以配置更多不同的 ADC 时钟频率，以实现更多的采样率。将例程

adc_13_adc1_adc2_sync 中的协同模式设置为单次延迟触发模式（见程序清单 3-3），就实现了对模拟输入引脚 ADC12_IN4 以 5MSPS 的采样率进行采样。

```
/* Select sync mode depending on your application. */
#define SYNC_MODE          SYNC_SINGLE_SERIAL
```

程序清单 3-3 将协同模式配置为单次延迟触发

注意：

- 不要为了更高的采样率而缩减采样时间 ADC_SSSTR，采样时间过短可能会使转换结果的误差超出误差值的设计范围。

建议：强烈建议使用 DMA 而非中断，以免数据丢失。

3.11.3 循环并行触发模式

3.11.3.1 说明

该模式下，ADC1 序列 A 的触发条件同时触发处于协同工作模式的所有 ADC 模块。若设置为单次并行触发，那么所有 ADC 模块在转换一次后就停止转换（如果扫描模式是序列 A 单次扫描）；若设置为循环并行触发，在 ADC1 序列 A 的触发条件同时触发处于协同工作模式的所有 ADC 模块后，每经过指定延迟之后，所有 ADC 模块会再次同时触发转换，如此循环，直到用户主动软件停止 ADC1 模块或禁止协同模式。循环并行触发模式工作示意图如图 3-13。

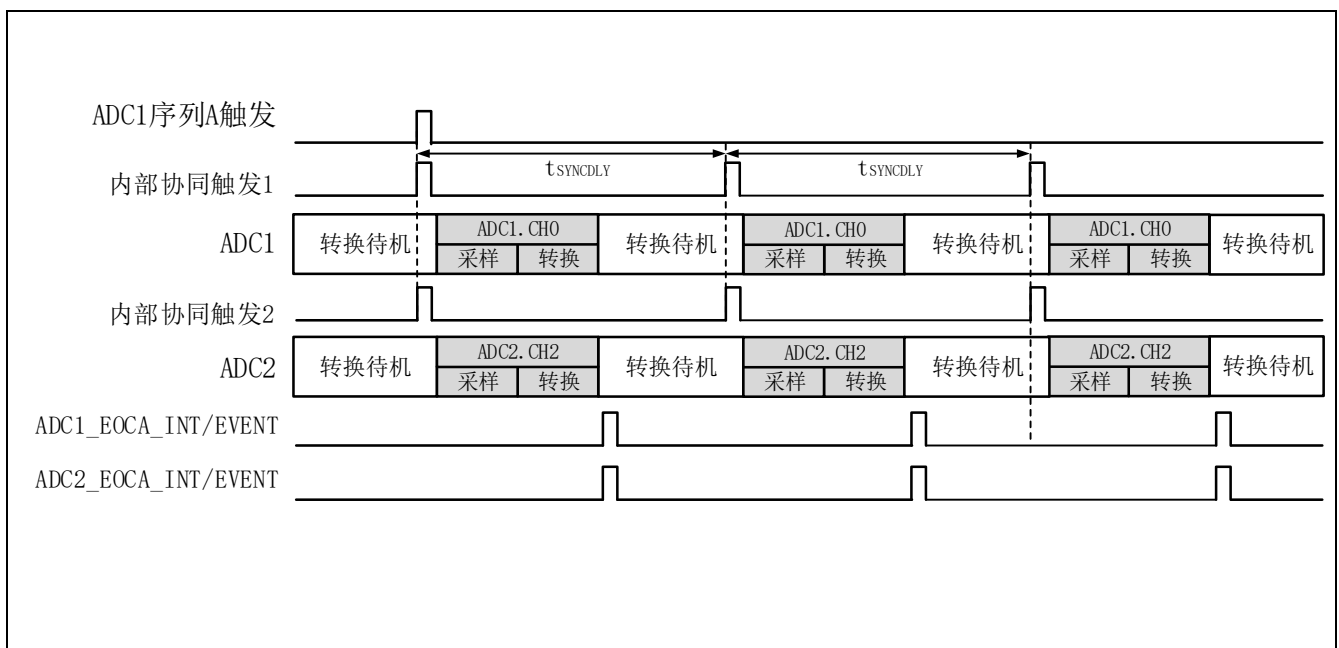


图 3-13 循环并行触发模式

3.11.3.2 应用

如果需要连续测量某模拟信号，可使用该模式，比如前面的测量电功率的应用。

3.11.4 循环延迟触发模式

3.11.4.1 说明

ADC1 序列 A 的触发条件触发 ADC1 之后，每经过设定的延迟后，依次循环不断触发 ADC2、ADC1、ADC2.....，直到用户主动软件停止 ADC1 模块或禁止协同模式。该模式工作示意图如图 3-14。

