Современные 32-разрядные ARM микроконтроллеры серии STM32: аналого-цифровой преобразователь

Олег Вальпа

В статье приведено описание аналого-цифрового преобразователя 32-разрядных ARM микроконтроллеров серии STM32 от компании STMicroelectronics. Рассмотрена архитектура и состав его регистров, а также приведены практические примеры программ.

ВВЕДЕНИЕ

Одним из наиболее важных и востребованных блоков для любого микроконтроллера является аналого-цифровой преобразователь (АЦП), именуемый в технической англоязычной литературе как Analog-to-Digital Converter (ADC).

Он позволяет преобразовывать аналоговые сигналы в цифровые значения, которые в дальнейшем могут обрабатываться процессором. Суть преобразования сводится к сравнению аналогового сигнала с опорным напряжением и формирование числового значения, соответствующего аналоговому сигналу в диапазоне разрядности АЦП. Например, 12-разрядный АЦП с опорным напряжением 3.3 В преобразует аналоговые сигналы от 0 до 3.3 В в диапазон цифровых значений от 0 до 4095 единиц с определенной периодичностью во времени.

С помощью АЦП микроконтроллер способен решать гораздо больше задач для автоматизации процессов. Кроме того, при наличии встроенного в микроконтроллер АЦП существенно упрощается схема разрабатываемого устройства и значительно уменьшается его стоимость. Как правило, кроме самого АЦП микроконтроллер имеет встроенный аналоговый мультиплексор, позволяющий увеличить количество аналоговых входов путем поочередной коммутации нескольких выводов микроконтроллера и последующего преобразования сигналов от этих выводов.

Микроконтроллеры серии STM32 [1] также имеют встроенный 12-разрядный АЦП последовательного приближения с тактовой частотой до 14 МГц. Некоторые модели STM32 имеют даже два или три блока АЦП.

АРХИТЕКТУРА И ФУНКЦИОНИРОВАНИЕ АЦП

Рассмотрим подробнее архитектуру и функционирование блока АЦП для STM32. Аналого-цифровое преобразование сигналов данным АЦП может осуществляться в одиночном режиме, непрерывном режиме, режиме сканирования или прерывистом режиме. Результат преобразования сохраняется в 16-разрядных регистрах АЦП с выравниванием данных по левому или по правому краю.

Встроенный АЦП микроконтроллеров серии STM32 обладает расширенными функциями. Основные характеристики этого АЦП:

- разрядность — 12 бит;
- минимальное время преобразования — 1 мкс;
- количество каналов — 16 внешних и 2 внутренних;
- количество значений времени преобразования для каждого канала — 8;
- возможность задания одиночного или непрерывного преобразования;
- автоматическая калибровка;
- наличие оконного компаратора;
- возможность запуска преобразования от внешних источников;
- работа с блоком прямого доступа к памяти (ПДП).

АЦП питается от источника с напряжением от 2.4 до 3.6 В. Источник опорного напряжения (ИОН) АЦП соединен либо внутренне с напряжением питания АЦП, либо со специальными внешними выводами, в зависимости от модели и количества выводов кристалла.

Структурная схема АЦП микроконтроллеров серии STM32 приведена на рисунке 1.

Как видно из рисунка, АЦП имеет 16 внешних аналоговых каналов и 2 внутренних, которые коммутируются с помощью мультиплексора входов. Внешние каналы подключены к выводам микроконтроллера. Перед использованием этих выводов в качестве аналоговых необходимо сконфигурировать их как аналоговые входы. Дополнительные два канала задействованы под внутренние сигналы. Один — для контроля внутреннего опорного напряжения, а второй — для датчика температуры, который расположен на кристалле.

Для данного АЦП введены такие специальные термины, как регулярные и инжектированные каналы. Суть этих понятий заключается в способе опроса каналов. Регулярные каналы опрашиваются и преобразуются с определенной периодичностью, то есть регулярностью. Инжектированные, то есть внедренные каналы, опрашиваются по специальному запросу от программы между регулярными опросами.

Если необходимо опрашивать периодически несколько каналов, то для АЦП можно сформировать список этих каналов и записать его в специальные регистры. После этого АЦП будет поочередно преобразовывать сигналы

