

32-bit microcontrollers

HC 32F 460 Series Analog to Digital Converter ADC

Applicable objects

F Series	HC 32F460
----------	-----------

Table of Contents Table of Contents

1	Abstract	4
2	ADC Introduction	4
2.1	Function Introduction	4
2.2	Main Features	5
2.3	Pin Configuration	6
3	ADC Applications	7
3.1	Analog Input Pins and Channels	7
3.2	Sampling time and conversion time of analog inputs	8
3.3	Mode and Function	9
3.4	Sequence A Single Scan Mode	10
3.4.1	Description	10
3.4.2	Applications	10
3.5	Sequence A Continuous scan mode	11
3.5.1	Description	11
3.5.2	Applications	11
3.6	Dual sequence scanning mode	12
3.6.1	Description	12
3.6.2	Applications	14
3.7	Convert data averaging function	14
3.7.1	Description	14
3.7.2	Applications	14
3.8	Analog Watchdog	15
3.8.1	Description	15
3.8.2	Applications	15
3.9	Internal analog channels	16
3.9.1	Description	16
3.9.2	Applications	16
3.10	Programmable Gain Amplifier PGA	17
3.10.1	Description	17
3.10.2	Applications	17
3.11	Synergy Model	18

3.11.1	Single parallel trigger mode	19
3.11.2	Single delayed trigger mode	20
3.11.3	Cyclic parallel trigger mode.....	22
3.11.4	Cyclic delayed trigger mode	23
4	Routine explanation	25
4.1	Basic Process.....	25
4.2	Application source code description	26
4.2.1	Application basic structure.....	26
4.2.2	Important source code explanation	27
4.2.3	Program execution phenomenon.....	30
5	Summary	31
6	Version Information & Contact.....	32

2.2 Main Features

The ADC of HC32F460 series MCUs has the following main features:

1) High Performance

- Configurable for 12-, 10-, and 8-bit resolution
- The frequency ratio between the peripheral clock (digital clock) PCLK4 and the A/D converter clock ADCLK can be selected as follows:
- $PCLK4 : ADCLK = 1 : 1, 2 : 1, 4 : 1, 8 : 1, 1 : 2, 1 : 4$
- ADCLK can be selected as a PLL clock asynchronous to the system clock HCLK, where $PCLK4 : ADCLK = 1:1$
- Sampling rate: 2.5MSPS ($PCLK4 = ADCLK = 60MHz$, 12-bit resolution, sampling 11 cycles)
- Sampling time can be set independently for each channel
- Independent data register for each channel
- Data register configurable data alignment
- Continuous multiple conversion averaging function
- Analog watchdog to monitor conversion results
- ADC module can be set to stop when not in use

2) Analog input channels

- Up to 16 external analog input sampling channels
- 1 internal reference voltage / 8bit DAC output detection channel
- Up to 16 external analog input pins, which can be freely mapped to sample channels

3) Conversion start conditions

- Software starts to convert (only sequence A is supported)
- Peripheral event triggered start conversion (supports Sequence A and Sequence B)
- External pin triggered start conversion (supports Sequence A and Sequence B)

4) Conversion Mode

- Sequence A Single Scan
- Sequence A Continuous Scan
- Dual sequence scanning
- Synergy Model

5) Interrupt and event signal output

- Sequence A End-of-scan interrupt and event ADC_EOCA
- Sequence A End-of-scan interrupt and event ADC_EOCB
- Analog watchdog channel comparison interrupt and event ADC_CHCMP, sequence comparison interrupt and event ADC_SEQCMP
- All 4 event outputs above can initiate DMA

2.3 Pin Configuration

ADC1 of HC32F460 series MCU has 17 sampling channels and supports up to 16 external analog input pins, channels 0~15 can be freely mapped with external analog input pins, and channel 16 is used for internal reference voltage/8bitDAC output detection. ADC2 has 9 sampling channels and supports up to 8 external analog input pins, channels 0 to 7 can be freely mapped with external analog input pins, and channel 8 is used for internal reference voltage/8bitDAC output detection.

3 ADC Applications

3.1 Analog Input Pins and Channels

The configuration of the analog input pins, etc. of the ADC module of the HC32F460 series MCU is shown in Table 3-1.

Proj ects		Unit 1 (ADC1)	Unit 2 (ADC2)
Power supply		VCCA	
		VSSA/VREFL	
Base voltage		VREFH	
Analog Channel	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	Internal analog channel (reference voltage/8bitDAC output)
	CH9	ADC12_IN9	-
	CH10	ADC12_IN10	-
	CH11	ADC12_IN11	-
	CH12	ADC1_IN12	-
	CH13	ADC1_IN13	-
	CH14	ADC1_IN14	-
	CH15	ADC1_IN15	-
	CH16	Internal analog channel (reference voltage/8bitDAC output)	-
PGA		ADC1_IN0~3, ADC12_IN4~7. Any 1 channel of 8bitDAC_1 output	-

Proj ects		Unit 1 (ADC1)	Module 2 (ADC2)
Hardware trigger source	External Pins	ADTRG1	ADTRG2
	Around the film	IN_TRG10	IN_TRG20
		IN_TRG11	IN_TRG21

Table 3-1 ADC Pins and Channels

By default, CH0 (channel 0) of ADC1 corresponds to analog input pin ADC1_IN0, CH1 corresponds to ADC1_IN1, and CH16 is an internal analog channel that can only be used to detect internal reference voltage, 8bitDAC1 or 8bitDAC2. i.e., by default bit0 of ADC1 channel selection register ADC1_CHSELRA0 for sequence A (or ADC1_CHSELRB0 for sequence B) is set to 1, that is, the analog input pin ADC1_IN0 is selected, and bit1 is set to 1, that is, the analog input pin ADC1_IN1 is selected; the channel selection register ADC1_CHSELRA1 for sequence A (or ADC1_CHSELRB0 for sequence B) is set to 1, that is, the analog input pin ADC1_IN1 is selected. CHSELRA1 of sequence A (or ADC1_CHSELRB1 of sequence B), bit0 is set to 1, which means the internal analog input is selected for detecting the internal reference voltage, 8bitDAC1 or 8bitDAC2.

However, the ADC modules of the HC32F460 series feature free mapping of analog input pins to channels (except for the channels used to detect internal analog inputs) to meet different application requirements of users. For example, pin ADC12_IN10 can be mapped to one channel of ADC1 (e.g. CH0, not to CH16), or to multiple channels at the same time (e.g. CH0, CH2 and CH3).

The default correspondence between ADC2 channels and pins and channel remapping is similar to that of ADC1.

For channel remapping, the firmware routine `adc_11_channel_remap` gives its specific usage.

3.2 Sampling time and conversion time of analog inputs

For more details about ADC time, please refer to the user's manual, "Analog Input Sampling Time and Conversion Time" section, register `ADC_SSTR` and ADC electrical characteristics section, and set the sampling time strictly according to the manual. Especially when sampling multiple channels, if the sampling time of a channel is set too small, it may

cause coupling between adjacent channels (e.g. CH0, CH5, CH7 in sequence A, then CH0 and CH5, CH5 and CH7 are adjacent channels) through the sampling capacitance, and the conversion result will be inaccurate.

3.3 Modes and functions

The channels of the ADC can be configured as Sequence A or Sequence B. Sequence A and Sequence B can be individually set with different trigger sources. There are four scanning modes for both sequences:

- Sequence A Single scan;
- Sequence A continuous scan;
- Single scan for sequence A and single scan for sequence B;
- Sequence A continuous scan, sequence B single scan.