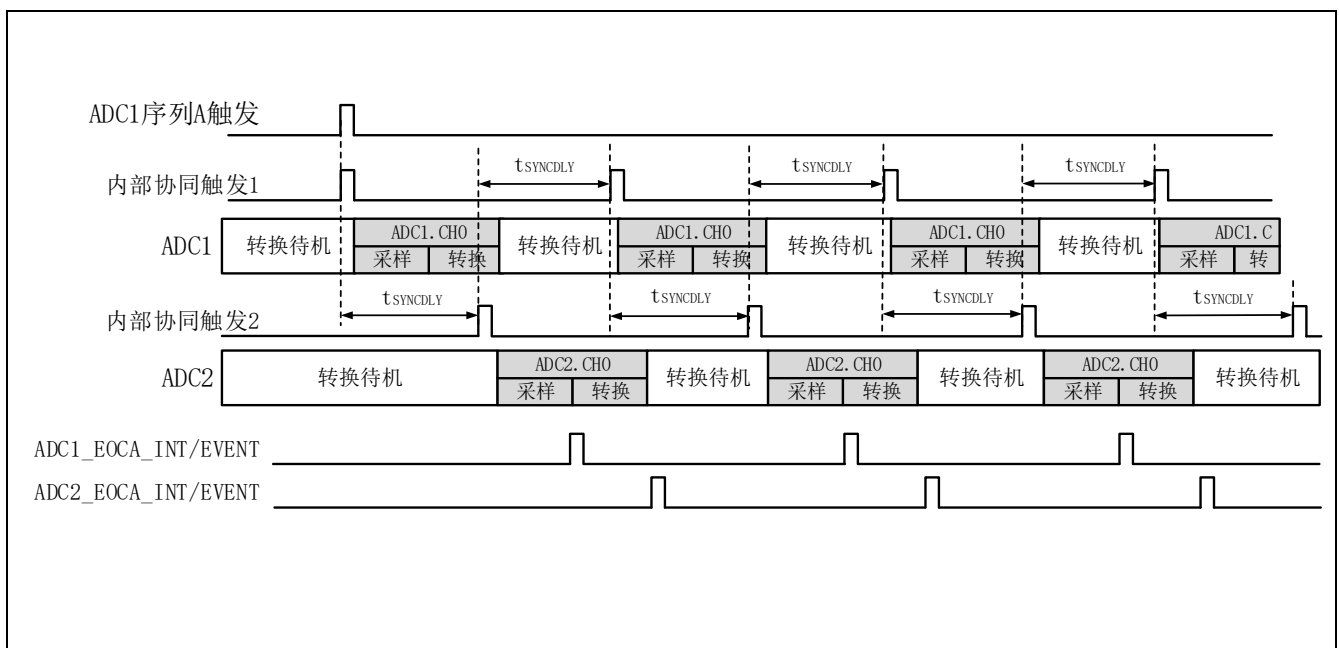


图 3-14 循环延迟触发模式

3.11.4.2 应用

该模式适用于一些，需要对模拟输入进行循环交替采样而对采样率要求不是很高的应用。这里简单介绍下循环延迟触发模式采样率的计算方法。具体要求请参考用户手册 17.3.8 节以及 17.4.16 节寄存器 ADC_SYNCCCR 的 SYNCDELAY 设置要求。假设如下条件：

1. ADC 时钟为 60MHz；
2. 采样时间 ADC_SSTR 设置为 11；
3. 由 17.3.8 节和 17.4.16 节相关要求，SYNCDELAY 可设置为 17，示意图如图 3-15。

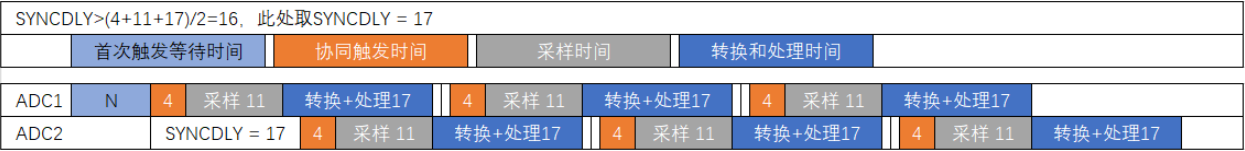


图 3-15 循环延迟触发模式采样示意图
 其中时间 N，只在首次触发时才有，具体数值由触发条件决定，细节请参考用户手册 17.3.8 节。由以上可得采样率为： $60 / \text{SYNCDLY} = 3.5\text{MSPS}$ 。