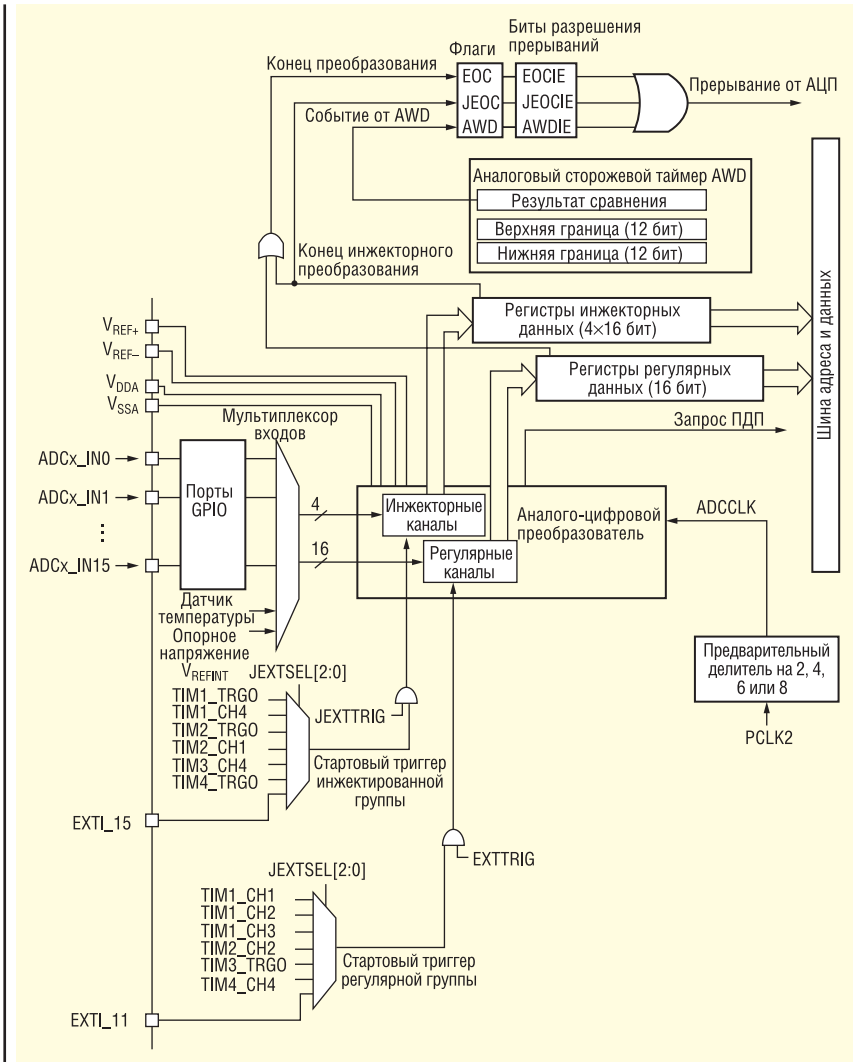

Рис. 1. Структурная схема АЦП микроконтроллеров STM32

из каналов данного списка последовательно друг за другом. Например, для поочередного измерения напряжения в каналах 3, 7, 5 и 1 необходимо записать эту последовательность в специальные регистры и запустить процесс преобразования АЦП. В результате данные каналы будут последовательно опрашиваться и преобразовываться, а результат этих преобразований записываться в регистры данных. При этом порядок следования каналов не имеет значения. Более того, один и тот же канал может опрашиваться несколько раз за цикл. Количество измерений в группе регулярных каналов может достигать 16. Допускается непрерывное измерение выбранных каналов, то есть по окончанию измерения автоматически запускается новый цикл.

Максимальное количество измерений в группе инжектированных каналов равно четырем. Если запустить измерение инжектированных каналов, то измерение регулярных каналов будет приостановлено. Затем будет выполнено измерение заданных инжектированных каналов и вновь восстановлено измерение каналов регулярной группы.

Рассмотрим пример того, как можно применить на практике использование регулярных и инжектированных каналов. Допустим, необходимо постоянно измерять напряжение на пяти выводах в определенной последовательности. Для этого необходимо включить эти каналы в регулярную группу и запустить преобразование. АЦП начнет последовательно опрашивать их и преобразовывать сигналы в цифровые значения. При необходимости по окончании преобразования будут формироваться прерывания. Таким образом, можно обрабатывать аналоговые сигналы и сохранять данные преобразования АЦП в автоматическом режиме. Если в это время возникает необходимость измерить температуру кристалла или аналоговый сигнал на каком-либо другом канале, то для того, чтобы не нарушать измерение в регулярном канале, можно запустить измерение инжектированного канала. При этом обработка регулярных каналов будет автоматически приостановлена на некоторое время, а после окончания измерения инжектированной группы — автоматически восстановлена.

Сохранение результата преобразования для четырех инжектированных каналов производится в соответствующих регистрах данных. АЦП имеет для каждого инжектированного канала отдельный регистр, то есть всего четыре регистра данных.

Однако для 18 регулярных каналов в АЦП имеется всего один регистр данных. С целью сохранения результатов преобразования для каждого канала можно по окончанию измерения в каждом канале формировать прерывание и переписывать данные из этого регистра в блок памяти процессора. Кроме того, можно воспользоваться специальным каналом ПДП, который позволит пересылать результаты каждого преобразования в заранее организованный в памяти буфер данных без участия процессора. Использование данного метода позволяет сократить частоту формирования прерываний до одного по завершении каждого цикла преобразования группы регулярных каналов. Кроме того, можно задействовать двойной буфер. Это позволит после заполнения АЦП первого буфера сформировать прерывание и приступить к обработке результатов процессором. В это время АЦП начнет заполнять с помощью ПДП второй буфер. Затем очередность поменяется и так далее.

В нижней части схемы, изображенной на рисунке, приведены источники, которые могут запускать процесс преобразования отдельно для регулярной и инжектированной группы каналов. Это могут быть сигналы от таймеров, внешний сигнал или два специальных разряда управляющих регистров, установив которые можно программно запустить преобразование в регулярной или инжектированной группе.