Each channel can also set the average function, which can calculate the average value of conversion after scanning the set number of times continuously and save the average value into the data register; analog watchdog AWD compares the conversion result after the channel conversion, and can generate channel comparison interrupt and event ADC_CHCMP, and generate sequence comparison interrupt and event ADC_SEQCMP according to the comparison result of each channel after the whole sequence scan; programmable gain amplifier PGA can amplify the analog input signal before conversion; ADC1 and ADC2 can convert simultaneously or consecutively in co-working mode; analog input pins can be free-imaged with ADC channels. SEQCMP; programmable gain amplifier PGA to amplify the analog input signal before conversion; ADC1 and ADC2 can be converted simultaneously or continuously alternately in co-working mode; analog input pins can be free-imaged with the ADC channels

The shot, combined with the synergy mode, enables high frequency scanning of the specified analog input.

3.4 Sequence A Single Scan Mode

3.4.1 Description

In this mode, the ADC performs a single scan of a single or multiple channels and stops when the conversion is complete, as illustrated in Figure 3-1.

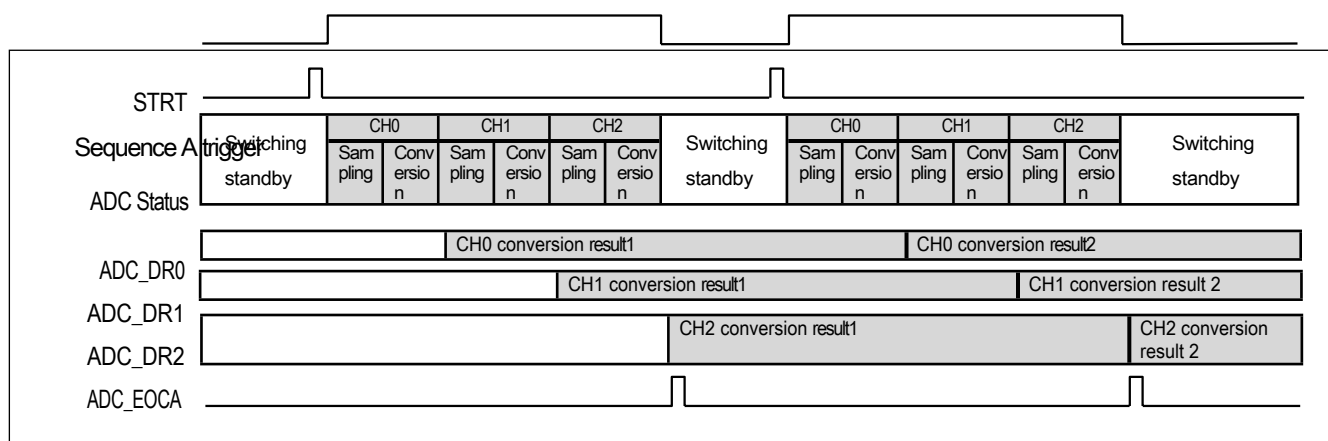


Figure 3-1 ADC Single Scan Mode

3.4.2 Applications

This mode can be set for single or multiple channels. When multiple channels are set, multiple channels can be scanned sequentially, and these channels can be set with different sampling times, so that the user does not have to stop the ADC in the middle of the scan, i.e., the next channel can be rescanned with different sampling times, avoiding additional CPU load and heavy software development.

This mode is the simplest ADC mode with flexible application mode. For example, before the system starts, this mode can be used to detect some status information of the system, such as voltage, pressure, temperature, etc., to determine whether the system can start normally; in the system operation, this mode can be used to detect the system status on demand to obtain the real-time status of the system.

The application routine `adc_01_sa_base` gives the specific usage of this mode.

3.5 Sequence A Continuous scan mode

3.5.1 Description

Continuous scan mode allows continuous scanning of a single channel or multiple channels, as shown in Figure 3-2. Continuous scan mode allows the ADC to work in the background. Therefore, the ADC can continuously (cyclically) scan channels without any CPU intervention. In addition, DMA can be used in continuous scan mode, thus reducing the CPU load.

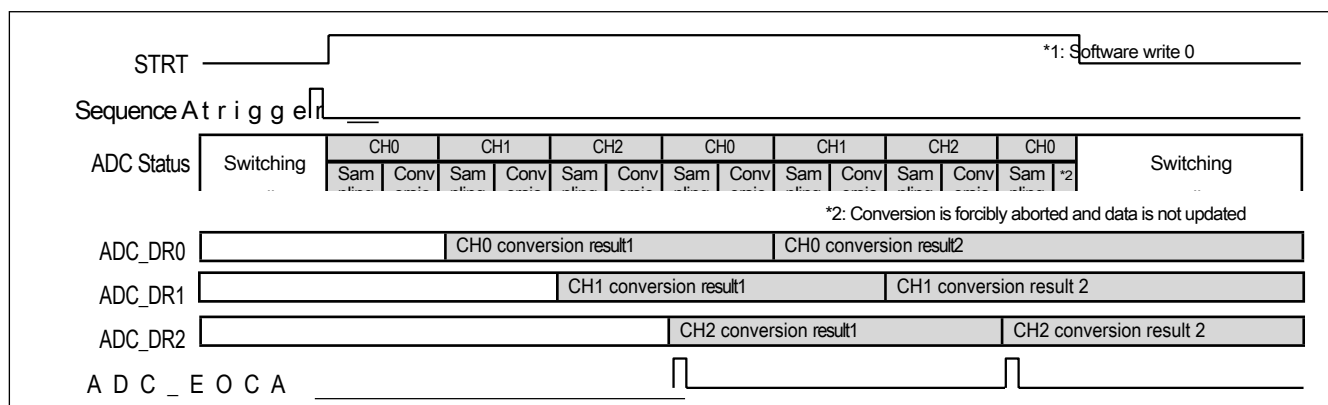


Figure 3-2 ADC Continuous Scan Mode

3.5.2 Applications

This mode, when set up as a single channel, can be used for applications such as monitoring battery voltage, measuring and adjusting oven temperatures. When used to adjust the oven temperature, the system will read the temperature and compare it to the user set temperature. When the oven temperature reaches the desired temperature, power is turned off to the heating resistor.

When multiple channels are set up, it is similar to multi-channel single scan mode, except that instead of stopping scanning after the last channel of the sequence is completed, scanning starts again from the first channel and continues in an infinite loop. The multi-channel continuous scan mode can be used to monitor multiple voltages and temperatures in a multi-battery charger. The system reads the voltage and temperature of each battery during the charging process. When the voltage or temperature reaches a maximum value, the corresponding battery is disconnected from the charger.

The application routine `adc_01_sa_base` has the settings for this mode and a simple application method.

3.6 Dual sequence scanning mode

3.6.1 Description

The two modes of "Single Scan for Sequence A, Single Scan for Sequence B" and "Continuous Scan for Sequence A, Single Scan for Sequence B" are combined here to introduce the Dual Sequence Scan mode. The Dual Sequence Scan mode only adds the Sequence B scan to the first two modes. In Dual Sequence Scan mode, Sequence B must be triggered by external pins or internal events, and software start is not valid for Sequence B. Sequence A can be scanned by software start, or by external pins or internal events. Sequence B has a higher priority than Sequence A. Its competition with Sequence A is shown in Table 3-2.

