|  |  |
| --- | --- |
| Introduction  This reference manual targets application developers. It provides complete information on  how to use the STM32H742xx, STM32H743/53xx and STM32H750xB microcontroller  memory and peripherals.  The STM32H742, STM32H743/753 and STM32H750 are lines of microcontrollers with  different memory sizes, packages and peripherals.  For ordering information, mechanical, and electrical device characteristics refer to the  corresponding datasheets.  For information on the Arm® Cortex®-M7 with FPU core, refer to the corresponding Arm  Technical Reference Manuals. | Введение  Данное справочное руководство предназначено для разработчиков приложений. В нем содержится полная информация о том. как использовать микроконтроллеры STM32H742xx, STM32H743/53xx и STM32H750xB памяти и периферийных устройств.  STM32H742, STM32H743/753 и STM32H750 - это линейки микроконтроллеров с различными объемами памяти, корпусами и периферийными устройствами.  Информация о заказе, механические и электрические характеристики устройств приведены в соответствующие технические паспорта.  Информацию о ядре Arm® Cortex®-M7 с FPU см. в соответствующих руководствах Arm Технические справочные руководства. |
|  |  |
| Analog-to-digital converters (ADC) | Analog-to-digital converters (ADC) |
| Introduction  This section describes the ADC implementation:  • ADC1 and ADC2 are tightly coupled and can operate in dual mode (ADC1 is master).  • ADC3 is instantiated separately.  Each ADC consists of a 16-bit successive approximation analog-to-digital converter.  Each ADC has up to 20 multiplexed channels. A/D conversion of the various channels can be performed in single, continuous, scan or discontinuous mode. The result of the ADC is stored in a left-aligned or right-aligned 32-bit data register.  The ADCs are mapped on the AHB bus to allow fast data handling.  The analog watchdog features allow the application to detect if the input voltage goes  outside the user-defined high or low thresholds.  A built-in hardware oversampler allows to improve analog performances while off-loading the related computational burden from the CPU.  An efficient low-power mode is implemented to allow very low consumption at low  frequency. | Введение  В этом разделе описывается реализация АЦП:  - АЦП1 и АЦП2 тесно связаны между собой и могут работать в двойном режиме (АЦП1 является ведущим).  - АЦП3 выполнен отдельно.  Каждый АЦП состоит из 16-разрядного аналого-цифрового преобразователя последовательного приближения.  Каждый АЦП имеет до 20 мультиплексированных каналов. АЦП различных каналов может выполняться в режиме  одиночном,  непрерывном,  сканирование или  прерывистом режиме.  Результат АЦП хранится в 32-битном регистре данных с левым или правым выравниванием.  АЦП отображены на шине AHB, что обеспечивает быструю обработку данных.  Функции аналогового сторожевого таймера позволяют приложению обнаружить, если входное напряжение выходит  за пределы заданных пользователем пороговых значений высокого или низкого уровня.  Встроенный аппаратный оверсемплер <#oversampling> позволяет улучшить аналоговые характеристики, одновременно разгружая с центрального процессора соответствующую вычислительную нагрузку.  Эффективный режим низкого энергопотребления реализован для обеспечения очень низкого потребления на низкой частоте. |
| ADC main features  • High-performance features  – Up to 2x ADCs which can operate in dual mode  – 16, 14, 12, 10 or 8-bit configurable resolution  – ADC conversion time is independent from the AHB bus clock frequency  – Faster conversion time by lowering resolution  – Can manage Single-ended or differential inputs (programmable per channels)  – AHB slave bus interface to allow fast data handling  – Self-calibration (both offset and linearity)  – Channel-wise programmable sampling time  – Up to four injected channels (analog inputs assignment to regular or injected  channels is fully configurable)  – Hardware assistant to prepare the context of the injected channels to allow fast  context switching  – Data alignment with in-built data coherency  – Data can be managed by GP-DMA for regular channel conversions with FIFO  – Data can be routed to DFSDM for post processing  – 4 dedicated data registers for the injected channels  • Oversampler  – 32-bit data register  – Oversampling ratio adjustable from 2 to 1024x  – Programmable data right and left shift  • Low-power features  – Speed adaptive low-power mode to reduce ADC consumption when operating at low frequency  – Allows slow bus frequency application while keeping optimum ADC performance  – Provides automatic control to avoid ADC overrun in low AHB bus clock frequency  application (auto-delayed mode)  • Each ADC features an external analog input channel  – Up to 6 fast channels from dedicated GPIO pads  – Up to 14 slow channels from dedicated GPIO pads  • In addition, there are 5 internal dedicated channels  – Internal reference voltage (VREFINT)  – Internal temperature sensor (VSENSE)  – BVAT monitoring channel (VBAT/4)  – Connection to DAC internal channels  • Start-of-conversion can be initiated:  – by software for both regular and injected conversions  – by hardware triggers with configurable polarity (internal timers events or GPIO  input events) for both regular and injected conversions  • Conversion modes  – Each ADC can convert a single channel or can scan a sequence of channels  – Single mode converts selected inputs once per trigger  – Continuous mode converts selected inputs continuously  – Discontinuous mode  • Dual ADC mode for ADC1 and 2  • Interrupt generation at ADC ready, the end of sampling, the end of conversion (regular or injected), end of sequence conversion (regular or injected), analog watchdog 1, 2 or 3 or overrun events  • 3 analog watchdogs per ADC  • ADC input range: VREF– ≤ VIN ≤ VREF+  Figure 135 shows the block diagram of one ADC. | Основные характеристики АЦП  - Высокопроизводительные характеристики  - До 2х АЦП, которые могут работать в двойном режиме  - Конфигурируемое разрешение 16, 14, 12, 10 или 8 бит  - Время преобразования АЦП не зависит от тактовой частоты шины AHB  - Ускоренное время преобразования при снижении разрешения  - Может управлять односторонними или дифференциальными входами (программируется для каждого канала)  - Интерфейс ведомой шины AHB для быстрой обработки данных  - Самокалибровка (как смещение, так и линейность)  - Программируемое по каналам время выборки  - До четырех инжектируемых каналов (назначение аналоговых входов на обычные или инжектируемые каналы полностью конфигурируемо)  - Аппаратный помощник для подготовки состояния инжектированных каналов для обеспечения быстрого  переключение контекста  - Выравнивание данных со встроенной когерентностью данных  - Данные могут управляться GP-DMA для преобразования обычных каналов с помощью FIFO  - Данные могут быть направлены в DFSDM для последующей обработки.  - 4 выделенных регистра данных для инжектируемых каналов  - Оверсемплер  - 32-битный регистр данных  - Коэффициент передискретизации регулируется от 2 до 1024x  - Программируемый сдвиг данных вправо и влево  - Функции низкого энергопотребления  - Адаптивный режим пониженного энергопотребления для снижения потребления АЦП при работе на низкой частоте  - Позволяет использовать шины с низкой частотой при сохранении оптимальной производительности АЦП  - Обеспечивает автоматическое управление для предотвращения разгона АЦП при низкой тактовой частоте шины AHB  (режим автоматической задержки)  - Каждый АЦП имеет внешний аналоговый входной канал  - До 6 быстрых каналов с выделенных колодок GPIO  - До 14 медленных каналов с выделенных колодок GPIO  - Кроме того, имеется 5 внутренних выделенных каналов  - Внутреннее опорное напряжение (VREFINT)  - Внутренний датчик температуры (VSENSE)  - Канал мониторинга BVAT (VBAT/4)  - Подключение к внутренним каналам ЦАП  - Запуск преобразования может быть инициирован  - программно для обычного и инжектированного преобразования  - аппаратными триггерами с настраиваемой полярностью (события внутренних таймеров или GPIO  входные события) для обычного и инжектируемого преобразования.  - Режимы преобразования  - Каждый АЦП может преобразовывать один канал или сканировать последовательность каналов  - Одиночный режим преобразует выбранные входы один раз за триггер  - Непрерывный режим преобразует выбранные входы непрерывно  - Непрерывный режим  - Режим двойного АЦП для АЦП1 и 2  - Генерация прерывания при готовности АЦП, окончании выборки, окончании преобразования (обычного или инжектированного), окончании последовательного преобразования (обычного или инжектированного). или инжекции), конец последовательного преобразования (обычного или инжектированного), аналогового сторожевого таймера 1, 2 или 3 или события превышения  - 3 аналоговых сторожевых таймера на АЦП  - Диапазон входных сигналов АЦП: VREF- ≤ VIN ≤ VREF+  На рисунке 135 показана блок-схема одного АЦП. |