Отличительной особенностью данного АЦП является наличие в нем аналогового сторожевого таймера (AWD от англ. Analog Watch Dog), представляющего собой оконный компаратор. Он имеет в своем составе два регистра для задания границ сравнения. В регистр нижней границы и регистр верхней границы программно записы-

ваются соответствующие значения для контроля уровня измеряемого сигнала. Номер канала, к которому нужно подключить компаратор, записывается в специальный регистр. Если измеряемое напряжение заданного канала выйдет за указанные границы, то в АЦП будет установлен соответствующий флаг и сформируется запрос на прерывание.

Данный оконный компаратор может оказаться очень полезным для автоматического контроля определенного параметра. С его помощью можно, например, следить за температурой контроллера, во избежание его перегрева. Для этого можно регулярно, скажем, ежесекундно, измерять уровень сигнала от датчика температуры. Если он превысит заданный порог, будет необходимо предпринять определенные действия. Например, снизить тактовую частоту, сформировать предупреждающий сигнал и тому подобное. Оконный компаратор позволяет выполнять подобную процедуру без отвлечения процессора от выполнения основной программы, экономя при этом ресурсы процессора и памяти программ. Для запуска оконного компаратора необходимо выполнить его инициализацию путем записи уровней границ, включить контролируемый канал в список регулярных каналов и разрешить прерывание от него. Если уровень контролируемого сигнала выйдет за границы заданного диапазона, компаратор сработает и будет вызвана функция обработчика прерывания данного события.

АЦП способен формировать три сигнала прерывания: конец преобразования, конец преобразования инжектированной группы и сигнал от оконного компаратора.

ОПИСАНИЕ РЕГИСТРОВ АЦП

Блок АЦП включает в свой состав довольно большое количество регистров. Но это логично, учитывая его насыщенную функциональность и количество каналов для преобразования. Карта регистров АЦП представлена в таблице 1.

Все регистры можно сгруппировать по функциональному назначению.

В результате такой организации образуются следующие группы:

- регистр состояния ADC_SR — содержит биты, указывающие на состояние АЦП;
- регистры управления ADC_CR1 и ADC_CR2 — определяют режим работы АЦП;
- регистры ADC_SMPR1 и ADC_SMPR2 — задают время преобразования АЦП;
- регистры ADC_JOFR1...ADC_JOFR4 — определяют смещение данных в инжектированной группе каналов;
- регистры ADC_HTR и ADC_LTR — задают верхнюю и нижнюю границу для оконного компаратора;
- регистры ADC_SQR1...ADC_SQR3 — задают последовательность каналов регулярной группы;
- регистр ADC_JSQR — задает последовательность каналов инжектированной группы;
- регистры данных ADC_JDR1...ADC_JDR4 — содержат результат преобразования для регистров инжектированной группы каналов;
- регистр данных DR — содержит результат преобразования для регулярной группы каналов.

Рассмотрим назначение разрядов в каждой из представленных групп регистров АЦП подробнее.

Регистр ADC_SR содержит биты, которые устанавливаются аппаратно. Обнуляются они программно или при сбросе. Бит EOC сбрасывается автоматически после чтения регистра ADC_DR. Назначение бит следующее:

- бит 4 STRT устанавливается при старте преобразования регулярного канала;
- бит 3 JSTRT устанавливается при старте преобразования инжектированного канала;
- бит 2 JEOC указывает на окончание преобразования инжектированного канала;
- бит 1 EOC обозначает окончание преобразования регулярного или инжектированного канала;
- бит 0 AWD устанавливается при срабатывании оконного компаратора.

Регистр ADC_CR1 содержит следующие биты управления:

- бит 23 AWDEN управляет оконным компаратором регулярного канала;
- бит 22 JAWDEN управляет оконным компаратором инжектированного канала;
- биты 15...13 DISCNUM[2:0] определяют количество регулярных каналов, преобразование которых будет выполняться в прерывистом режиме, по приходу события внешнего запуска;
- бит 12 JDISCEN запрещает и разрешает прерывистый режим для инжектированных каналов;
- бит 11 DISCEN запрещает и разрешает прерывистый режим для регулярных каналов;
- бит 10 JAUTO запрещает и разрешает режим автоматического преобразования для инжектированных каналов;
- бит 9 AWDSGL разрешает работу оконного компаратора для всех каналов или для того канала, который определен битами AWDCH[4:0];
- бит 8 SCAN запрещает и разрешает режим сканирования каналов, заданных регистрами ADC_SQRx или ADC_JSQRx;
- бит 7 JEOCIE запрещает и разрешает прерывания по окончанию преобразования инжекционных каналов;
- бит 6 AWDIE запрещает и разрешает прерывания от оконного компаратора АЦП;
- бит 5 EOCIE запрещает и разрешает прерывания по окончанию преобразования в регулярной или инжектированной группе;
- биты 4...0 AWDCH[4:0] определяют номер канала, к которому подключен оконный компаратор.