4 例程讲解

4.1 基本流程

ADC 应用程序流程可简单用图 4-1 表示。

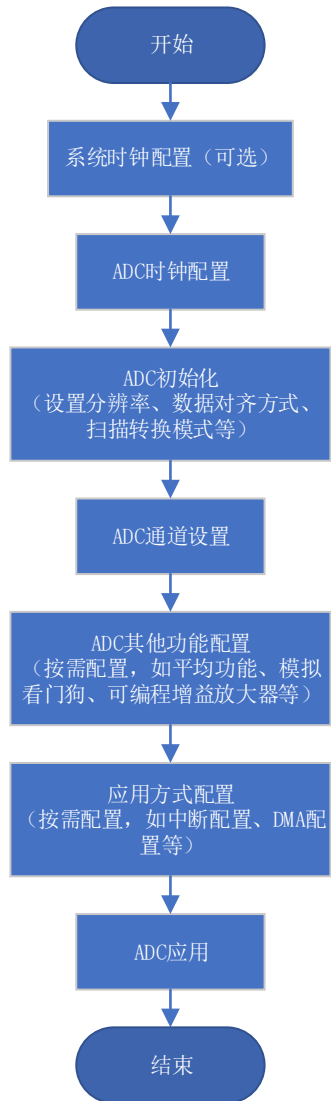


图 4-1 ADC 应用流程图

4.2 应用程序源码说明

用户可以根据 4.1 节的流程图编写自己的 ADC 应用程序，也可以通过华大半导体的网站下载 HC32F460 系列 MCU 的设备驱动库（Device Driver Library, DDL），参考其中的 ADC 例程。这里以其中的例程 adc_13_adc1_adc2_sync，对 ADC 应用程序的源码做一个简单的说明。

4.2.1 应用程序基本结构

ADC 应用程序例程基本结构如程序清单 4-1。

```
int32_t main(void)
{
    /* Config a new system clock. */
    SystemClockConfig();

    /* Config ADCs. */
    AdcConfig();

    /*
     * Config indicate pins.
     * Its purpose is simply to indicate the sampling rate.
     */
    IndicatePinConfig();

    /****** Configuration end, application start *****/
    AOS_SW_Trigger();

    while (1u)
    {
        // Your application code.
    }
}
```

程序清单 4-1 ADC 应用程序例程基本结构

说明：

1. SystemClockConfig(): 配置新的系统时钟，当前应用程序配置为 168MHz；
2. AdcConfig(): ADC 的所有配置，包括初始化、通道设置等；
3. IndicatePinConfig(): 当前应用程序利用 ADC1 和 ADC2 协同工作模式，实现了对同一模拟输入引脚 5MSPS 采样率的高频采样，在实现过程中，需要测试采样率是否达标，故配置了两个引脚来输出方波（引脚在 ADC 每完成一次采样转换，都会反转电平），以检测采样率，PB5 指示 ADC1 采样状态，PD8 指示 ADC2 采样状态；
4. AOS_SW_Trigger(): 协同工作模式只能由 ADC1 的触发外部触发引脚或内部触发事件触发，当前应用程序配置为 AOS 软件触发。

4.2.2 重要源码讲解

下面对详细介绍该应用程序中 ADC 的配置。当前应用程序的 ADC 配置包括时钟配置、初始化配置、通道配置、触发源配置和同步模式配置，如程序清单 4-2。

```
static void AdcConfig(void)
{
    AdcClockConfig();
    AdcInitConfig();
    AdcChannelConfig();
    AdcTriggerConfig();
    AdcSyncConfig();
    AdcDmaConfig();
}
```