| ADC clocks  Dual clock domain architecture (revision V devices)  The dual clock-domain architecture means that the ADC clock is independent from the AHB bus clock.  The input clock is the same for all ADCs and can be selected between two different clock sources (see Figure 136: ADC Clock scheme):  1. The ADC clock can be a specific clock source, named adc\_ker\_ck\_input which is independent and asynchronous with the AHB clock.  It can be configured in the RCC (refer to RCC Section for more information on how to  generate the ADC clock (adc\_ker\_ck\_input) dedicated clock).  To select this scheme, CKMODE[1:0] bits of the ADCx\_CCR register must be reset.  2. The ADC clock can be derived from the system clock or system clock divided by two (adc\_sclk). In this mode, a programmable divider factor can be selected (/1, 2 or 4  according to bits CKMODE[1:0]).  To select this scheme, CKMODE[1:0] bits of the ADCx\_CCR register must be different  from “00”. adc\_sclk is equal to sys\_ck when HPRE is set to 0, otherwise it corresponds  to sys\_ck/2.  In both case, the clock divider factor of 2 is applied to the clock provided to the ADC analog block (fadc\_ker\_ck).  Option 1) has the advantage of reaching the maximum ADC clock frequency whatever the AHB clock scheme selected. The ADC clock can eventually be divided by the following ratio:  1, 2, 4, 6, 8, 10, 12, 16, 32, 64, 128, 256; using the prescaler configured with bits  PRESC[3:0] in the ADCx\_CCR register.  Option 2) has the advantage of using the system without additional PLL. In addition, when adc\_sclk is twice faster than the adc\_hclk clock, the latency between the trigger and the start of conversion is fixed. This can be useful when the ADC is triggered by a timer and if the application requires that the ADC is precisely triggered without any uncertainty (otherwise, an uncertainty of the trigger instant is added by the resynchronizations between the two clock domains).  The clock configured through CKMODE[1:0] bits must be compliant with the analog ADC operating frequency specified in the product datasheet.  Note: adc\_sclk is the system clock or system clock divided by two: when the ABH prescaler is set to 1 (HPRE[3:0] = 0XXX in RCC\_CFGR register), adc\_sclk is equal to sys\_clk, otherwise adc\_sclk corresponds to sys\_clk/2. | Тактовые частоты АЦП  Архитектура с двумя тактовыми доменами (устройства ревизии V)  Архитектура с двумя тактовыми доменами означает, что тактовый генератор АЦП не зависит от тактового генератора шины AHB.  Входной тактовый генератор одинаков для всех АЦП и может быть выбран между двумя различными источниками тактового сигнала (см. Рисунок 136: Схема тактового генератора АЦП):  1. Тактовый генератор АЦП может быть определенным источником тактового сигнала, названным adc\_ker\_ck\_input, который является независимым и асинхронным с тактовым генератором AHB.  Он может быть настроен в RCC (см. раздел RCC для получения дополнительной информации о том, как генерировать выделенный тактовый генератор АЦП (adc\_ker\_ck\_input)).  Для выбора этой схемы биты CKMODE[1:0] регистра ADCx\_CCR должны быть сброшены.  2. Тактовый генератор АЦП может быть получен от системного генератора или от системного генератора, деленного на два (adc\_sclk). В этом режиме может быть выбран программируемый коэффициент делителя (/1, 2 или 4  в соответствии с битами CKMODE[1:0]).  Для выбора этой схемы биты CKMODE[1:0] регистра ADCx\_CCR должны быть отличны от "00". adc\_sclk равен sys\_ck, когда HPRE установлен в 0, в противном случае он соответствует sys\_ck/2.  В обоих случаях к тактовому генератору, подаваемому на аналоговый блок АЦП (fadc\_ker\_ck), применяется коэффициент делителя тактового генератора, равный 2.  Вариант 1) имеет преимущество в достижении максимальной тактовой частоты АЦП независимо от выбранной тактовой схемы AHB. В конечном итоге тактовая частота АЦП может быть разделена в следующем соотношении:  1, 2, 4, 6, 8, 10, 12, 16, 32, 64, 128, 256; используя прескалер, сконфигурированный битами PRESC[3:0] в регистре ADCx\_CCR.  Вариант 2) имеет преимущество использования системы без дополнительного PLL. Кроме того, когда adc\_sclk в два раза быстрее тактового генератора adc\_hclk, задержка между триггером и начало преобразования фиксировано. Это может быть полезно, когда АЦП запускается таймером и если приложение требует точного запуска АЦП без какой-либо неопределенности (в противном случае неопределенность момента запуска добавляется из-за ресинхронизации между двумя тактовыми доменами).  Тактовый генератор, конфигурируемый через биты CKMODE[1:0], должен соответствовать рабочей частоте аналогового АЦП. указанной в техническом описании изделия.  Примечание: adc\_sclk - это системный тактовый генератор или системный тактовый генератор, деленный на два: когда прескалер ABH установлен на 1 (HPRE[3:0] = 0XXX в регистре RCC\_CFGR), adc\_sclk равен sys\_clk, иначе adc\_sclk соответствует sys\_clk/2. |
| Dual clock domain architecture (revision Y devices)  The dual clock-domain architecture means that the ADC clock is independent from the AHB bus clock.  The input clock is the same for the all ADCs and can be selected between two different  clock sources (see Figure 137: ADC clock scheme):  1. The ADC clock can be a specific clock source, named adc\_ker\_ck\_input which is  independent and asynchronous with the AHB clock.  It can be configured in the RCC (refer to RCC Section for more information on how to  generate the ADC clock (adc\_ker\_ck\_input) dedicated clock).  To select this scheme, CKMODE[1:0] bits of the ADCx\_CCR register must be reset.  2. The ADC clock can be derived from the AHB clock of the ADC bus interface, divided by a programmable factor (1, 2 or 4). In this mode, a programmable divider factor can be selected (/1, 2 or 4 according to bits CKMODE[1:0]).  To select this scheme, CKMODE[1:0] bits of the ADCx\_CCR register must be different  from “00”.  Note: For option 2), a prescaling factor of 1 (CKMODE[1:0]=01) can be used only if the AHB prescaler is set to 1 (HPRE[3:0] = 0XXX in RCC\_CFGR register, corresponding to a clock duty cycle of 50 %).  Option 1) has the advantage of reaching the maximum ADC clock frequency whatever the AHB clock scheme selected. The ADC clock can eventually be divided by the following ratio:  1, 2, 4, 6, 8, 10, 12, 16, 32, 64, 128, 256; using the prescaler configured with bits  PRESC[3:0] in the ADCx\_CCR register.  Option 2) has the advantage of bypassing the clock domain resynchronizations. This can be useful when the ADC is triggered by a timer and if the application requires that the ADC is precisely triggered without any uncertainty (otherwise, an uncertainty of the trigger instant is added by the resynchronizations between the two clock domains).  The clock configured through CKMODE[1:0] bits must be compliant with the operating  frequency specified in the product datasheet. | Архитектура с двумя тактовыми доменами (устройства ревизии Y)  Архитектура с двумя тактовыми доменами означает, что тактовый генератор АЦП не зависит от тактового генератора шины AHB.  Входной тактовый генератор одинаков для всех АЦП и может быть выбран между двумя различными источников тактовых импульсов (см. Рисунок 137: Схема тактовых импульсов АЦП):  1. Тактовым генератором АЦП может быть специальный источник тактового сигнала, названный adc\_ker\_ck\_input, который является  независимым и асинхронным с тактовым генератором AHB.  Он может быть настроен в RCC (см. раздел RCC для получения дополнительной информации о том, как  генерировать выделенный тактовый генератор АЦП (adc\_ker\_ck\_input)).  Для выбора этой схемы биты CKMODE[1:0] регистра ADCx\_CCR должны быть сброшены.  2. Тактовый генератор АЦП может быть получен из тактового генератора AHB интерфейса шины АЦП, разделенного на программируемый коэффициент (1, 2 или 4). В этом режиме программируемый коэффициент делителя может быть (/1, 2 или 4 в соответствии с битами CKMODE[1:0]).  Для выбора этой схемы биты CKMODE[1:0] регистра ADCx\_CCR должны быть отличны от "00".  Примечание: Для варианта 2) коэффициент предварительного масштабирования 1 (CKMODE[1:0]=01) может использоваться только в том случае, если AHB прескалер установлен на 1 (HPRE[3:0] = 0XXX в регистре RCC\_CFGR, что соответствует тактовому циклу 50 %).  Преимущество варианта 1) заключается в достижении максимальной тактовой частоты АЦП независимо от выбранной схемы тактирования AHB. В конечном итоге тактовая частота АЦП может быть разделена в следующем соотношении:  1, 2, 4, 6, 8, 10, 12, 16, 32, 64, 128, 256; используя прескалер, сконфигурированный битами PRESC[3:0] в регистре ADCx\_CCR.  Вариант 2) имеет то преимущество, что позволяет обойти ресинхронизацию тактового домена. Это может быть полезным, когда АЦП запускается таймером, и если приложение требует, чтобы АЦП был точного срабатывания без какой-либо неопределенности (в противном случае неопределенность момента срабатывания добавляется в результате ресинхронизации между двумя тактовыми доменами).  Тактовый генератор, конфигурируемый через биты CKMODE[1:0], должен соответствовать рабочей рабочей частоте, указанной в техническом описании изделия. |