Регистр ADC_CR2 содержит биты, имеющие следующее назначение:

- бит 23 TSVREFE подключает канал опорного напряжения и датчика температуры, расположенного на кристалле;
- бит 22 SWSTART запускает преобразование регулярных каналов;
- бит 21 JSWSTART запускает преобразование инжектированных каналов;
- бит 20 EXTTRIG запрещает и разрешает использование внешнего запуска для регулярных каналов;
- биты 19...17 EXTSEL[2:0] определяют источник, который будет запускать преобразование в регулярном канале (000=TIM1_CH1, 001=TIM1_CH2, 010=TIM1_CH3, 011=TIM2_CH2, 100=TIM3_TRGO, 101=TIM4_CH4, 110=EXTI_11, 111=SWSTART);
- бит 15 JEXTTRIG запрещает и разрешает использование внешнего запуска для инжектированных каналов;
- биты 14...12 JEXTSEL[2:0] определяют источник, который будет запускать преобразование в инжектированном канале (000=TIM1_TRGO, 001=TIM1_CH4, 010=TIM2_TRGO, 011=TIM2_CH1, 100=TIM3_CH4, 101=TIM4_TRGO, 110=EXTI_15, 111=JSWSTART);
- бит 11 ALIGN задает выравнивание результата преобразования в регистре данных по правому (0) или левому (1) краю;
- бит 8 DMA запрещает и разрешает использование блока ПДП (DMA);

Таблица 1. Карта регистров АЦП

Сдвиг	Регистр	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	ADC_SR	Резерв																											STRT	JSTRT	EOC	EOC	AWD
	Исх. значение																												0	0	0	0	*0*
0x04	ADC_SR1	Резерв								AWDEN	JAWDEN	Резерв						DISC MUM[2:0]			JDISCEN	DISCEN	JAUTO	AWD SGL	SCAN	JEOC IE	AWDIE	EOCIE	AWDCH[4:0]				
	Исх. значение									0	0							0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x08	ADC_SR2	Резерв								TSVREFE	SWSTART	JSWSTART	EXTTRIG	EXTSEL[2:0]			Резерв	JEXTTRIG	JEXTSEL[2:0]			ALIGN	Резерв		DMA	Резерв			RSTCAL	CAL	CONT	ADON
	Исх. значение									0	0	0	0	0	0	0		0	0	0	0	0			0				0	0	0	0
0x0C	ADC_SMPR1	SMPx_x																															
	Исх. значение	0	0	0	0	0	0		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x10	ADC_SMPR2	SMPx_x																															
	Исх. значение	0	0	0	0	0	0		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x14	ADC_JOFR1	Резерв																JOFFSET1[11:0]															
	Исх. значение																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x18	ADC_JOFR2	Резерв																JOFFSET2[11:0]															
	Исх. значение																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x1C	ADC_JOFR3	Резерв																JOFFSET3[11:0]															
	Исх. значение																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x20	ADC_JOFR4	Резерв																JOFFSET4[11:0]															
	Исх. значение																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x24	ADC_HTR	Резерв																HT[11:0]															
	Исх. значение																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x28	ADC_LTR	Резерв																LT[11:0]															
	Исх. значение																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x2C	ADC_SQR1	Резерв								L[3:0]				SQ16[4:0]					SQ15[4:0]					SQ14[4:0]					SQ13[4:0]			
	Исх. значение									0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x30	ADC_SQR2	Резерв	SQ12[4:0]					SQ11[4:0]						SQ10[4:0]					SQ9[4:0]					SQ8[4:0]					SQ7[4:0]			
	Исх. значение		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x34	ADC_SQR3	Резерв	SQ6[4:0]					SQ5[4:0]						SQ4[4:0]					SQ3[4:0]					SQ2[4:0]					SQ1[4:0]			
	Исх. значение		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x38	ADC_JSQR	Резерв										JL[1:0]		JSQ4[4:0]					JSQ3[4:0]					JSQ2[4:0]					JSQ1[4:0]			
	Исх. значение											0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x3C	ADC_JDR1	Резерв																JDATA[15:0]															
	Исх. значение																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x40	ADC_JDR2	Резерв																JDATA[15:0]															
	Исх. значение																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x44	ADC_JDR3	Резерв																JDATA[15:0]															
	Исх. значение																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x48	ADC_JDR4	Резерв																JDATA[15:0]															
	Исх. значение																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x4C	ADC_DR	Резерв																DATA[15:0]															
	Исх. значение																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

- бит 3 RSTCAL сбрасывает калибровку;
- бит 2 CAL запускает калибровку АЦП путем записи 1, которая по окончании процесса сбрасывается аппаратно;
- бит 1 CONT запускает непрерывное преобразование;
- бит 0 ADON отключает и включает модуль АЦП.

Регистры ADC_SMPR1 и ADC_SMPR2 задают время преобразования в каждом канале. Для каждого канала выделено по три разряда, что позволяет задать 8 значений времени преобразования в циклах (000 = 1.5 цикла, 001 = 7.5 цикла, 010 = 13.5 цикла, 011 = 28.5 цикла, 100 = 41.5 цикла, 101 = 55.5 цикла, 110 = 71.5 цикла, 111 = 239.5 цикла).

Регистры ADC_JOFR1...ADC_JOFR4 служат для задания смещения каждого канала инжектированной группы (JOFR1, JOFR2, JOFR3, JOFR4). В эти регистры можно записывать 12-битное значение, которое автоматически вычитается из результата преобразования АЦП. Если результирующее значение станет отрицательным, то регистр результата инжектированной группы дополнится битом знака отрицательного значения.