A/D conversion	Trigger signal generation	Processing	
		ADC_CR1.RSCHSEL=0	ADC_CR1.RSCHSEL=1
Sequence A conversion process	Sequence A trigger	Invalid trigger signal	
	Sequence B trigger	1) The conversion of sequence A is interrupted and sequence B is started Conversion 2) After the conversion of sequence B is completed, sequence A continues the conversion from the interrupted channel	1) The conversion of sequence A is interrupted and sequence B is started Conversion 2) After the conversion of sequence B is completed, sequence A is reconverted from the first channel
Sequence B conversion process	Sequence A trigger	After the conversion of all channels of sequence B is completed, start the conversion of sequence A	
	Sequence B trigger	Invalid trigger signal	
AD idle, sequence A, B triggered at the same time		Sequence B is started first, and after all channels are converted, sequence A conversion is started	

Table 3-2 Competition between Sequence A and Sequence B

When ADC_CR1.RSCHSEL is configured to 0, when sequence A is interrupted, the scan continues from the interrupted channel when it resumes, as shown in Figure 3-3.

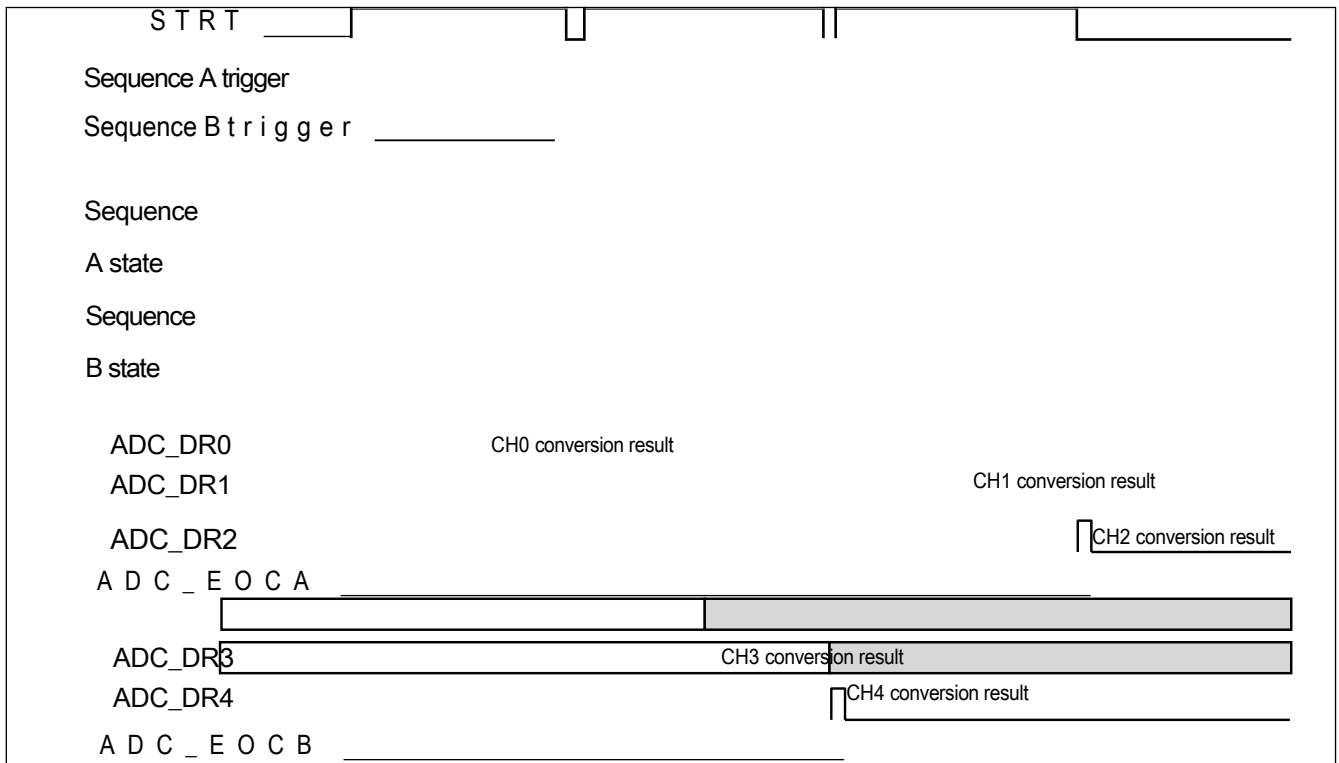


Figure 3-3 Sequence A Resume Scan from Interrupted Channel

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。

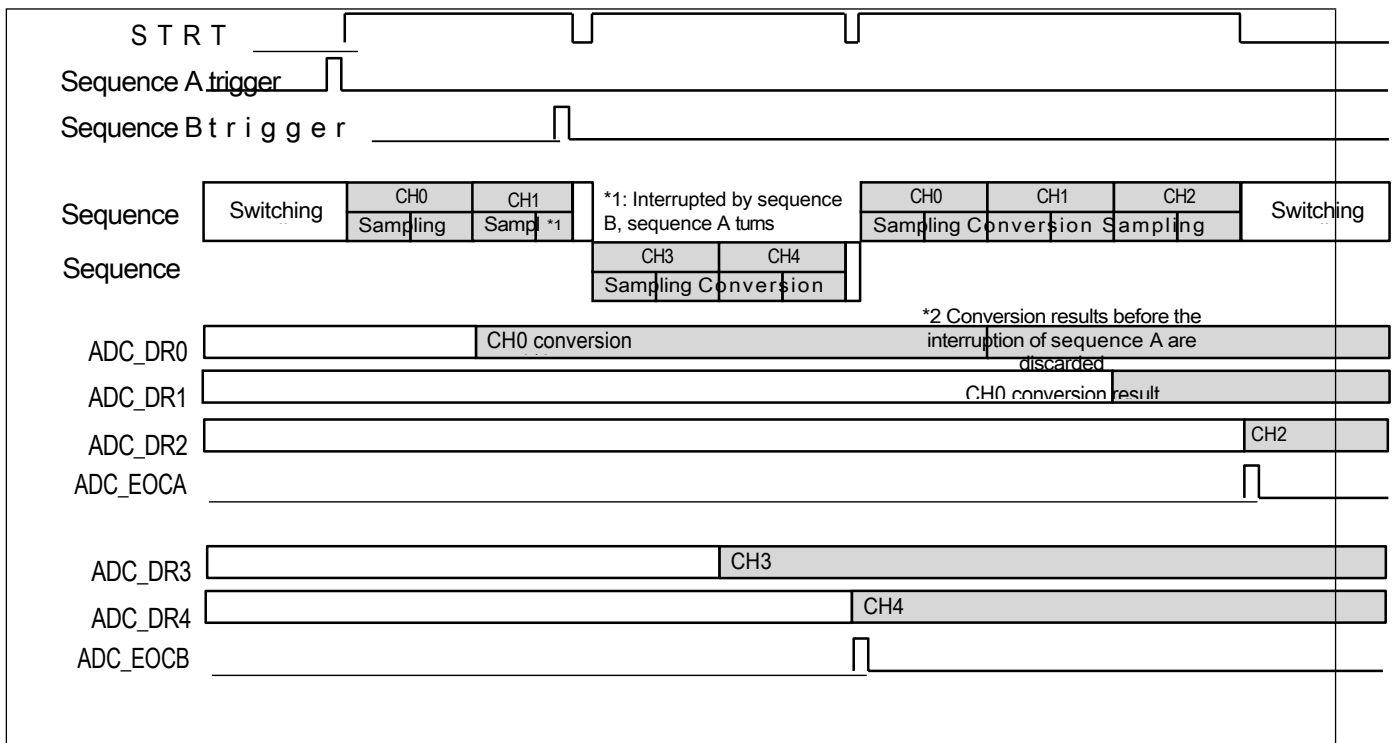


Figure 3-4 Sequence A resumes scanning from the first channel of this sequence

Cautio
n:

- The same channel cannot be selected for Sequence A and Sequence B. The same trigger source cannot be selected

Switching standby	CH0		CH1		*1: Interrupted by sequence B, conversion strong The system is suspended and the data is not updated	CH1		CH2		Switching standby
	Sam pling	Conv ersio n	Sam pling	*1		Sam pling	Conv ersio n	Sam pling	Conv ersio n	

3.6.2 Applications

Channels requiring real-time response (higher priority) scanning can be added to the application in the first two modes by configuring them as sequences

B. The routine `adc_04_sa_sb_event_trigger` implements the basic usage of double sequence scanning.

3.7 Convert data averaging function

3.7.1 Description

The conversion data averaging function can be set to average the conversion results after 2, 4, 8, 16, 32, 64, 128 or 256 consecutive scans, and then save them to the data register. This function removes a certain amount of noise component, making the results more accurate. The advantage of this function is that it can improve the accuracy of the ADC without any hardware changes, but the disadvantage is that it reduces the conversion speed and frequency (equivalent to reducing the effective sampling rate).