| Clock ratio constraint between ADC clock and AHB clock  There are generally no constraints to be respected for the ratio between the ADC clock and the AHB clock except if some injected channels are programmed. In this case, it is mandatory to respect the following ratio:  • Fadc\_hclk ≥ Fadc\_ker\_ck / 4 if the resolution of all channels are 16-bit, 14-bit,12-bit or 10-bit  • Fadc\_hclk ≥ Fadc\_ker\_ck / 3 if there are some channels with resolutions equal to 8-bit (and none with lower resolutions) | Ограничение соотношения тактовых импульсов между тактовыми импульсами АЦП и AHB  Обычно не существует ограничений на соотношение между тактовым генератором АЦП и тактовым генератором AHB. AHB, за исключением случаев, когда программируются некоторые инжектируемые каналы. В этом случае необходимо обязательно соблюдать следующее соотношение:  - Fadc\_hclk ≥ Fadc\_ker\_ck / 4, если разрешение всех каналов 16-бит, 14-бит, 12-бит или 10- бит  - Fadc\_hclk ≥ Fadc\_ker\_ck / 3, если есть несколько каналов с разрешением 8-бит (и ни одного с меньшим разрешением). ни одного с меньшим разрешением) |
| BOOST control  The ADC Boost mode can be controlled through the BOOST bit in the ADC\_CR register.  This bit must be set according to the ADC clock setting. Refer to the ADC\_CR register  description. | Управление BOOST  Режим усиления АЦП может управляться через бит BOOST в регистре ADC\_CR.  Этот бит должен быть установлен в соответствии с настройками тактового генератора АЦП. См. описание регистра ADC\_CR  описание. |
| ADC1/2/3 connectivity  ADC1 and ADC2 are tightly coupled and share some external channels as described in the following figures.  ADC3 is instantiated separately, but some inputs are shared with ADC1 and ADC2. | Подключение АЦП1/2/3  АЦП1 и АЦП2 тесно связаны между собой и имеют общие внешние каналы, как описано на следующих рисунках.  АЦП3 выполнен отдельно, но некоторые входы являются общими с АЦП1 и АЦП2. |
| Slave AHB interface  The ADCs implement an AHB slave port for control/status register and data access. The features of the AHB interface are listed below:  • Word (32-bit) accesses  • Single cycle response  • Response to all read/write accesses to the registers with zero wait states.  The AHB slave interface does not support split/retry requests, and never generates AHB errors. | Ведомый интерфейс AHB  В АЦП реализован ведомый порт AHB для управления/регистра состояния и доступа к данным. На данном ресурсе особенности интерфейса AHB перечислены ниже:  - Словесный (32-битный) доступ  - Одноцикловый отклик  - Реакция на все обращения к регистрам чтения/записи с нулевым состоянием ожидания.  Ведомый интерфейс AHB не поддерживает запросы split/retry и никогда не генерирует ошибки AHB. |
| 25.4.6 ADC deep-power-down mode (DEEPPWD) and ADC voltage regulator  (ADVREGEN)  By default, the ADC is in deep-power-down mode where its supply is internally switched off to reduce the leakage currents (the reset state of bit DEEPPWD is 1 in the ADC\_CR register).  To start ADC operations, it is first needed to exit deep-power-down mode by clearing bit DEEPPWD=0.  Then, it is mandatory to enable the ADC internal voltage regulator by setting the bit  ADVREGEN=1 into ADC\_CR register. The software must wait for the startup time of the  ADC voltage regulator (TADCVREG\_STUP) before launching a calibration or enabling the ADC. This delay must be implemented by software.  The LDO status can be verified by checking the LDORDY bit in ADC\_ISR register (refer to Section 25.3: ADC implementation for the availability of the LDO regulator status).  For the startup time of the ADC voltage regulator, refer to device datasheet for  TADCVREG\_STUP parameter.  After ADC operations are complete, the ADC can be disabled (ADEN=0). It is possible to save power by also disabling the ADC voltage regulator. This is done by writing bit  ADVREGEN=0.  Then, to save more power by reducing the leakage currents, it is also possible to re-enter in ADC deep-power-down mode by setting bit DEEPPWD=1 into ADC\_CR register. This is particularly interesting before entering Stop mode.  Note: Writing DEEPPWD=1 automatically disables the ADC voltage regulator and bit ADVREGEN is automatically cleared.  Note: When the internal voltage regulator is disabled (ADVREGEN=0), the internal analog calibration is kept.  In ADC deep-power-down mode (DEEPPWD=1), the internal analog calibration is lost and it is necessary to either relaunch a calibration or apply again the calibration factor which was previously saved (refer to Section 25.4.8: Calibration (ADCAL, ADCALDIF, ADCALLIN, ADC\_CALFACT)). | 25.4.6 Режим глубокого отключения питания АЦП (DEEPPWD) и регулятор напряжения АЦП  (ADVREGEN)  По умолчанию АЦП находится в режиме глубокого энергосбережения, когда его питание внутренне отключено для уменьшения токов утечки (состояние сброса бита DEEPPWD в ADC\_CR равно 1). в регистре ADC\_CR).  Чтобы начать работу АЦП, сначала необходимо выйти из режима глубокого выключения питания, сбросив бит DEEPPWD=0.  Затем необходимо включить внутренний регулятор напряжения АЦП, установив бит ADVREGEN=1 в регистре ADC\_CR. Программное обеспечение должно выждать время запуска регулятора напряжения АЦП.  регулятора напряжения АЦП (TADCVREG\_STUP) перед запуском калибровки или включением АЦП. Эта задержка должна быть реализована программно.  Состояние LDO можно проверить, проверив бит LDORDY в регистре ADC\_ISR (см. Раздел 25.3: Реализация АЦП для получения информации о состоянии регулятора LDO).  Время запуска регулятора напряжения АЦП см. в техническом описании устройства для параметр TADCVREG\_STUP.  После завершения операций АЦП может быть отключен (ADEN=0). Можно экономии энергии, отключив также регулятор напряжения АЦП. Это делается путем записи бита ADVREGEN=0.  Затем, чтобы сэкономить еще больше энергии за счет уменьшения токов утечки, можно снова войти в режим глубокого выключения АЦП. АЦП в режим глубокого отключения питания, установив бит DEEPPWD=1 в регистре ADC\_CR. Это особенно интересно перед входом в режим Stop.  Примечание: Запись DEEPPWD=1 автоматически отключает регулятор напряжения АЦП, и бит ADVREGEN автоматически очищается.  Примечание: Когда внутренний регулятор напряжения отключен (ADVREGEN=0), внутренняя аналоговая калибровка сохраняется.  В режиме глубокого отключения питания АЦП (DEEPPWD=1) внутренняя аналоговая калибровка теряется и необходимо либо заново запустить калибровку, либо снова применить калибровочный коэффициент, который был (см. раздел 25.4.8: Калибровка (ADCAL, ADCALDIF, ADCALLIN, ADC\_CALFACT)). |