程序清单 4-2 ADC 配置

说明：

1. **AdcClockConfig()**: 配置异步时钟 UPLL 为 ADC 时钟，即模拟电路时钟和数字接口时钟都采用 UPLL。当前配置 UPLL 为 60MHz，见程序清单 4-3。

```
static void AdcClockConfig(void)
{
    stc_clk_xtal_cfg_t stcXtalCfg;
    stc_clk_upll_cfg_t stcUpllCfg;

    MEM_ZERO_STRUCT(stcXtalCfg);
    MEM_ZERO_STRUCT(stcUpllCfg);

    /* Use XTAL as UPLL source. */
    stcXtalCfg.enFastStartup = Enable;
    stcXtalCfg.enMode = ClkXtalModeOsc;
    stcXtalCfg.enDrv = ClkXtalLowDrv;
    CLK_XtalConfig(&stcXtalCfg);
    CLK_XtalCmd(Enable);

    /* Set UPLL out 240MHz. */
    stcUpllCfg.pllmDiv = 1u;
    /* upll = 8M(XTAL) / pllmDiv * plln */
    stcUpllCfg.plln = 30u;
    stcUpllCfg.PllpDiv = 16u;
    stcUpllCfg.PllqDiv = 16u;
    stcUpllCfg.PllrDiv = 4u;
    CLK_SetPllSource(ClkPllSrcXTAL);
    CLK_UpllConfig(&stcUpllCfg);
    CLK_UpllCmd(Enable);
    CLK_SetPeriClkSource(ClkPeriSrcUpll);
}
```

程序清单 4-3 配置异步时钟 UPLL 为 ADC 时钟

2. **AdcInitConfig()**: 协同模式用到了 ADC1 和 ADC2，该初始化配置程序同时初始化了 ADC1 和 ADC2，详见程序清单 4-4。

```
static void AdcInitConfig(void)
{
    stc_adc_init_t stcAdcInit;

    MEM_ZERO_STRUCT(stcAdcInit);
}
```

```

/* ADC1 and ADC2 use the same configuration in sync mode. */
stcAdcInit.enResolution = AdcResolution_12Bit;
stcAdcInit.enDataAlign = AdcDataAlign_Right;
stcAdcInit.enAutoClear = AdcClren_Disable;
stcAdcInit.enScanMode = SA_SCAN_MODE;

/* 1. Enable ADC1. */
ENABLE_ADC1();
/* 2. Initialize ADC1. */
ADC_Init(M4_ADC1, &stcAdcInit);

/* 1. Enable ADC2. */
ENABLE_ADC2();
/* 2. Initialize ADC2. */
ADC_Init(M4_ADC2, &stcAdcInit);
}

```

程序清单 4-4 ADC1 和 ADC2 初始化

3. **AdcChannelConfig()**: ADC 通道配置，包括模拟输入引脚模式设置（设置为模拟输入模式）、ADC 通道添加、通道采样时间设置、所属序列设置等。当前，ADC1 和 ADC2 设置的相同的外部输入引脚 ADC12_IN4，对应的 ADC1 通道为 ADC1_CH4，ADC2 通道为 ADC2_CH0；采样时间为 11 个周期；协同模式只支持序列 A，故两个 ADC 的扫描序列都配置为序列 A。配置详见程序清单 4-5。

```

static void AdcChannelConfig(void)
{
    stc_adc_ch_cfg_t stcChCfg;
    uint8_t au8Adc1SaSampTime[ADC1_SA_CHANNEL_COUNT] =
        ADC1_SA_CHANNEL_SAMPLE_TIME;
    uint8_t au8Adc2SaSampTime[ADC2_SA_CHANNEL_COUNT] =
        ADC2_SA_CHANNEL_SAMPLE_TIME;

    MEM_ZERO_STRUCT(stcChCfg);

    stcChCfg.u32Channel = ADC1_SA_CHANNEL;
    stcChCfg.u8Sequence = AdcSequence_A;
    stcChCfg.pu8SampTime = au8Adc1SaSampTime;
    /* 1. Set the ADC pin to analog mode. */
    AdcSetChannelPinMode(M4_ADC1, ADC1_CHANNEL, Pin_Mode_Ana);
    /* 2. Add ADC channel. */
    ADC_AddAdcChannel(M4_ADC1, &stcChCfg);

    stcChCfg.u32Channel = ADC2_SA_CHANNEL;
    stcChCfg.pu8SampTime = au8Adc2SaSampTime;
    /* 1. Set the ADC pin to analog mode. */
    /* Not need any more. ADC2 selects the same analog input with ADC1. */
    //AdcSetChannelPinMode(M4_ADC2, ADC2_CHANNEL, Pin_Mode_Ana);
    /* 2. Add ADC channel. */
    ADC_AddAdcChannel(M4_ADC2, &stcChCfg);
}