Регистры ADC_LTR и ADC_HTR имеют по 12 разрядов, в которые записываются значения верхней и нижней границы оконного компаратора.

Регистры ADC_SQR1... ADC_SQR3 служат для задания номеров каналов, которые будут опрашиваться в регулярной группе, и общее количество каналов. Разряды SQx[4:0] этих регистров задают номер канала, где x – это номер позиции для преобразования от 1 до 16.

Количество каналов в группе задается разрядами L[3:0] регистра ADC_SQR1.

Например, необходимо регулярно опрашивать 7 каналов в очередности 1, 3, 1, 5, 9, 9, 1. Для этого нужно записать в регистр ADC_SQR3 следующие значения: SQ1[4:0]=1, SQ2[4:0]=3, SQ3[4:0]=1, SQ4[4:0]=5, SQ5[4:0]=9, SQ6[4:0]=9. В регистр ADC_SQR2 требуется записать номер последнего седьмого опрашиваемого канала: SQ7[4:0]=1. После этого следует задать количество опрашиваемых каналов регулярной группы для регистра ADC_SQR1: L[3:0]=7. Из приведенной информации видно, что каналы можно опрашивать в любой последовательности. Кроме того, любой канал можно опрашивать несколько раз.

Регистр ADC_JSQR, подобно регистрам ADC_SQRx, задает последовательность опрашиваемых каналов и их количество для инжектированной группы каналов. Как видно из структуры регистра, максимальное количество преобразований в инжектированной группе равно четырем. Количество каналов задается в разрядах JL[1:0].

Регистры ADC_JDR1...ADC_JDR4 хранят данные каналов инжектированной группы. Максимальное количество каналов в инжектированной группе равно четырем, соответственно имеется четыре регистра данных. В эти регистры помещается результат преобразований в каждом выбранном канале. Поскольку регистры 16-разрядные, а АЦП 12-разрядный, предусмотрена возможность выравнивания результата измерения по левому или по правому краю этих регистров.

Регистр ADC_DR содержит результат преобразования АЦП в регулярной группе. В отличие от инжектированной группы, в которой четыре регистра данных, он — один на всю группу. Его структура подобна регистрам JDRx, за исключением старших разрядов 16...31. Эти разряды используются при работе АЦП в сдвоенном режиме, совместно со вторым АЦП, именуемым ADC2.

РЕЖИМЫ РАБОТЫ АЦП

АЦП позволяет использовать несколько режимов работы. Рассмотрим их поочередно, начиная с режима одиночного преобразования. В этом режиме АЦП выполняет всего одно преобразование. Оно запускается после установки разряда ADON в регистре ADC_CR2 для регулярных каналов, или от внешнего сигнала для регулярного и инжектированного каналов. При этом разряд CONT регистра ADC_CR2 должен быть равен нулю. После окончания преобразования в выбранном регулярном канале результат преобразования сохраняется в регистре ADC_DR и устанавливается флаг EOC. Если установлен разряд EOCIE, генерируется прерывание. Для инжектированного канала результат преобразования сохраняется в регистре ADC_DRJ1 и устанавливается флаг JEOC. Если установлен разряд JEOCIE, генерируется прерывание. После этого работа АЦП останавливается.

В режиме непрерывного преобразования АЦП начинает очередное преобразование после завершения текущего. Этот режим запускается от внешнего источника или путем установки разряда ADON регистра ADC_CR2. При этом разряд CONT регистра ADC_CR2 должен быть равен единице. После каждого преобразования выполняется результат преобразования и сохраняется, аналогично режиму одиночного преобразования.

В режиме сканирования АЦП выполняет поочередное преобразование группы каналов. Этот режим выбирается установкой разряда SCAN регистра ADC_CR1. Если этот разряд установлен, АЦП сканирует все каналы, выбранные в регистрах ADC_SQRx для регулярных каналов или в регистре ADC_JSQR для инжектированных каналов. Если разряд CONT установлен, то преобразование не останавливается на последнем канале, а вновь запускается с первого канала. Если установлен разряд DMA,

Листинг 1

```
//=========================
// Функция инициализации АЦП
//=========================
void Init_ADC(void)
{
RCC->APB2ENR |= RCC_APB2ENR_IOPAEN; // Разрешить тактирование порта PORTA
// Сконфигурировать PORTA.6 как аналоговый вход
GPIOA->CRL &= ~GPIO_CRL_MODE6; // Очистить биты MODE
GPIOA->CRL &= ~GPIO_CRL_CNF6; // Очистить биты CNF
RCC->APB2ENR |= RCC_APB2ENR_ADC1EN; // Включить тактирование АЦП
RCC->CFGR &= ~RCC_CFGR_ADCPRE; // Задать значение делителя тактовой частоты
ADC1->CR1 = 0; // Обнулить регистр управления
ADC1->SQR1 = 0; // Обнулить регистр SQR1
ADC1->CR2 |= ADC_CR2_CAL; // Пуск калибровки
while (!(ADC1->CR2 & ADC_CR2_CAL)){}; // Ждать окончания калибровки
ADC1->CR2 = ADC_CR2_EXTSEL; // Выбрать источником запуска разряд SWSTART
ADC1->CR2 |= ADC_CR2_EXTTRIG; // Разрешить внешний запуск регулярного канала
ADC1->CR2 |= ADC_CR2_ADON; // Включить АЦП
}
//=====================================================
// Функция запуска преобразования АЦП и чтение выбранного канала
// Вход - номер канала для преобразования
// Выход - результат преобразования
//=====================================================
uint16_t RD_ADC(uint8_t nk)
{
ADC1->SQR3 = nk; // Задать номер канала
ADC1->CR2 |= ADC_CR2_SWSTART; // Пуск преобразования
while(!(ADC1->SR & ADC_SR_EOC)){}; // Ждать окончания преобразования
return ADC1->DR; // Считать результат преобразования
}
//=========================
// Главный модуль программы
//=========================
void main(void)
{
unsigned int ai; // Инициализация переменных
// Другие команды …
while(1) // Бесконечный цикл
{
Init_ADC(); // Выполнить инициализацию и пуск АЦП
ai = RD_ADC(6); // Считать данные АЦП для канала 6
// Другие команды …
}
}
```