| 25.4.7 Single-ended and differential input channels  Channels can be configured to be either single-ended input or differential input by writing into bits DIFSEL[19:0] in the ADC\_DIFSEL register. This configuration must be written while the ADC is disabled (ADEN=0).  In single-ended input mode, the analog voltage to be converted for channel “i” is the  difference between the external voltage VINP[i] (positive input) and VREF- (negative input). In differential input mode, the analog voltage to be converted for channel “i” is the difference between the external voltage VINP[i] (positive input) and VINN[i] (negative input).  The output data for the differential mode is an unsigned data. When VINP[i] equals VREF-, VINN[i] equals VREF+ and the output data is 0x0000 (16-bit resolution mode). When VINP[i] equals VREF+, VINN[i] equals VREF- and the output data is 0xFFFF.    When ADC is configured as differential mode, both input should be biased at VREF+ / 2 voltage.  The input signal are supposed to be differential (common mode voltage should be fixed).  For a complete description of how the input channels are connected for each ADC, refer to Section 25.4.4: ADC1/2/3 connectivity.  Caution: When configuring the channel “i” in differential input mode, its negative input voltage is connected to VINN[i]. As a consequence, channel “i+n”, which is connected to VINN[i], should not be converted at same time by different ADCs. Some channels are shared between ADC1/ADC2: this can make the channel on the other ADC unusable. | 25.4.7 Односторонние и дифференциальные входные каналы  Каналы могут быть сконфигурированы как односторонний или дифференциальный вход путем записи в биты DIFSEL[19:0] в регистре ADC\_DIFSEL. Эта конфигурация должна быть записана, пока АЦП отключен (ADEN=0).  В режиме одностороннего входа аналоговое напряжение, подлежащее преобразованию для канала "i", равно разница между внешним напряжением VINP[i] (положительный вход) и VREF- (отрицательный вход). В режиме дифференциального входа аналоговое напряжение, подлежащее преобразованию для канала "i", представляет собой разность между внешним напряжением VINP[i] (положительный вход) и VINN[i] (отрицательный вход).  Выходные данные для дифференциального режима - это данные без знака. Когда VINP[i] равен VREF-, VINN[i] равен VREF+ и выходные данные равны 0x0000 (режим 16-битного разрешения). Когда VINP[i] равно VREF+, VINN[i] равно VREF- и выходные данные равны 0xFFFF.    Когда АЦП сконфигурирован как дифференциальный режим, оба входа должны быть смещены на VREF+ / 2 напряжение.  Входной сигнал должен быть дифференциальным (напряжение общей моды должно быть фиксированным).  Полное описание подключения входных каналов для каждого АЦП приведено в разделе Раздел 25.4.4: Подключение АЦП1/2/3.  Внимание: При конфигурировании канала "i" в режиме дифференциального входа, его отрицательное входное напряжение подключается к VINN[i]. Как следствие, канал "i+n", который подключен к VINN[i], не должен не должны одновременно преобразовываться разными АЦП. Некоторые каналы разделяются между ADC1/ADC2: это может сделать канал другого АЦП непригодным для использования. |
| 25.4.8 Calibration (ADCAL, ADCALDIF, ADCALLIN, ADC\_CALFACT)  Each ADC provides an automatic calibration procedure which drives all the calibration  sequence including the power-on/off sequence of the ADC. During the procedure, the ADC calculates a calibration factor which is 11-bits of offset or 160-bits of linearity and which is applied internally to the ADC until the next ADC power-off. During the calibration procedure, the application must not use the ADC and must wait until calibration is complete.  The calibration is preliminary to any ADC operation. It removes the systematic errors which may vary from chip to chip and allows to compensate offset and linearity deviation.  The calibration factor for the offset to be applied for single-ended input conversions is  different from the factor to be applied for differential input conversions:  • Write ADCALDIF=0 before launching a calibration which will be applied for single-  ended input conversions.  • Write ADCALDIF=1 before launching a calibration which will be applied for differential input conversions.  The linearity correction must be done once only, regardless of single / differential  configuration.  • Write ADCALLIN=1 before launching a calibration which will run the linearity calibration same time as the offset calibration.  • Write ADCALLIN=0 before launching a calibration which will not run the linearity  calibration but only the offset calibration.  The calibration is then initiated by software by setting bit ADCAL=1. It can be initiated only when the ADC is disabled (when ADEN=0). ADCAL bit stays at 1 during all the calibration sequence. It is then cleared by hardware as soon the calibration completes. At this time, the associated calibration factor is stored internally in the analog ADC and also in the bits CALFACT\_S[10:0] or CALFACT\_D[10:0] of ADC\_CALFACT register (depending on single-ended or differential input calibration). The 160-bit linearity calibration factor can be accessed using the ADC\_CALFACT2 register with ADEN set to 1.  The internal analog calibration is kept if the ADC is disabled (ADEN=0). However, if the ADC is disabled for extended periods, it is recommended that a new offset calibration cycle is run before enabling again the ADC.  The internal analog calibration is lost each time the power of the ADC is removed (example, when the product enters in STANDBY or VBAT mode). In this case, to avoid spending time recalibrating the ADC, it is possible to re-write the calibration factor into the ADC\_CALFACT and ADC\_CALFACT2 register without recalibrating, supposing that the software has previously saved the calibration factor delivered during the previous calibration.  The calibration factor can be written if the ADC is enabled but not converting (ADEN=1 and ADSTART=0 and JADSTART=0). Then, at the next start of conversion, the calibration factor will automatically be injected into the analog ADC. This loading is transparent and does not add any cycle latency to the start of the conversion. It is recommended to recalibrate when VREF+ voltage changed more than 10%.  Refer to the datasheets for the clock cycle requirement for both linear and offset calibration. | 25.4.8 Калибровка (ADCAL, ADCALDIF, ADCALLIN, ADC\_CALFACT)  Каждый АЦП обеспечивает процедуру автоматической калибровки, которая управляет всей последовательностью калибровки, включая последовательность включения/выключения питания.  последовательность, включая последовательность включения/выключения питания АЦП. Во время процедуры АЦП вычисляет коэффициент калибровки, который составляет 11 бит смещения или 160 бит линейности и который применяется внутри АЦП до следующего выключения питания АЦП. Во время процедуры калибровки, приложение не должно использовать АЦП и должно дождаться завершения калибровки.  Калибровка является предварительной перед любой операцией АЦП. Она устраняет систематические ошибки, которые которые могут варьироваться от микросхемы к микросхеме, и позволяет компенсировать смещение и отклонение линейности.  Калибровочный коэффициент для смещения, применяемый для преобразования с односторонним входом, отличается от коэффициента, применяемого для преобразования с односторонним входом.  отличается от коэффициента, применяемого для преобразования дифференциального входа:  - Запишите ADCALDIF=0 перед запуском калибровки, которая будет применяться для одностороннего преобразования.  преобразования для односторонних входов.  - Запишите ADCALDIF=1 перед запуском калибровки, которая будет применяться для дифференциального преобразования. входных преобразований.  Коррекция линейности должна выполняться только один раз, независимо от конфигурации одиночного/дифференциального прибора.  конфигурации.  - Запишите ADCALLIN=1 перед запуском калибровки, которая запустит калибровку линейности одновременно с калибровкой смещения.  - Запишите ADCALLIN=0 перед запуском калибровки, которая не будет запускать калибровку линейности, а только калибровку смещения.  Затем калибровка инициируется программно путем установки бита ADCAL=1. Она может быть инициирована только когда АЦП отключен (когда ADEN=0). Бит ADCAL остается равным 1 в течение всей калибровки. последовательности. После завершения калибровки он очищается аппаратными средствами. В это время соответствующий коэффициент калибровки сохраняется внутри аналогового АЦП, а также в битах CALFACT\_S[10:0] или CALFACT\_D[10:0] регистра ADC\_CALFACT (в зависимости от калибровки одностороннего или дифференциального входа). или дифференциального входа). 160-битный коэффициент калибровки линейности может быть доступен получить доступ с помощью регистра ADC\_CALFACT2 с ADEN, установленным в 1.  Внутренняя аналоговая калибровка сохраняется, если АЦП отключен (ADEN=0). Однако, если АЦП отключен на длительное время, рекомендуется запустить новый цикл калибровки смещения перед повторным включением АЦП.  Внутренняя аналоговая калибровка теряется при каждом отключении питания АЦП (например, когда устройство переходит в режим STANDBY или VBAT). В этом случае, чтобы не тратить время на повторную калибровку АЦП, можно повторно записать калибровочный коэффициент в ADC\_CALFACT и ADC\_CALFACT2 без повторной калибровки, предполагая, что программное обеспечение ранее сохранила калибровочный коэффициент, полученный во время предыдущей калибровки.  Калибровочный коэффициент может быть записан, если АЦП включен, но не выполняет преобразование (ADEN=1 и ADSTART=0 и JADSTART=0). Тогда, при следующем начале преобразования, калибровочный коэффициент будет автоматически введен в аналоговый АЦП. Эта загрузка является прозрачной и не добавляет никакой задержки цикла к началу преобразования. Рекомендуется проводить повторную калибровку, если напряжение VREF+ изменяется более чем на 10%.  Требования к тактовому циклу для линейной калибровки и калибровки смещения см. в технических описаниях. |