3.7.2 Applications

For different applications, a different number of consecutive scans can be set, which depends on the required accuracy, minimum conversion speed, etc. Application notes do not provide a separate routine for this function, there is a way to configure this function in the routine `adc_01_sa_base`.

3.8 Analog Watchdog

3.8.1 Description

The analog watchdog of HC32F460 series MCU can be configured as upper/lower limit comparison or interval comparison, as shown in Figure 3-5.

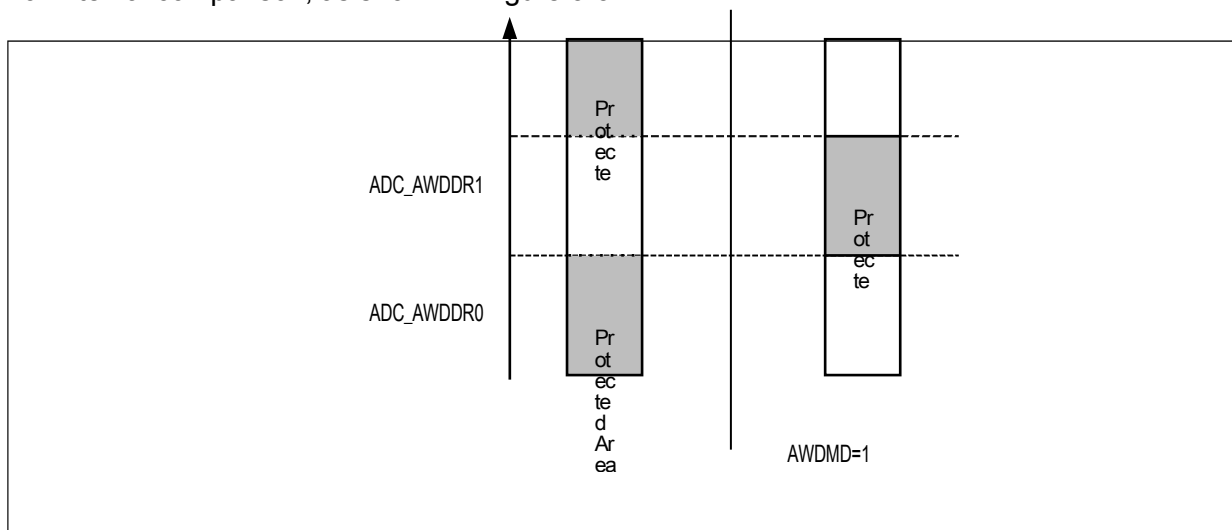


Figure 3-5 Analog Watchdog Comparison Conditions

The user can pre-set the comparison conditions and the corresponding upper and lower limits or intervals. At the end of the channel conversion, an analog watchdog evaluates the conversion results

If the comparison condition is met, the channel comparison interrupt and event ADC_CHCMP will be generated; at the end of the whole sequence scan, the sequence comparison interrupt and event ADC_SEQCMP will be generated according to the comparison result of each channel; each channel that enables the analog watchdog will generate an interrupt and event ADC_CHCMP as long as its conversion result meets the comparison condition; each sequence

After the condition is met, only one interrupt and event ADC_SEQCMP are generated for a sequence. i.e., an ADC module can generate multiple interrupts and events ADC_SEQCMP after a scan round, up to two (since there are at most two sequences) interrupts and events ADC_SEQCMP.) interrupts and events ADC_SEQCMP.

Caution:

- Using both ADC_CHCMP interrupts and ADC_SEQCMP interrupts is not recommended.

3.8.2 Applications

In some control systems, the range of signals such as voltage, pressure, and temperature

needs to be strictly monitored. Using an analog watchdog can quickly detect abnormal conditions of these signals and make corresponding countermeasures to ensure equipment safety. The routine `adc_08_sa_sb_awd_base` gives the configuration and basic application method of the analog watchdog; however, usually, the interrupt usage of the analog watchdog is more efficient and commonly used. The routine `adc_09_sa_sb_awd_interrupt` gives the interrupt configuration and usage of the analog watchdog.

3.9 Internal analog channels

3.9.1 Description

Both ADC1 and ADC2 have a channel for detecting internal analog inputs, channel 16 and channel 8, respectively, to detect three selectable internal analog input quantities, internal reference voltage, 8bitDAC1 output or 8bitDAC2 output, as shown in Figure 3-6.

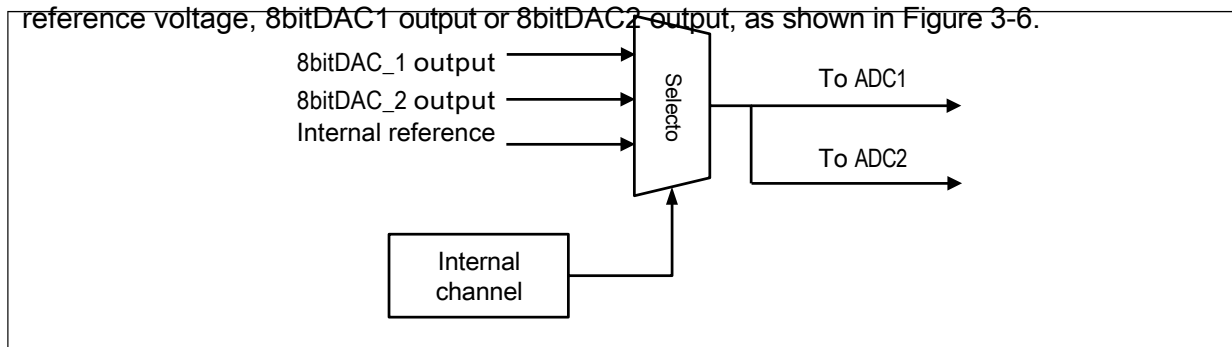


Figure 3-6 Internal Analog Channel Selection

Caution:

n:

- Only one of the ADC1 and ADC2 internal detection channels can be selected to detect one of the three internal analog inputs, and the ADC1 and ADC2 internal detection channels cannot be used simultaneously.

3.9.2 Applications

In some systems, the reference voltage of the ADC may be unstable for some reasons, so that the actual voltage value of the analog input is not known. In this case, the internal detection channel of ADC1 or ADC2 can be used to detect the internal reference voltage (1.1V when the system voltage is normal) to invert the current reference voltage of the ADC, so as to know the current actual voltage value of the analog input. The specific implementation is as follows:

1. The ADC value of the internal reference voltage measured at a known reference voltage VREF1 is VAL1;
2. With the operation of the system, the reference voltage may drop as the voltage of the supply source (such as the battery) decreases, at which time the ADC value of the internal reference voltage is measured as VAL2, and the reference voltage is set to VREF2 at this time;

3. Since the internal reference voltage is constant, we have: $VAL1 \times VREF1 = VAL2 \times VREF2$; $VREF2 = (VAL1 \times VREF1) / VAL2$.