контроллер прямого доступа к памяти используется для передачи результата в память по окончании каждого преобразования после установки разряда EOC.

Если в регулярной группе используется более одного канала, то для сохранения результата рекомендуется использовать DMA. Это позволяет избежать потери данных, которые уже хранятся в регистре ADC_DR. В конце преобразования регулярная группа формирует запрос DMA для сохранения результата из регистра ADC_DR в место, заданное пользователем.

Режимы работы АЦП с инжектированными каналами также разнообразны. Чтобы использовать запуск инжекционного канала, надо чтобы бит JAUTO был обнулен, а бит SCAN в регистре ADC_CR1 — установлен. Запуск преобразования сигналов для группы регулярных каналов производится либо внешним сигналом, либо установкой бита ADON в регистре ADC_CR2. Если внешний запуск инжектированных каналов происходит во время преобразования сигналов группы регулярных каналов, то текущее преобразование приостанавливается и запускается преобразование последовательности инжекционных каналов в режиме однократного сканирования.

Затем возобновляется преобразование группы регулярных каналов, начиная с того канала, который был прерван. Если во время инжекционного преобразования происходит событие запуска регулярных каналов, то оно не прерывает это преобразование, но запуск регулярной последовательности выполняется сразу после окончания обработки инжекционной последовательности. При использовании запуска инжектированных каналов внешним событием необходимо убедиться, что интервал между событиями запуска дольше, чем время преобразования инжекционных каналов. Например, если длительность последовательности преобразований составляет 28 циклов тактового АЦП, то минимальный интервал между событиями запуска должен быть 29 циклов.

Если установлен бит JAUTO, то после завершения преобразования группы регулярных каналов автоматически запускается преобразование группы инжекционных каналов. Это можно использовать, чтобы преобразовать последовательность из 17–20 каналов, заданных в регистрах ADC_JSQR и ADC_SQRx. В этом режиме механизм запуска внешним событием должен быть отключен. Если, в дополнение к биту JAUTO установлен бит CONT, то, сразу после преобразования группы инжекционных каналов вновь запускается преобразование группы регулярных каналов. Использовать одновременно оба режима, автоматический запуск и запуск внешним событием, невозможно.

КАЛИБРОВКА АЦП

АЦП имеет встроенный механизм автоматической калибровки. Калибровка значительно уменьшает погрешность оцифровки, которая вызывается неоднородностью внутренних конденсаторов выборки и хранения сигналов. Во время калибровки вычисляется цифровое значение АЦП для каждого конденсатора в виде корректирующего кода. Во время всех последующих преобразований этот код используется для компенсации ошибки преобразования сигналов.

Перед началом калибровки АЦП должен находиться в отключенном состоянии, то есть бит ADON должен быть равен нулю, по крайней мере в течение двух циклов тактового АЦП. Запуск калибровки производится установкой бита CAL в регистре ADC_CR2. Как только калибровка закончена, бит CAL сбрасывается аппаратно, после чего может быть выполнено нормальное преобразование. Рекомендуется калибровать АЦП один раз при подаче питания. Коды калибровки сохраняются в ADC_DR при завершении фазы калибровки.

ВРЕМЯ ПРЕОБРАЗОВАНИЯ АЦП

АЦП поддерживает возможность раздельного программирования времени преобразования в каждом из каналов. Предусмотрена возможность выбора 8 дискретных значений из диапазона 1.5...239.5 циклов. Для каждого канала время задается индивидуально, с помощью разрядов SMPx[2:0] регистров ADC_SMPR1 и ADC_SMPR2.

Полное время преобразования T_{conv} вычисляется по формуле: $T_{conv} = T_{smt} +$ + 12.5 циклов, где T_{smt} — это программно заданное время выборки. Например, частота ADCCLK задана равной 14 МГц, а время выборки 1.5 цикла. Тогда $T_{conv} = 1.5 + 12.5 = 14$ циклов = 1мкс.