| Software procedure to calibrate the ADC  1. Ensure DEEPPWD=0, ADVREGEN=1 and verify that the ADC voltage regulator  startup time has elapsed by checking the LDORDY bit in ADC\_ISR (refer to  Section 25.3: ADC implementation for the availability of the LDO regulator status).  2. Ensure that ADEN=0.  3. Select the input mode for this calibration by setting ADCALDIF=0 (Single-ended input) or ADCALDIF=1 (Differential input). Select if Linearity calibration enable or not by ADCALLIN=1(enabled) or ADCALLIN=0(disabled).  4. Set ADCAL=1.  5. Wait until ADCAL=0.  6. The offset calibration factor can be read from ADC\_CALFACT register.  7. The linearity calibration factor can be read from ADC\_CALFACT2 register, following the procedure described in Section : Linearity calibration reading procedure (ADEN must be set to 1 prior to accessing ADC\_CALFACT2 register). | Программная процедура калибровки АЦП  1. Убедитесь, что DEEPPWD=0, ADVREGEN=1 и проверьте, что регулятор напряжения АЦП время запуска регулятора напряжения АЦП истекло путем проверки бита LDORDY в ADC\_ISR (см. Раздел 25.3: Реализация АЦП для получения информации о состоянии регулятора LDO).  2. Убедитесь, что ADEN=0.  3. Выберите режим ввода для этой калибровки, установив ADCALDIF=0 (односторонний вход) или ADCALDIF=1 (дифференциальный вход). Выберите, включена или нет калибровка линейности путем ADCALLIN=1 (включена) или ADCALLIN=0 (отключена).  4. Установите ADCAL=1.  5. Подождите, пока ADCAL=0.  6. Коэффициент калибровки смещения можно считать из регистра ADC\_CALFACT.  7. Коэффициент калибровки линейности может быть считан из регистра ADC\_CALFACT2, следуя процедуре, описанной в разделе : Процедура считывания калибровки линейности (ADEN должен быть должен быть установлен в 1 перед обращением к регистру ADC\_CALFACT2). |
| Software procedure to re-inject a calibration factor into the ADC  1. Ensure ADEN=1 and ADSTART=0 and JADSTART=0 (ADC enabled and no conversion is ongoing).  2. Write CALFACT\_S and CALFACT\_D with the new offset calibration factors.  3. Write LINCALFACT bits with the new linearity calibration factors, following the  procedure described in Section : Linearity calibration writing procedure.  4. When a conversion is launched, the calibration factor will be injected into the analog ADC only if the internal analog calibration factor differs from the one stored in bits CALFACT\_S for single-ended input channel or bits CALFACT\_D for differential input channel. | Программная процедура повторного ввода калибровочного коэффициента в АЦП  1. Убедитесь, что ADEN=1 и ADSTART=0 и JADSTART=0 (АЦП включен и преобразование не выполняется).  2. Запишите в CALFACT\_S и CALFACT\_D новые коэффициенты калибровки смещения.  3. Запишите биты LINCALFACT с новыми коэффициентами калибровки линейности, следуя процедуре, описанной в разделе : Процедура записи калибровки линейности.  4. При запуске преобразования калибровочный коэффициент будет введен в аналоговый АЦП только в том случае, если внутренний аналоговый калибровочный коэффициент отличается от коэффициента, сохраненного в битах CALFACT\_S для канала одностороннего входа или битах CALFACT\_D для канала дифференциального входа. канала. |
| Calibrating single-ended and differential analog inputs with a single ADC  If the ADC is supposed to convert both differential and single-ended inputs, two calibrations must be performed, one with ADCALDIF=0 and one with ADCALDIF=1. The procedure is the following:  1. Disable the ADC.  2. Calibrate the ADC in single-ended input mode (with ADCALDIF=0) and Linearity  calibration enable (with ADCALLIN=1). This updates the registers CALFACT\_S[10:0]  and LINCALFACT[159:0].  3. Calibrate the ADC in Differential input modes (with ADCALDIF=1) and Linearity  calibration disable (with ADCALLIN=0). This updates the register CALFACT\_D[10:0].  4. Enable the ADC, configure the channels and launch the conversions. Each time there is a switch from a single-ended to a differential inputs channel (and vice-versa), the calibration will automatically be injected into the analog ADC. | Калибровка односторонних и дифференциальных аналоговых входов с помощью одного АЦП  Если АЦП должен преобразовывать как дифференциальные, так и односторонние входы, необходимо выполнить две калибровки должны быть выполнены, одна с ADCALDIF=0 и одна с ADCALDIF=1. Процедура заключается в следующем следующая:  1. Отключите АЦП.  2. Откалибруйте АЦП в режиме одностороннего входа (с ADCALDIF=0) и линейности калибровки (с ADCALLIN=1). При этом обновляются регистры CALFACT\_S[10:0] и LINCALFACT[159:0].  3. Откалибруйте АЦП в режимах дифференциального входа (с ADCALDIF=1) и линейной калибровки (с ADCALLIN=1).  калибровка отключена (при ADCALLIN=0). При этом обновляется регистр CALFACT\_D[10:0].  4. Включите АЦП, настройте каналы и запустите преобразования. Каждый раз, когда происходит происходит переключение с канала одностороннего входа на канал дифференциального входа (и наоборот), калибровка будет автоматически вводиться в аналоговый АЦП. |
| Linearity calibration writing procedure  The six LINCALRDYW1..6 control/status bits in ADC\_CR are reset when the calibration has not yet been done or a new linearity calibration factor have been rewritten. It is possible to force directly a linearity calibration factor or re-inject it using the following procedure:  1. Ensure DEEPPWD=0, ADVREGEN=1 and that ADC voltage regulator startup time has  elapsed by checking the LDORDY bit in ADC\_ISR (refer to Section 25.3: ADC  implementation for the availability of the LDO regulator status).  2. Set ADEN = 1 and wait until ADRDY=1.  3. Write ADC\_CALFACT2[9:0] with previously saved linearity correction factor  bits[159:150].  4. Set LINCALRDYW6 bit.  5. Poll LINCALRDYW6 bit until returned value is one, indicating linearity correction  bits[159:150] have been effectively written.  6. Write ADC\_CALFACT2[29:0] with previously saved linearity correction factor  bits[149:120].  7. Set LINCALRDYW5 bit.  8. Poll LINCALRDYW5 bit until returned value is one, indicating linearity correction  bits[149:120] have been effectively written.  9. Write ADC\_CALFACT2[29:0] with previously saved linearity correction factor  bits[119:90].  10. Set LINCALRDYW4 bit.  11. Poll LINCALRDYW4 bit until returned value is one, indicating linearity correction  bits[119:90] have been effectively written.  12. Write ADC\_CALFACT2[29:0] with previously saved linearity correction factor  bits[89:60].  13. Set LINCALRDYW3 bit.  14. Poll LINCALRDYW3 bit until returned value is one, indicating linearity correction  bits[89:60] have been effectively written.  15. Write ADC\_CALFACT2[29:0] with previously saved linearity correction factor  bits[59:30].  16. Set LINCALRDYW2 bit.  17. Poll LINCALRDYW2 bit until returned value is one, indicating linearity correction  bits[59:30] have been effectively written.  18. Write ADC\_CALFACT2[29:0] with previously saved linearity correction factor bits[29:0].  19. Set LINCALRDYW1 bit.  20. Poll LINCALRDYW1 bit until returned value is one, indicating linearity correction  bits[29:0] have been effectively written.  Note: The software is allowed to toggle a single LINCALRDYWx bit at once (other bits left  unchanged), otherwise causing unexpected behavior.  The software is allowed to update the linearity calibration factor by writing  LINCALRDYW1..6 bits only when ADEN=1 and ADSTART=0 and JADSTART=0 (ADC  enabled and no conversion is ongoing). | Процедура записи калибровки линейности  Шесть битов управления/состояния LINCALRDYW1..6 в ADC\_CR сбрасываются, когда калибровка еще не была выполнена или был записан новый коэффициент калибровки линейности. Можно непосредственно принудительно ввести коэффициент калибровки линейности или перезаписать его с помощью следующей процедуры:  1. Убедитесь, что DEEPPWD=0, ADVREGEN=1 и что время запуска регулятора напряжения АЦП прошло  проверкой бита LDORDY в ADC\_ISR (см. раздел 25.3: АЦП  реализация для получения информации о состоянии регулятора LDO).  2. Установите ADEN = 1 и подождите, пока ADRDY=1.  