From this, the current reference voltage is known as VREF2 and it is easy to get the actual voltage of the analog input. The example `adc_10_internal_channel` shows the various configurations and simple usage of the internal channel.

3.10 Programmable Gain Amplifier PGA

3.10.1 Description

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。

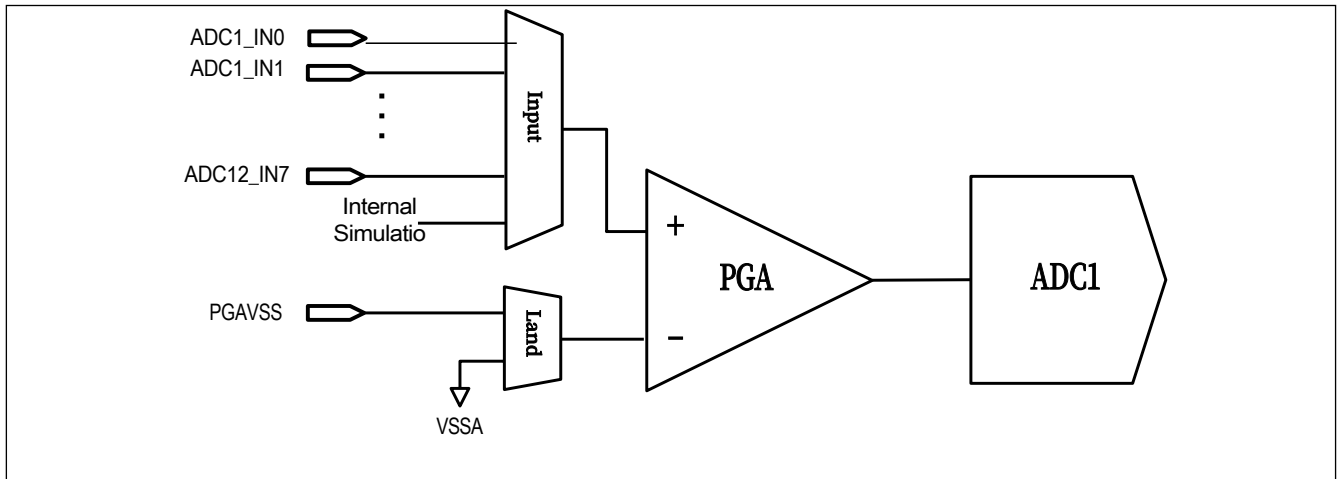


Figure 3-7 PGA working diagram

Caution:

- Only ADC1 supports PGAs;
- The PGA channel corresponds directly to the analog input pins, and its corresponding pin-mapped channel must be mapped by ADC1 Sequence A or Sequence B. The channel selection register of B is selected in order to amplify this analog input.

3.10.2 Applications

The user can select the appropriate amplification for the analog input according to the actual application scenario. The analog input voltage and amplification must meet the following conditions:

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

Where V_{CCA} is the analog supply voltage, Gain is the amplification, V_I is the analog input voltage, please refer to the user manual for details

PGA related section. For the configuration and usage of the PGA, please refer to the routine `adc_12_adc1_pga`.

3.11 Synergy Model

The HC32F460 series MCUs have two ADC modules that can be used in ADC co-working mode. In co-working mode, the conversion start can only be triggered by the ADC1 trigger source and software start is invalid.

The synergy mode can be configured in the following four synergy modes:

- Single parallel trigger mode
- Single delayed trigger mode
- Cyclic parallel trigger mode
- Cyclic Delayed

Trigger Mode Notes:

- (a) The collaboration mode can only be configured by ADC1;
- ADCs set to work in co-operative mode should have the same configuration (scan mode, resolution, data alignment, etc.) as far as possible; the specific channels need not be the same, but the number of channels and corresponding sampling times should be the same;
- (a) When using single trigger, set the ADC of the co-working mode to sequence A single scan or cyclic scan; when using cyclic trigger mode, set it to sequence A single scan;
- Prohibit multiple ADC modules from sampling an analog input simultaneously;
- Please strictly follow the instructions in the Synergy Mode related section of the user manual to set the sampling time (ADC_SSTR register) and SYNCDELAY of the Synergy Mode Control Register ADC_SYNCRR.
- Please disable sequence B to avoid disrupting the synchronization.

The routine `adc_13_adc1_adc2_sync` defines these four modes and implements the basic configuration and application of these four modes.

```
/*
 * SYNC_SINGLE_SERIAL.
 * ADC1 and ADC2 only work once after being triggered.
 * Mode AdcMode_SAOOnce 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_SAOOnce 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_SAOnce 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_SAOnce is valid.
* ADC1 and ADC2 start at the same time.
*/
#define SYNC_CONTINUOUS_PARALLEL (6u)

```

Procedure List 3-1 Synergy Mode Definition

Simply modify the macro definition SYNC_MODE in Program Listing 3-1 to specify the specific synergy mode, e.g. set the synergy mode to cyclic delay trigger mode, see Program Listing 3-2:

```

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

```

Program Listing 3-2 Configuring Synergy Mode as Cyclic Delayed Trigger

3.11.1 Single parallel trigger mode

3.11.1.1 Description

In this mode, the trigger condition of ADC1 sequence A triggers all ADC modules in the cooperative mode at the same time, as shown in Figure

3-8. The ADC1 Sequence A trigger condition triggers only once for ADC modules in co-working mode, and whether these ADC modules stop after one sample conversion depends on their Sequence A scan mode.

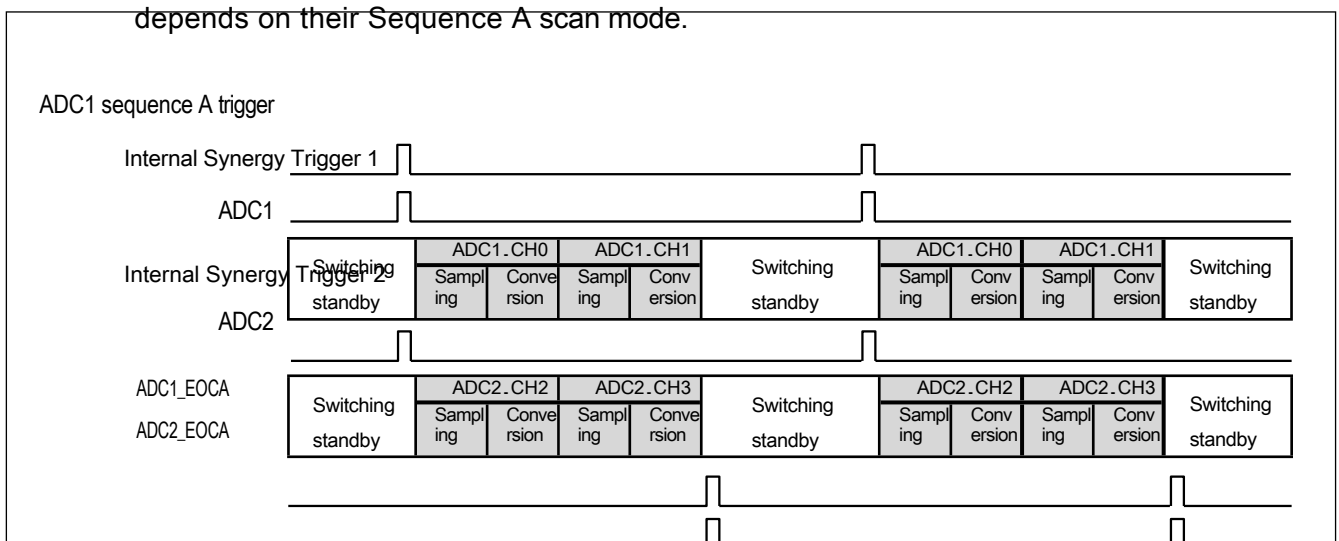


Figure 3-8 Single parallel trigger mode

3.11.1.2 Applications

Parallel triggering is characterized by the fact that ADCs in co-operative mode can simultaneously sample and convert their respective analog inputs.