ДАТЧИК ТЕМПЕРАТУРЫ АЦП

Встроенный в кристалл микроконтроллера датчик температуры позволяет измерять температуру самого кристалла. Датчик подключен к 16-му входу АЦП. Рекомендуемое время выборки при опросе датчика составляет 17.1 мкс. Когда датчик не используется, его можно отключить. Напряжение на выходе датчика изменяется линейно с температурой. В разных кристаллах эта линия смещается до 45 градусов, что связано с технологией изготовления. Внутренний датчик больше подходит не для измерения абсолютного значения температуры, а для контроля ее изменения.

Для чтения температуры необходимо выбрать канал 16, задать время выборки не менее 17.1 мкс и установить разряд TSVREFE в регистре ADC_CR2 для включения датчика. После этого можно запустить преобразование АЦП установкой разряда ADON или внешним событием и прочитать результат преобразования.

Вычисляется значение температуры по формуле:

$$T[°C] = (V25 - Vsense) / Avg\_Slope + 25,$$

где V25 — значение измерения при 25 градусах, имеющее типовое значение 1.41 В, Vsense — текущее измеренное значение, а Avg_Slope — коэффициент из таблицы на кристалл, имеющий типовое значение 4.3 мВ/°С.

ОКОННЫЙ КОМПАРАТОР

Оконный компаратор может использоваться для мониторинга выбранного регулярного или инжектированного канала, или всех регулярных или инжектированных каналов. Помимо мониторинга напряжения, функция оконного компаратора может использоваться в качестве детектора пересечения нуля.

Флаг AWD устанавливается, если цифровое значение канального сигнала, измеренного с помощью АЦП, больше или меньше заданного уровня. Этот уровень задается регистрами ADC_HTR и ADC_LTR. Прерывание может быть разрешено разрядом AWDIE регистра ADC_CR1. Каналы АЦП, контролируемые оконным компаратором, задаются согласно таблице 2.

Таблица 2. Каналы АЦП, контролируемые оконным компаратором

Канал	Разряды регистра управления ADC_CR1		
	AWDSGL	AWDEN	JAWDEN
Отсутствует	0 или 1	0	0
Все инжектированные каналы	0	0	1
Все регулярные каналы	0	1	0
Все инжектированные и регулярные каналы	0	1	1
Один инжектированный канал*	1	0	1
Один регулярный канал*	1	1	0
Один инжектированный или регулярный канал*	1	1	1
* Выбирается разрядами AWDCH[4:0]			

Разряд AWDSGL задает количество контролируемых каналов. Если он равен 0, то контролируются все каналы, а если 1 — то один канал. Номер канала при этом задается с помощью разрядов AWDCH[4:0].

Разряд AWDEN разрешает контроль каналов регулярной группы, а JAWDEN — контроль каналов инжектированной группы.

ПРИМЕРЫ ПРОГРАММ

Рассмотрим примеры программ для работы с АЦП. Начнем с простей-

Листинг 2

```
//=========================
// Функция инициализации АЦП
//=========================
void Init_ADC(void)
{
RCC->APB2ENR |= RCC_APB2ENR_IOPAEN; // Разрешить тактирование порта PORTA
// Сконфигурировать PORTA.6 как аналоговый вход
GPIOA->CRL &= ~GPIO_CRL_MODE6; // Очистить биты MODE
GPIOA->CRL &= ~GPIO_CRL_CNF6; // Очистить биты CNF
RCC->APB2ENR |= RCC_APB2ENR_ADC1EN; // Включить тактирование АЦП
RCC->CFGR &= ~RCC_CFGR_ADCPRE; // Задать значение делителя тактовой частоты
ADC1->CR1 = 0; // Обнулить регистр управления
ADC1->SQR1 = 0; // Обнулить регистр SQR1
ADC1->CR2 |= ADC_CR2_CAL; // Пуск калибровки
while (!(ADC1->CR2 & ADC_CR2_CAL)){}; // Ждать окончания калибровки
ADC1->CR2 = ADC_CR2_EXTSEL; // Выбрать источником запуска разряд SWSTART
ADC1->CR2 |= ADC_CR2_EXTTRIG; // Разрешить внешний запуск регулярного канала
ADC1->CR2 |= ADC_CR2_CONT; // Включить режим непрерывного преобразования
ADC1->CR2 |= ADC_CR2_ADON; // Включить АЦП
ADC1->CR2 |= ADC_CR2_SWSTART; // // Пуск преобразования
}
//=========================
// Главный модуль программы
//=========================
void main(void)
{
unsigned int ai; // Инициализация переменных
Init_ADC(); // Выполнить инициализацию и пуск АЦП
// Другие команды …
while(1) // Бесконечный цикл
{
ai=ADC1->DR; // Считать результат преобразования
// Другие команды …
}
}
```