3. Запишите ADC\_CALFACT2[9:0] с сохраненным ранее коэффициентом коррекции линейности  биты[159:150].  4. Установить бит LINCALRDYW6.  5. Опрашивать бит LINCALRDYW6 до тех пор, пока возвращаемое значение не станет единицей, что указывает на коррекцию линейности  биты[159:150] были эффективно записаны.  6. Записать ADC\_CALFACT2[29:0] с ранее сохраненным коэффициентом коррекции линейности  биты[149:120].  7. Установить бит LINCALRDYW5.  8. Опрашивать бит LINCALRDYW5 до тех пор, пока возвращаемое значение не станет единицей, что указывает на коррекцию линейности  биты[149:120] были эффективно записаны.  9. Записать ADC\_CALFACT2[29:0] с ранее сохраненным коэффициентом коррекции линейности  биты[119:90].  10. Установить бит LINCALRDYW4.  11. Опрашивать бит LINCALRDYW4 до тех пор, пока возвращаемое значение не станет единицей, что указывает на коррекцию линейности  биты[119:90] были эффективно записаны.  12. Записать ADC\_CALFACT2[29:0] с ранее сохраненным коэффициентом коррекции линейности  биты[89:60].  13. Установить бит LINCALRDYW3.  14. Опрашивать бит LINCALRDYW3 до тех пор, пока возвращаемое значение не станет единицей, что указывает на коррекцию линейности  биты[89:60] были эффективно записаны.  15. Записать ADC\_CALFACT2[29:0] с ранее сохраненным коэффициентом коррекции линейности  биты[59:30].  16. Установить бит LINCALRDYW2.  17. Опрашивать бит LINCALRDYW2 до тех пор, пока возвращаемое значение не станет равным единице, указывая на то, что коррекция линейности  биты[59:30] были эффективно записаны.  18. Записать ADC\_CALFACT2[29:0] с сохраненными ранее битами коррекции линейности[29:0].  19. Установить бит LINCALRDYW1.  20. Опрашивать бит LINCALRDYW1 до тех пор, пока возвращаемое значение не станет единицей, указывая на то, что коррекция линейности  биты[29:0] были эффективно записаны.  Примечание: Программе разрешается переключать только один бит LINCALRDYWx одновременно (остальные биты остаются  неизменными), иначе это приведет к неожиданному поведению.  Программе разрешено обновлять коэффициент калибровки линейности путем записи значения  LINCALRDYW1..6 только при ADEN=1 и ADSTART=0 и JADSTART=0 (АЦП  включен и преобразование не выполняется). |
| 25.4.9 ADC on-off control (ADEN, ADDIS, ADRDY)  First of all, follow the procedure explained in Section 25.4.6: ADC deep-power-down mode (DEEPPWD) and ADC voltage regulator (ADVREGEN)).  Once DEEPPWD = 0 and ADVREGEN = 1, the ADC can be enabled and the ADC needs a stabilization time of tSTAB before it starts converting accurately, as shown in Figure 144. Two control bits enable or disable the ADC:  • ADEN=1 enables the ADC. The flag ADRDY will be set once the ADC is ready for  operation.  • ADDIS=1 disables the ADC. ADEN and ADDIS are then automatically cleared by  hardware as soon as the analog ADC is effectively disabled.  Regular conversion can then start either by setting ADSTART=1 (refer to Section 25.4.19:  Conversion on external trigger and trigger polarity (EXTSEL, EXTEN, JEXTSEL, JEXTEN))  or when an external trigger event occurs, if triggers are enabled.  Injected conversions start by setting JADSTART=1 or when an external injected trigger  event occurs, if injected triggers are enabled.  Software procedure to enable the ADC  1. Clear the ADRDY bit in the ADC\_ISR register by writing ‘1’.  2. Set ADEN=1.  3. Wait until ADRDY=1 (ADRDY is set after the ADC startup time). This can be done  using the associated interrupt (setting ADRDYIE=1).  4. Clear the ADRDY bit in the ADC\_ISR register by writing ‘1’ (optional).  Software procedure to disable the ADC  1. Check that both ADSTART=0 and JADSTART=0 to ensure that no conversion is  ongoing. If required, stop any regular and injected conversion ongoing by setting  ADSTP=1 and JADSTP=1 and then wait until ADSTP=0 and JADSTP=0.  2. Set ADDIS=1.  3. If required by the application, wait until ADEN=0, until the analog ADC is effectively  disabled (ADDIS will automatically be reset once ADEN=0). | 25.4.9 Управление включением-выключением АЦП (ADEN, ADDIS, ADRDY)  Прежде всего, выполните процедуру, описанную в разделе 25.4.6: Режим глубокого отключения питания АЦП (DEEPPWD) и регулятор напряжения АЦП (ADVREGEN)).  Как только DEEPPWD = 0 и ADVREGEN = 1, АЦП может быть включен, и АЦП необходимо время стабилизации tSTAB до начала точного преобразования, как показано на рисунке 144. Два управляющих бита включают или выключают АЦП:  - ADEN=1 включает АЦП. Флаг ADRDY будет установлен, когда АЦП будет готов к работе.  - ADDIS=1 отключает АЦП. ADEN и ADDIS автоматически очищаются аппаратными средствами как только аналоговый АЦП будет отключен.  Затем можно начать регулярное преобразование, установив ADSTART=1 (см. раздел 25.4.19:  Преобразование при внешнем триггере и полярности триггера (EXTSEL, EXTEN, JEXTSEL, JEXTEN))  или при возникновении события внешнего триггера, если триггеры включены.  Инжектированные преобразования начинаются при установке JADSTART=1 или при возникновении события внешнего инжектированного триггера, если инжектируемые триггеры разрешены.  Программная процедура включения АЦП  1. Очистите бит ADRDY в регистре ADC\_ISR, записав '1'.  2. Установите ADEN=1.  3. Подождите, пока ADRDY=1 (ADRDY устанавливается после времени запуска АЦП). Это может быть сделано используя соответствующее прерывание (установка ADRDYIE=1).  4. Очистить бит ADRDY в регистре ADC\_ISR, записав '1' (необязательно).  Программная процедура отключения АЦП  1. Убедитесь, что ADSTART=0 и JADSTART=0, чтобы убедиться, что преобразование не продолжается. Если требуется, остановите все обычные и инжектируемые преобразования, установив значения  ADSTP=1 и JADSTP=1, а затем подождите, пока ADSTP=0 и JADSTP=0.  2. Установите ADDIS=1.  3. Если требуется по условиям применения, подождите, пока ADEN=0, пока аналоговый АЦП не будет фактически  (ADDIS будет автоматически сброшен, как только ADEN=0). |
| 25.4.10 Constraints when writing the ADC control bits  The software can write the RCC control bits to configure and enable the ADC clock (refer to RCC Section), the control bits DIFSEL in the ADC\_DIFSEL register, ADCx\_CCR register and the control bits ADCAL and ADEN in the ADC\_CR register, only if the ADC is disabled (ADEN must be equal to 0).  The software is then allowed to write the control bits ADSTART, JADSTART and ADDIS of the ADC\_CR register only if the ADC is enabled and there is no pending request to disable the ADC (ADEN must be equal to 1 and ADDIS to 0).  For all the other control bits of the ADC\_CFGR, ADC\_SMPRy, ADC\_TRy, ADC\_SQRy,  ADC\_JDRy, ADC\_OFRy and ADC\_IER registers:  • For control bits related to configuration of regular conversions, the software is allowed to write them only if the ADC is enabled (ADEN=1) and if there is no regular conversion ongoing (ADSTART must be equal to 0).  • For control bits related to configuration of injected conversions, the software is allowed to write them only if the ADC is enabled (ADEN=1) and if there is no injected  conversion ongoing (JADSTART must be equal to 0).  The software can write ADSTP or JADSTP control bits in the ADC\_CR register only if the ADC is enabled and eventually converting and if there is no pending request to disable the ADC (ADSTART or JADSTART must be equal to 1 and ADDIS to 0).  The software can write the register ADC\_JSQR at any time, when the ADC is enabled (ADEN=1).  The software is allowed to write the ADC\_JSQR register only when JADSTART is cleared to 0 (no injected conversion is ongoing) unless the context queue is enabled (JQDIS=0 in ADC\_CFGR register).  Note: There is no hardware protection to prevent these forbidden write accesses and ADC behavior may become in an unknown state. To recover from this situation, the ADC must be disabled (clear ADEN=0 as well as all the bits of ADC\_CR register). | 25.4.10 Ограничения при записи битов управления АЦП  Программное обеспечение может записывать управляющие биты RCC для конфигурирования и включения тактового генератора АЦП (см. Раздел RCC), управляющие биты DIFSEL в регистре ADC\_DIFSEL, регистре ADCx\_CCR и управляющие биты ADCAL и ADEN в регистре ADC\_CR, только если АЦП отключен (ADEN должен быть равен 0).  Затем программе разрешается записывать управляющие биты ADSTART, JADSTART и ADDIS регистра ADC\_CR, только если АЦП отключен (ADEN должен быть равен 0). регистра ADC\_CR, только если АЦП включен и нет ожидающего запроса на отключение АЦП (ADEN должен быть равен 0). АЦП (ADEN должен быть равен 1, а ADDIS - 0).  