For example, single-phase or three-phase instantaneous electrical power is measured and plotted: $P_n(t) = U_n(t) \times I_n(t)$. In this case, both voltage and current should be measured, and then the instantaneous power should be calculated as the product of voltage $U_n(t)$ and current $I_n(t)$.

To measure single-phase power, one channel of ADC1 and one channel of ADC2 can be used together, with one channel measuring voltage and one channel measuring current, as in Figure 3-9; to measure three-phase power, three channels of ADC1 and three channels of ADC2 can be used together, with three channels measuring voltage and three channels measuring current, as in Figure 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)
$P(t) = U(t) \times I(t) = CHx \times CHy$						
ADC1		CHu: Uph1(t)		CHv: Uph2(t)		CHw: Uph3(t)
ADC2		CHx: Iph1(t)		CHy: Iph2(t)		CHz: Iph3(t)
$P_{ph1}(t) = U_{ph1}(t) \times I_{ph1}(t) = CHu \times CHx$ $P_{ph2}(t) = U_{ph2}(t) \times I_{ph2}(t) = CHv \times CHy$ $P_{ph3}(t) = U_{ph3}(t) \times I_{ph3}(t) = CHw \times CHz$						

Figure 3-9 Measurement of single-phase electrical power

Figure 3-10 Measuring three-phase electrical power

If you only need to measure the instantaneous power at a certain moment, you can use the single parallel trigger mode; if you want to measure continuously, you can use it in combination with a timer, use the timer to trigger the synergy mode at regular intervals, or combine the sequence A continuous scanning mode, or use the cyclic parallel trigger mode, depending on the required scanning frequency and other needs to choose the appropriate solution.

3.11.2 Single delayed trigger mode

3.11.2.1 Description

The trigger condition of ADC1 sequence A triggers ADC1 and then triggers ADC2 to start sampling conversion after a set delay, as shown in Figure 3-11. The trigger condition of ADC1 sequence A triggers ADC modules in co-working mode only once, and whether these ADC modules stop after sampling conversion once depends on their sequence A scan mode.

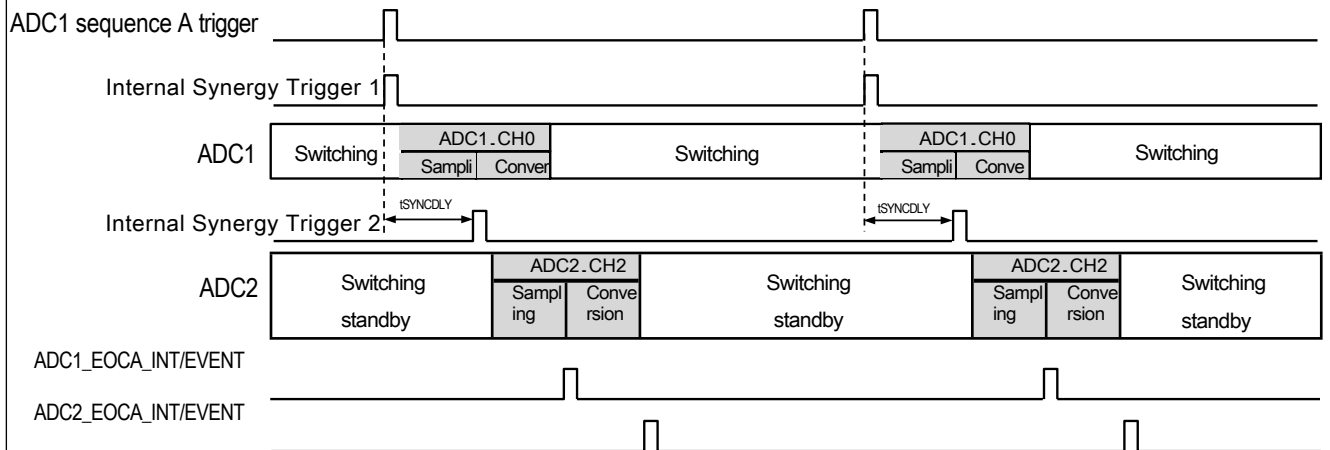
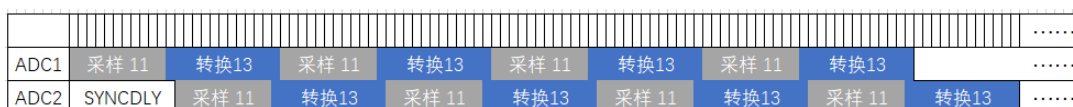


Figure 3-11 Single delayed trigger mode

3.11.2.2 Applications

In combination with the Sequence A continuous scan mode, high frequency sampling of the same analog input is possible. For example, if the maximum frequency of the signal to be converted is 2.5 MHz, the sampling rate should be greater than or equal to two times the frequency of the signal (in accordance with Shannon's sampling theorem). Since the maximum sampling rate of an ADC is 2.5MSPS, it does not meet the requirements of the sampling theorem. In this case, the sampling rate can be increased to 5MSPS in this way. the key settings are as follows:

1. Setting the ADC clock to 60 MHz;
2. ADC1 and ADC2 set the scan mode to Sequence A continuous scan;
3. ADC1 and ADC2 are configured with the same analog input and set the same number of sampling periods $ADC_SSTR = 11$;
4. Configure the synchronization mode as single delay trigger and set the synchronization delay time $ADC_SYNCCR.SYNCDLY = 12$; this results in the working schematic as shown



in Figure 3-12.

Figure 3-12 ADC Co-Mode for 5MSPS Sample Rate

As seen in Figure 3-12, the sampling rate is $60 / \text{ADC_SYNCCR.SYNCDLY} = 5\text{MSPS}$, which meets the application requirements. The user can also configure more different ADC clock frequencies to achieve more sample rates. Set the routine

The synergy mode in adc_13_adc1_adc2_sync is set to single delay trigger mode (see program listing 3-3), which enables sampling of the analog input pin ADC12_IN4 at a sample rate of 5MSPS.

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

Program Listing 3-3 Configuring Synergy Mode as a Single Delayed Trigger

Caution

n:

- Do not reduce the sampling time ADC_SSTR for higher sampling rate, too short sampling time may make the conversion result error out of the design range of error value.

Recommendation: It is highly recommended to use DMA instead of interrupt to avoid data loss.

3.11.3 Cyclic parallel trigger mode

3.11.3.1 Description

In this mode, the trigger condition of ADC1 sequence A triggers all ADC modules in co-operative mode at the same time. If set to single parallel trigger, then all ADC modules stop converting after one conversion (if the scan mode is a single scan of sequence A); if set to cyclic parallel trigger, after the trigger condition of ADC1 sequence A triggers all ADC modules in co-working mode at the same time, after each specified delay, all ADC modules will trigger the conversion again at the same time, and so on, until the user until the ADC1 module is stopped or the cooperative mode is disabled by the user's active software. The operation diagram of the cyclic parallel trigger mode is shown in Figure 3-13.