Листинг 3

```
//=========================
// Функция инициализации АЦП
//=========================
void Init_ADC(void)
{
RCC->APB2ENR |= RCC_APB2ENR_IOPAEN; // Разрешить тактирование порта PORTA
// Сконфигурировать PORTA.3 как аналоговый вход
GPIOA->CRL &= ~GPIO_CRL_MODE3; // Очистить биты MODE
GPIOA->CRL &= ~GPIO_CRL_CNF3; // Очистить биты CNF
// Сконфигурировать PORTA.4 как аналоговый вход
GPIOA->CRL &= ~GPIO_CRL_MODE4; // Очистить биты MODE
GPIOA->CRL &= ~GPIO_CRL_CNF4; // Очистить биты CNF
// Сконфигурировать PORTA.5 как аналоговый вход
GPIOA->CRL &= ~GPIO_CRL_MODE5; // Очистить биты MODE
GPIOA->CRL &= ~GPIO_CRL_CNF5; // Очистить биты CNF
// Сконфигурировать PORTA.6 как аналоговый вход
GPIOA->CRL &= ~GPIO_CRL_MODE6; // Очистить биты MODE
GPIOA->CRL &= ~GPIO_CRL_CNF6; // Очистить биты CNF
RCC->APB2ENR |= RCC_APB2ENR_ADC1EN; // Включить тактирование АЦП
RCC->CFGR &= ~RCC_CFGR_ADCPRE; // Задать значение делителя тактовой частоты
ADC1->CR1 = 0; // Обнулить регистр управления
ADC1->CR2 |= ADC_CR2_CAL; // Пуск калибровки
while (!(ADC1->CR2 & ADC_CR2_CAL)){}; // Ждать окончания калибровки
ADC1->CR2 = ADC_CR2_JEXTSEL; // Выбрать источником запуска разряд JSWSTART
ADC1->CR2 |= ADC_CR2_JEXTTRIG; // Разрешить внешний запуск инжектированной
группы
ADC1->CR2 |= ADC_CR2_CONT; // Включить режим непрерывного преобразования
ADC1->CR1 |= ADC_CR1_SCAN; // Включить режим сканирования нескольких каналов
ADC1->CR1 |= ADC_CR1_JAUTO; // Автоматический запуск инжектированной группы
ADC1->JSQR = (uint32_t)(4-1)<<20; // Задать количество каналов в инжектированной
группе=4
ADC1->JSQR |= (uint32_t)3<<(5*0); // Номер первого канала для преобразования=3
ADC1->JSQR |= (uint32_t)4<<(5*1); // Номер второго канала для преобразования=4
ADC1->JSQR |= (uint32_t)5<<(5*2); // Номер третьего канала для преобразования=5
ADC1->JSQR |= (uint32_t)6<<(5*3); // Номер четвертого канала для преобразования=6
ADC1->CR2 |= ADC_CR2_ADON; // Включить АЦП
ADC1->CR2 |= ADC_CR2_JSWSTART; // Пуск преобразования
}
//=========================
// Главный модуль программы
//=========================
void main(void)
{
unsigned int ai[4]; // Инициализация переменных
Init_ADC(); // Выполнить инициализацию и пуск АЦП
// Другие команды …
while(1) // Бесконечный цикл
{
// Считать результат преобразования
ai[0]=ADC1->JDR1; // Канал 3
ai[1]=ADC1->JDR2; // Канал 4
ai[2]=ADC1->JDR3; // Канал 5
ai[3]=ADC1->JDR4; // Канал 6
// Другие команды …
}
}
```

шего варианта: измерение уровня сигнала на одном канале с программным запуском. Для этого используем вывод порта PORTA.6 микроконтроллера. Пример программы приведен в листинге 1.

Данный пример демонстрирует программу, в которой необходимо регулярно запускать АЦП и ждать окончания преобразования.

Теперь рассмотрим пример другой программы, приведенный в листинге 2, в которой задействован режим непрерывного преобразования АЦП.

Данная программа запускает режим непрерывного преобразования заданного канала, и АЦП постоянно опрашивает выбранный канал, помещая результат в регистр данных. Поэтому в основной программе не требуется регулярно запускать АЦП. При таком способе в регистре ADC_DR информация об уровне сигнала на входе канала будет постоянно обновляться. Обновление будет происходить с периодичностью, которую можно изменять.

В рассмотренных примерах можно запустить непрерывное преобразование для одного канала. Если нужно сделать это для нескольких каналов, то без использования прерываний или блока ПДП не обойтись, так как для регулярной группы предусмотрен всего один регистр данных. Однако если количество каналов не превышает четырех, то подобным образом можно использовать и преобразование инжектированных каналов.

Максимальное количество каналов для инжектированной группы равно четырем. Для каждого такого канала предусмотрен свой регистр для хранения результата преобразования.

Допустим, необходимо опрашивать четыре канала с номерами: 3, 4, 5 и 6. Для этого можно запустить непрерывное измерение группы из четырех инжектированных каналов. При этом АЦП будет автоматически сканировать четыре заданных канала, и помещать результат в регистры данных ADC_JDR1, ADC_JDR2, ADC_JDR3 и ADC_JDR4. При этом можно будет считывать данные из нужного канала в любое время. Пример такой программы приведен в листинге 3.

После выполнения приведенной функции инициализации АЦП в регистрах данных инжектированной группы результаты преобразования, заданные в программе каналов, будут регулярно обновляться. Поэтому данные результаты можно будет считывать в произвольное время по мере необходимости.

Подробнее ознакомиться с блоком АЦП микроконтроллера STM32 можно на сайте [2].

Литература:

1. www.st.com.
2. www.st.com/web/en/resource/technical/document/reference_manual/CD00246267.pdf.

CNY

* Статья перепечатана из журнала «Современная электроника», № 3, 2015 г., с разрешения редакции, тел. +7 (495) 232-00-87, www.soel.ru