Для всех остальных битов управления ADC\_CFGR, ADC\_SMPRy, ADC\_TRy, ADC\_SQRy, ADC\_JDRy, ADC\_OFRy и ADC\_IER:  - Для управляющих битов, связанных с конфигурацией регулярных преобразований, программе разрешается записывать их, только если АЦП включен (ADEN=1) и если нет регулярного преобразования (ADSTART должен быть равен 0).  - Для управляющих битов, связанных с конфигурацией инжектируемых преобразований, программа имеет право записывать их только в том случае, если АЦП включен (ADEN=1) и если не выполняется инжектируемое преобразование (JADSTART должен быть равен 0).  Программа может записывать управляющие биты ADSTP или JADSTP в регистр ADC\_CR только в том случае, если АЦП включен и осуществляет преобразование, и если нет ожидающего запроса на отключение АЦП (ADSTART или JADSTART должны быть равны 1, а ADDIS - 0)  Программа может записать регистр ADC\_JSQR в любое время, когда АЦП включен (ADEN=1).  Запись в регистр ADC\_JSQR разрешена только тогда, когда JADSTART очищен на 0 (инжектируемое преобразование не выполняется), если только не разрешена контекстная очередь (JQDIS=0 в ADC\_CFGR).  Примечание: Не существует аппаратной защиты для предотвращения этих запрещенных доступов на запись, и АЦП может оказаться в неизвестном состоянии. Чтобы выйти из этой ситуации, АЦП должен быть отключить (очистить ADEN=0, а также все биты регистра ADC\_CR). |
| 25.4.11 Channel selection (SQRx, JSQRx)  There are up to 20 multiplexed channels per ADC:  • 6 fast analog inputs coming from Analog PADs and GPIO pads (ADCx\_INP/INN[0..5])  • Up to 14 slow analog inputs coming from GPIO pads (ADCx\_INP/INN[6..19]).  • The ADCs are connected to 5 internal analog inputs:  – the internal temperature sensor (VSENSE)  – the internal reference voltage (VREFINT)  –the BVAT monitoring channel (VBAT/4)  – DAC internal channels  Refer to Table ADC interconnection in Section 25.4.2: ADC pins and internal signals for the connection of the above internal analog inputs to external ADC pins or internal signals.  It is possible to organize the conversions in two groups: regular and injected. A group  consists of a sequence of conversions that can be done on any channel and in any order.  For instance, it is possible to implement the conversion sequence in the following order:  ADCx\_INP/INN3, ADCx\_INP/INN8, ADCx\_INP/INN2, ADCx\_INP/INN2, ADCx\_INP/INN0,  ADCx\_INP/INN2, ADCx\_INP/INN2, ADCx\_INP/INN15.  • A regular group is composed of up to 16 conversions. The regular channels and their  order in the conversion sequence must be selected in the ADC\_SQRy registers. The  total number of conversions in the regular group must be written in the L[3:0] bits in the ADC\_SQR1 register.  • An injected group is composed of up to 4 conversions. The injected channels and  their order in the conversion sequence must be selected in the ADC\_JSQR register.  The total number of conversions in the injected group must be written in the L[1:0] bits in the ADC\_JSQR register.  ADC\_SQRy registers must not be modified while regular conversions can occur. For this, the ADC regular conversions must be first stopped by writing ADSTP=1 (refer to Section 25.4.18: Stopping an ongoing conversion (ADSTP, JADSTP)).  The software is allowed to modify on-the-fly the ADC\_JSQR register when JADSTART is set to 1 only when the context queue is enabled (JQDIS=0 in ADC\_CFGR register).  Temperature sensor, VREFINT and VBAT internal channels  The internal reference voltage (VREFINT), the temperature sensor (VSENSE), and the VBAT channel are connected to ADC internal channels. Refer to Table ADC interconnection in Section 25.4.2: ADC pins and internal signals for details.  Note: To convert one of the internal analog channels, enable the corresponding analog sources by programming VREFEN, TSEN and VBATEN bits in the ADCx\_CCR registers. | 25.4.11 Выбор канала (SQRx, JSQRx)  На один АЦП приходится до 20 мультиплексированных каналов:  - 6 быстрых аналоговых входов, поступающих с аналоговых PAD и GPIO колодок (ADCx\_INP/INN[0..5])  - до 14 медленных аналоговых входов, поступающих с GPIO-падов (ADCx\_INP/INN[6..19]).  - АЦП подключены к 5 внутренним аналоговым входам:  - внутренний датчик температуры (VSENSE)  - внутреннее опорное напряжение (VREFINT)  -канал мониторинга BVAT (VBAT/4).  - внутренние каналы ЦАП  См. таблицу взаимосвязи АЦП в разделе 25.4.2: Выводы АЦП и внутренние сигналы для подключения вышеуказанных внутренних аналоговых входов к внешним контактам АЦП или внутренним сигналам.  Преобразования можно организовать в две группы: обычные и инжектируемые. Группа состоит из последовательности преобразований, которые могут выполняться на любом канале и в любом порядке.  Например, можно реализовать последовательность преобразований в следующем порядке:  ADCx\_INP/INN3, ADCx\_INP/INN8, ADCx\_INP/INN2, ADCx\_INP/INN2, ADCx\_INP/INN0,  ADCx\_INP/INN2, ADCx\_INP/INN2, ADCx\_INP/INN15.  - Регулярная группа состоит из до 16 преобразований. Регулярные каналы и их порядок в последовательности преобразования должны быть выбраны в регистрах ADC\_SQRy.  общее количество преобразований в регулярной группе должно быть записано в биты L[3:0] в регистре ADC\_SQR1.  - Инжектируемая группа состоит максимум из 4 преобразований. Инжектированные каналы и их порядок в последовательности преобразований должны быть выбраны в регистре ADC\_JSQR.  Общее количество преобразований в инжектируемой группе должно быть записано в биты L[1:0] в регистре ADC\_JSQR.  Регистры ADC\_SQRy не должны быть изменены, пока могут происходить регулярные преобразования. Для этого регулярные преобразования АЦП должны быть сначала остановлены путем записи ADSTP=1 (см. Раздел 25.4.18: Остановка текущего преобразования (ADSTP, JADSTP)).  Программному обеспечению разрешено изменять на лету регистр ADC\_JSQR, когда JADSTART установлен в 1, только если контекстная очередь разрешена (JQDIS=0 в регистре ADC\_CFGR).  Датчик температуры, внутренние каналы VREFINT и VBAT  Внутреннее опорное напряжение (VREFINT), датчик температуры (VSENSE) и канал VBAT подключены к внутренним каналам АЦП. См. таблицу "Взаимосвязь АЦП" в Раздел 25.4.2: Штырьки и внутренние сигналы АЦП для получения подробной информации.  Примечание: Чтобы преобразовать один из внутренних аналоговых каналов, включите соответствующие аналоговые источники, запрограммировав программирования битов VREFEN, TSEN и VBATEN в регистрах ADCx\_CCR. |
| 25.4.12 Channel preselection register (ADC\_PCSEL)  For each channel selected through SQRx or JSQRx, the corresponding ADC\_PCSEL bit  must be previously configured.  This ADC\_PCSEL bit controls the analog switch integrated in the I/O level. The ADC input MUX selects the ADC input according to the SQRx and JSQRx with very high speed, the analog switch integrated in the IO cannot react as fast as ADC mux does. To avoid the delay on analog switch control on IO, it is necessary to pre select the input channels which will be selected in the SQRx, JSQRx.  The selection is based on the VINP[i] of each ADC input. If ADC1 converts the  ADC123\_INP2(VINP[2]) as differential mode, ADC123\_INP6(VINP[6]) also needs to be  selected in ADC\_PCSEL.  Some I/Os are connected to several VINP[i] of the ADCx. The control inputs of the analog switch are ORed with the corresponding ADC\_PCSEL register bits. | 25.4.12 Регистр предварительного выбора канала (ADC\_PCSEL)  Для каждого канала, выбранного через SQRx или JSQRx, соответствующий бит ADC\_PCSEL должен быть предварительно сконфигурирован.  Этот бит ADC\_PCSEL управляет аналоговым переключателем, встроенным в уровень ввода/вывода. Вход АЦП MUX выбирает вход АЦП в соответствии с SQRx и JSQRx с очень высокой скоростью, аналоговый коммутатор, встроенный в уровень ввода/вывода, не может управлять входом АЦП. аналоговый переключатель, встроенный в IO, не может реагировать так же быстро, как ADC mux. Чтобы избежать задержки при управлении аналоговым переключателем на IO, необходимо предварительно выбрать входные каналы, которые будут выбраны в SQRx, JSQRx.  Выбор основан на VINP[i] каждого входа АЦП. Если АЦП1 преобразует ADC123\_INP2(VINP[2]) как дифференциальный режим, ADC123\_INP6(VINP[6]) также должен быть выбрать в ADC\_PCSEL.  Некоторые входы/выходы подключены к нескольким VINP[i] ADCx. Управляющие входы аналогового переключателя объединены в ИЛИ с соответствующими битами регистра ADC\_PCSEL. |