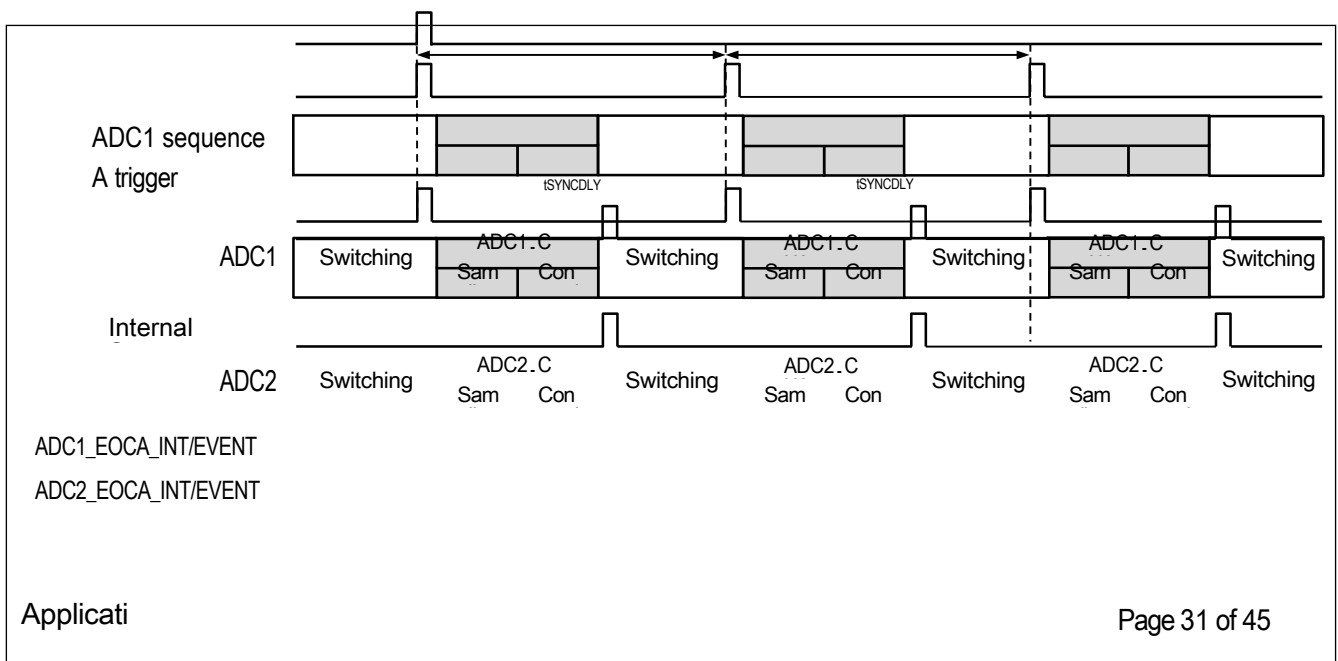


Figure 3-13 Cyclic parallel trigger mode

3.11.3.2 Applications

This mode can be used if a certain analog signal needs to be measured continuously, as in the previous application for measuring electrical power.

3.11.4 Cyclic delayed trigger mode

3.11.4.1 Description

The trigger condition of ADC1 sequence A triggers ADC1, and then after each set delay, it continues to trigger in a cycle

ADC2, ADC1, ADC2 until the ADC1 module is stopped by user-initiated software or the cooperative mode is disabled. The schematic of this mode operation is shown in

Figure 3-14.

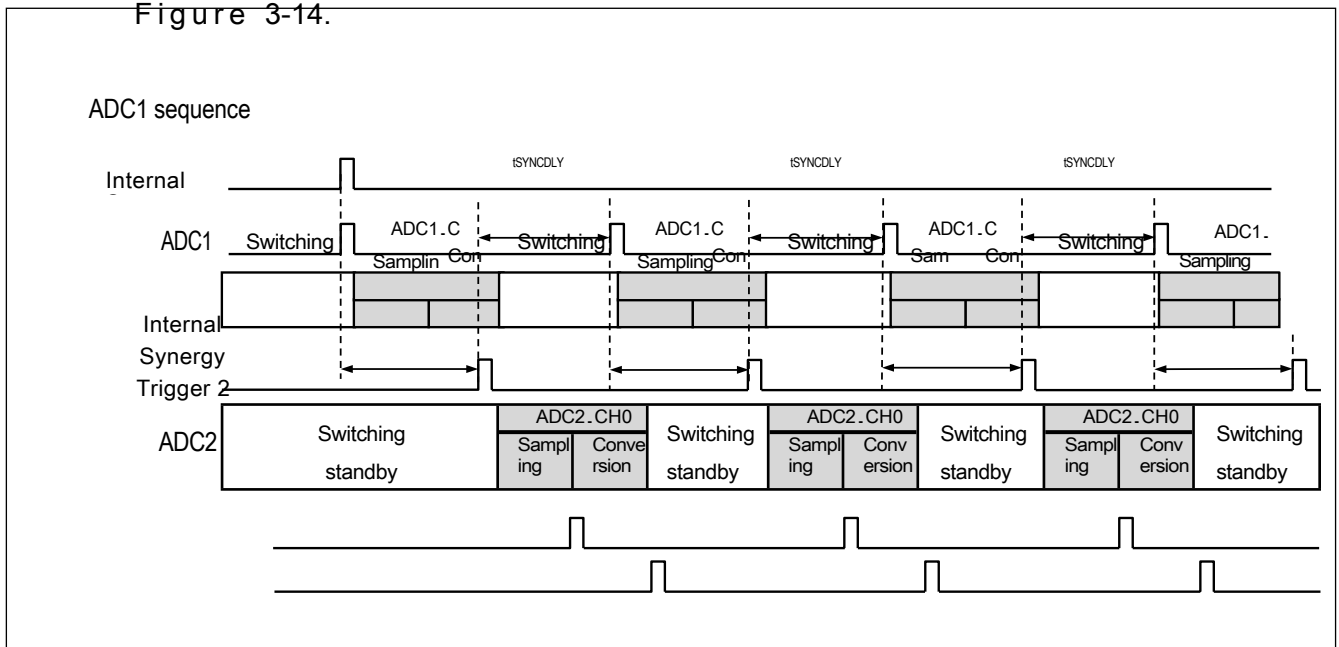


Figure 3-14 Cyclic delay trigger mode

3.11.4.2 Appli

catio

ns

This mode is suitable for some applications that require cyclic alternate sampling of analog inputs and do not require high sampling rates. Here is a brief description of how to calculate the sample rate for cyclic delay trigger mode. Please refer to Section 17.3.8 of the User's Manual and Section 17.4.16 of the User's Manual for the SYNCDELAY setting of register

ADC_SYNCRR. Assume the following conditions:

1. ADC clock of 60 MHz;
2. The sample time ADC_SSTR is set to 11;
3. As required by Section 17.3.8 and Section 17.4.16, SYNCCLY can be set to 17, as shown in Figure 3-15.



Figure 3-15 Sampling schematic of cyclic delayed trigger mode

where the time N, only at the first trigger, the specific value is determined by the trigger conditions, please refer to the user manual 17.3.8 for details

section. From the above, the sampling rate is: $60 / \text{SYNCDLY} = 3.5\text{MSPS}$.

4 Routine explanation

4.1 Basic Process

The ADC application flow can be simply represented in Figure 4-1.