```

程序清单 4-5 ADC 通道配置

4. **AdcTriggerConfig()**: ADC 触发源设置。协同模式只能由 ADC1 的触发条件触发，当前配置为 AOS 软件触发，如程序清单 4-6 所示。

```
static void AdcTriggerConfig(void)
{
    stc_adc_trg_cfg_t stcTrgCfg;

    MEM_ZERO_STRUCT(stcTrgCfg);
    /*
     * If select an event(@ref en_event_src_t) to trigger ADC,
     * AOS must be enabled first.
     */
    ENABLE_AOS();

    /* Select EVT_AOS_STRG as ADC1 sequence A trigger source. */
    stcTrgCfg.u8Sequence = AdcSequence_A;
    stcTrgCfg.enTrgSel   = AdcTrgsel_TRGX0;
    stcTrgCfg.enInTrg0   = ADC_SYNC_TRG_EVENT;

    ADC_ConfigTriggerSrc(M4_ADC1, &stcTrgCfg);
    ADC_TriggerSrcCmd(M4_ADC1, AdcSequence_A, Enable);
}
```

程序清单 4-6 ADC 触发源设置

5. **AdcSyncConfig():** ADC 协同模式配置，主要配置协同模式的具体工作方式和延迟触发的时间。当前，协同模式具体工作方式为单次延迟触发，延迟触发时间设置为 12 个周期，见程序清单 4-7。

```
static void AdcSyncConfig(void)
{
    stc_adc_sync_cfg_t stcSync;

    MEM_ZERO_STRUCT(stcSync);
    stcSync.enMode   = (en_adc_sync_mode_t)SYNC_MODE;
    stcSync.u8TrgDelay = ADC_SYNC_DELAY_TIME;
    ADC_ConfigSync(&stcSync);
    ADC_SyncCmd(Enable);
}
```

程序清单 4-7 ADC 协同模式配置

4.2.3 程序执行现象

将工程 adc_13_adc1_adc2_sync 编译，下载至目标板全速运行，示波器探头连接 PB5 和 PD8 两个引脚，可观察到如图 4-2 所示的波形图。

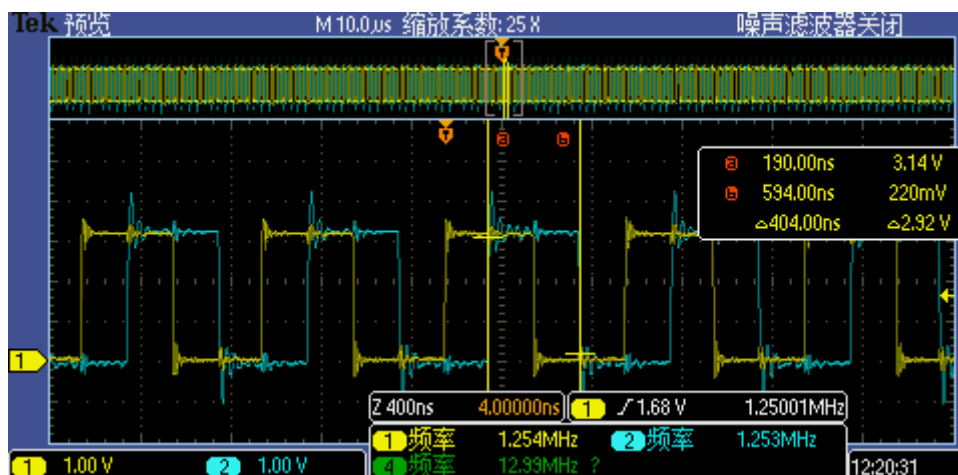


图 4-2 协同模式实现 5MSPS 采样率的执行情况

由于指示引脚在 ADC 每完成一次采样转换都会反转电平，所以由图可知，ADC1 和 ADC2 的采样频率均为 2.5MSPS，那么，总的采样频率即为 5MSPS。

5 总结

本应用笔记简要介绍了 HC32F460 系列 ADC 模块的各种功能以及可能的应用场景，并给出了 ADC 模块应用的基本流程，还提供了一种测试高频采样率的方法，在实际开发中，用户可根据具体应用场景按需配置和应用 ADC 模块。

6 版本信息 & 联系方式

日期	版本	修改记录
2019/3/15	Rev1.0	初版发布



如果您在购买与使用过程中有任何意见或建议，请随时与我们联系。

Email: mcu@hdsc.com.cn

网址: <http://www.hdsc.com.cn/mcu.htm>

通信地址: 上海市张江高科园区碧波路 572 弄 39 号

邮编: 201203