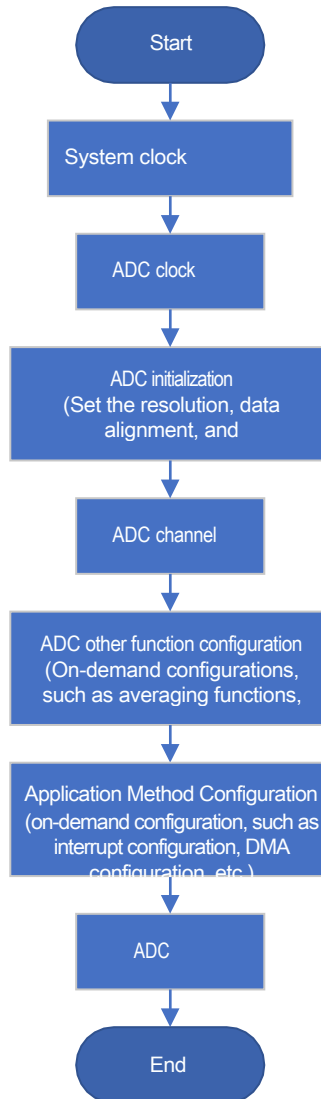


Figure 4-1 ADC Application Flow Chart

4.2 Application source code description

Users can write their own ADC applications according to the flowchart in Section 4.1, or download the Device Driver Library (DDL) of HC32F460 series MCU through the website of UW Semiconductors and refer to the ADC routines in it. Here is a brief explanation of the source code of ADC application with one of the routines `adc_13_adc1_adc2_sync`.

4.2.1 Basic application structure

The basic structure of the ADC application routine is shown in Program Listing 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,    applicationstart *****/
    AOS_SW_Trigger().

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

Program Listing 4-1 Basic Structure of ADC Application Routines

Descri ption:

1. `SystemClockConfig()`: configures the new system clock, currently configured to 168 MHz for the application;
2. `AdcConfig()`: all configurations of the ADC, including initialization, channel settings, etc;
3. `IndicatePinConfig()`: the current application uses ADC1 and ADC2 co-working mode to achieve a high frequency sampling of the same analog input pin with 5MSPS sampling rate, in the process of implementation, it is necessary to test whether the sampling rate is up to standard, so two pins are configured to output a square wave (the pins will invert the level for every sampling conversion completed by ADC) to detect sampling rate, PB5 indicates the ADC1 sampling status and PD8 indicates the ADC2 sampling status;

4. AOS_SW_Trigger(): Co-working mode can only be triggered by ADC1's trigger external trigger pin or internal trigger event, the current application is configured for AOS software trigger.

4.2.2 Important source code explanation

The configuration of the ADC in this application is described in detail below. The ADC configuration for the current application includes clock configuration, initialization configuration, channel configuration, trigger source configuration, and synchronization mode configuration, as shown in Listing 4-2.

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

Program Listing 4-2 ADC Configuration

Description:

1. `AdcClockConfig()`: Configure the asynchronous clock UPLL as ADC clock, i.e. both analog circuit clock and digital interface clock use UPLL. the current configuration UPLL is 60MHz, see program listing 4-3.

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

    MEM_ZERO_STRUCT(stcXtalCfg);
    MEM_ZERO_STRUCT(stcUpIICfg);

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

    /* Set UPLL out 240MHz.
    */ stcUpIICfg.pllMDiv = 1u;
    /* upll = 8M(XTAL) / pllMDiv * plln */
    stcUpIICfg.plln = 30u;
    stcUpIICfg.PllpDiv = 16u;
    stcUpIICfg.PllqDiv = 16u;
    stcUpIICfg.PllrDiv = 4u; CLK_
    SetPllSource(ClkPllSrcXTAL);
    CLK_UpIICfg(&stcUpIICfg);
    CLK_UpIICmd(Enable);
    CLK_SetPeriClkSource(ClkP erisrcUpIIR);
}
```

Program Listing 4-3 Configuring the Asynchronous Clock UPLL for ADC Clocks

2. `AdcInitConfig()`: ADC1 and ADC2 are used in cooperative mode, this initialization configuration program initializes both ADC1 and ADC2, see program listing 4-4 for

details.

```
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 = AdcClen_Disable;
stcAdcInit.enScanMode = SA_SCAN_MODE; stcAdcInit.
Right; stcAdcInit.enAutoClear = AdcClen_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).
}

```

Program Listing 4-4 Initialization of ADC1 and ADC2

3. `AdcChannelConfig()`: ADC channel configuration, including analog input pin mode setting (set to analog input mode), ADC channel addition, channel sampling time setting, belonging sequence setting, etc. Currently, ADC1 and ADC2 set the same external input pin ADC12_IN4, the corresponding ADC1 channel is ADC1_CH4, ADC2 channel is ADC2_CH0; the sampling time is 11 cycles; the cooperative mode only supports sequence A, so the scan sequence of both ADCs is configured as sequence A. See program list 4-5 for details.

```

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).
}

```

Program Listing 4-5 ADC Channel Configuration

4. `AdcTriggerConfig()`: ADC trigger source setting. Co-mode can only be triggered by ADC1 trigger condition, currently configured as AOS software trigger, as shown in program listing 4-6.

```
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);
}
```

Program List 4-6 ADC Trigger Source Setting

5. `AdcSyncConfig()`: ADC synergy mode configuration, mainly configuring the specific working mode of synergy mode and the delayed trigger of Time. Currently, the synergy mode works as a single delayed trigger, and the delayed trigger time is set to 12 cycles, see program list 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);
}
```

Program Listing 4-7 ADC Co-Mode Configuration

4.2.3 Program execution phenomenon

The project `adc_13_adc1_adc2_sync` is compiled, downloaded to the target board and run at full speed. The oscilloscope probe is connected to pins PB5 and PD8, and the waveforms shown in Figure 4-2 can be observed.

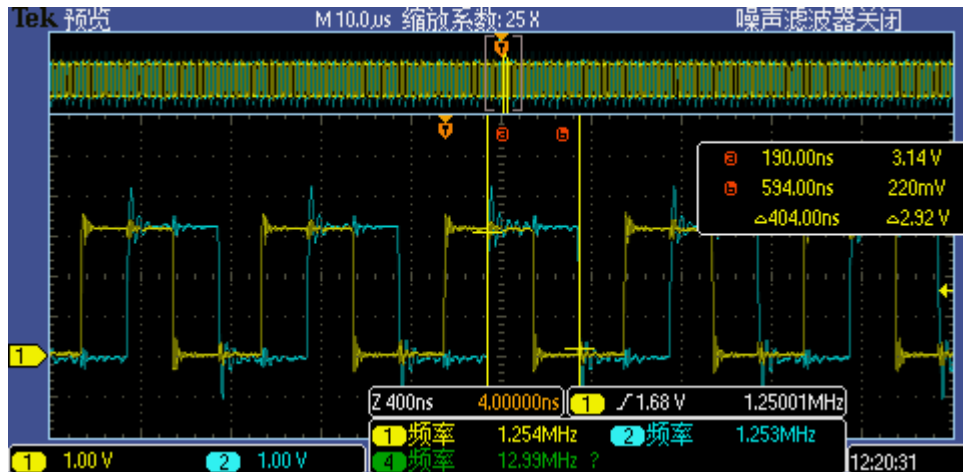


Figure 4-2 Execution of synergy mode to achieve 5MSPS sampling rate

Since the indicator pin reverses the level at every sample conversion completed by the ADC, the sampling frequency of both ADC1 and ADC2 is 2.5MSPS, so the total sampling frequency is 5MSPS, as can be seen from the figure.

5 Summary

This application note briefly introduces various functions of HC32F460 series ADC modules and possible application scenarios, and gives the basic flow of ADC module application, and also provides a method to test high frequency sampling rate, in the actual development, users can configure and apply ADC modules according to the specific application scenarios as needed.

6 Version Information & Contact

Date	Versions	Modify records
2019/3/15	Rev1.0	Initial Release
2020/8/26	Rev1.1	Update supported models



If you have any comments or suggestions in the process of purchase and use, please feel free to contact us.

Email: mcu@hdsc.com.cn

Website: <http://www.hdsc.com.cn/mcu.htm>

Address: 10/F, Block A, 1867 Zhongke Road, Pudong

New Area, Shanghai, 201203, P.R. China