| 25.4.13 Channel-wise programmable sampling time (SMPR1, SMPR2)  Before starting a conversion, the ADC must establish a direct connection between the  voltage source under measurement and the embedded sampling capacitor of the ADC. This sampling time must be enough for the input voltage source to charge the embedded capacitor to the input voltage level.  Each channel can be sampled with a different sampling time which is programmable using the SMP[2:0] bits in the ADC\_SMPR1 and ADC\_SMPR2 registers. It is therefore possible to select among the following sampling time values:  • SMP = 000: 1.5 ADC clock cycles  • SMP = 001: 2.5 ADC clock cycles  • SMP = 010: 8.5 ADC clock cycles  • SMP = 011: 16.5 ADC clock cycles  • SMP = 100: 32.5 ADC clock cycles  • SMP = 101: 64.5 ADC clock cycles  • SMP = 110: 387.5 ADC clock cycles  • SMP = 111: 810.5 ADC clock cycles  The total conversion time is calculated as follows:  TCONV = Sampling time + 7.5 ADC clock cycles  Example:  With Fadc\_ker\_ck = 24 MHz and a sampling time of 1.5 ADC clock cycles (14-bit mode):  TCONV = (1.5 + 7.5) ADC clock cycles = 9 ADC clock cycles = 0.375 µs (14 bit mode for fast channels)  The ADC notifies the end of the sampling phase by setting the status bit EOSMP (only for regular conversion).  Constraints on the sampling time for fast and slow channels  For each channel, SMP[2:0] bits must be programmed to respect a minimum sampling time as specified in the ADC characteristics section of the datasheets.  I/O analog switches voltage booster  The I/O analog switches resistance increases when the VDDA voltage is too low. This  requires to have the sampling time adapted accordingly (refer to the datasheet for electrical characteristics). This resistance can be minimized at low VDDA by enabling an internal voltage booster with BOOSTE bit in the SYSCFG\_PMCR register. | 25.4.13 Программируемое по каналам время выборки (SMPR1, SMPR2)  Перед началом преобразования АЦП должен установить прямое соединение между измеряемым источником напряжения и встроенным конденсатором выборки АЦП. Этот время выборки должно быть достаточным для того, чтобы источник входного напряжения зарядил встроенный конденсатор до конденсатора до уровня входного напряжения.  Каждый канал может иметь разное время выборки, которое программируется с помощью параметра битов SMP[2:0] в регистрах ADC\_SMPR1 и ADC\_SMPR2. Таким образом, можно выбрать одно из следующих значений времени выборки:  - SMP = 000: 1,5 такта АЦП  - SMP = 001: 2,5 такта АЦП  - SMP = 010: 8,5 тактовых циклов АЦП  - SMP = 011: 16,5 тактовых циклов АЦП  - SMP = 100: 32,5 тактовых циклов АЦП  - SMP = 101: 64,5 тактовых циклов АЦП  - SMP = 110: 387,5 тактовых циклов АЦП  - SMP = 111: 810,5 тактовых циклов АЦП  Общее время преобразования рассчитывается следующим образом:  TCONV = время выборки + 7,5 тактовых циклов АЦП.  Пример:  При Fadc\_ker\_ck = 24 МГц и времени выборки 1,5 такта АЦП (14-битный режим):  TCONV = (1,5 + 7,5) тактовых циклов АЦП = 9 тактовых циклов АЦП = 0,375 мкс (14-битный режим для быстрых каналов)  АЦП уведомляет об окончании фазы выборки установкой бита состояния EOSMP (только для обычного преобразования).  Ограничения на время выборки для быстрых и медленных каналов  Для каждого канала биты SMP[2:0] должны быть запрограммированы так, чтобы соблюдалось минимальное время выборки как указано в разделе характеристик АЦП в технических паспортах.  Усилитель напряжения аналоговых переключателей ввода/вывода  Сопротивление аналоговых переключателей ввода/вывода увеличивается, когда напряжение VDDA слишком низкое. Этот  требуется соответствующая адаптация времени выборки (электрические характеристики см. в техническом паспорте). характеристики). Это сопротивление может быть минимизировано при низком напряжении VDDA путем включения внутреннего усилителя напряжения с помощью бита BOOSTE. усилителя напряжения с помощью бита BOOSTE в регистре SYSCFG\_PMCR. |
| 25.4.14 Single conversion mode (CONT=0)  In Single conversion mode, the ADC performs once all the conversions of the channels.  This mode is started with the CONT bit at 0 by either:  • Setting the ADSTART bit in the ADC\_CR register (for a regular channel, with software  trigger selected)  • Setting the JADSTART bit in the ADC\_CR register (for an injected channel, with  software trigger selected)  • External hardware trigger event (for a regular or injected channel)  ADSTART bit or JADSTART bit must be set before triggering an external event.  Inside the regular sequence, after each conversion is complete:  • The converted data are stored into the 32-bit ADC\_DR register  • The EOC (end of regular conversion) flag is set  • An interrupt is generated if the EOCIE bit is set  Inside the injected sequence, after each conversion is complete:  • The converted data are stored into one of the four 32-bit ADC\_JDRy registers  • The JEOC (end of injected conversion) flag is set  • An interrupt is generated if the JEOCIE bit is set  After the regular sequence is complete:  • The EOS (end of regular sequence) flag is set  • An interrupt is generated if the EOSIE bit is set  After the injected sequence is complete:  • The JEOS (end of injected sequence) flag is set  • An interrupt is generated if the JEOSIE bit is set  Then the ADC stops until a new external regular or injected trigger occurs or until bit  ADSTART or JADSTART is set again.  Note: To convert a single channel, program a sequence with a length of 1. | 25.4.14 Режим одиночного преобразования (CONT=0)  В режиме одиночного преобразования АЦП выполняет однократное преобразование всех каналов.  Этот режим запускается с битом CONT в 0 либо путем:  - установкой бита ADSTART в регистре ADC\_CR (для обычного канала, с выбранным программным триггером).  выбран программный триггер)  - установкой бита JADSTART в регистре ADC\_CR (для инжектированного канала, с выбранным программным триггером)  выбран программный триггер)  - Событие внешнего аппаратного триггера (для обычного или инжектируемого канала)  Бит ADSTART или бит JADSTART должны быть установлены перед срабатыванием внешнего события.  Внутри обычной последовательности, после завершения каждого преобразования:  - преобразованные данные сохраняются в 32-битном регистре ADC\_DR  - Устанавливается флаг EOC (конец регулярного преобразования)  - Генерируется прерывание, если установлен бит EOCIE  Внутри инжектируемой последовательности, после завершения каждого преобразования:  - Преобразованные данные сохраняются в одном из четырех 32-битных регистров ADC\_JDRy  - Устанавливается флаг JEOC (конец инжектированного преобразования).  - Генерируется прерывание, если установлен бит JEOCIE.  После завершения регулярной последовательности:  - Устанавливается флаг EOS (конец регулярной последовательности).  - Прерывание генерируется, если установлен бит EOSIE  После завершения инжектированной последовательности:  - Устанавливается флаг JEOS (конец инжектированной последовательности)  - Генерируется прерывание, если установлен бит JEOSIE.  Затем АЦП останавливается до появления нового внешнего регулярного или инжектируемого триггера или до тех пор, пока бит  ADSTART или JADSTART не будет установлен снова.  Примечание: Для преобразования одного канала запрограммируйте последовательность длиной 1. |
| 25.4.15 Continuous conversion mode (CONT=1)  This mode applies to regular channels only.  In continuous conversion mode, when a software or hardware regular trigger event occurs, the ADC performs once all the regular conversions of the channels and then automatically re-starts and continuously converts each conversions of the sequence. This mode is started with the CONT bit at 1 either by external trigger or by setting the ADSTART bit in the ADC\_CR register.  Inside the regular sequence, after each conversion is complete:  • The converted data are stored into the 32-bit ADC\_DR register  • The EOC (end of conversion) flag is set  • An interrupt is generated if the EOCIE bit is set  After the sequence of conversions is complete:  • The EOS (end of sequence) flag is set  • An interrupt is generated if the EOSIE bit is set  Then, a new sequence restarts immediately and the ADC continuously repeats the  conversion sequence.  Note: To convert a single channel, program a sequence with a length of 1.  It is not possible to have both discontinuous mode and continuous mode enabled: it is  forbidden to set both DISCEN=1 and CONT=1.  Injected channels cannot be converted continuously. The only exception is when an injected channel is configured to be converted automatically after regular channels in continuous mode (using JAUTO bit), refer to Auto-injection mode section). | 25.4.15 Режим непрерывного преобразования (CONT=1)  Этот режим применяется только к обычным каналам.  В режиме непрерывного преобразования, когда происходит программное или аппаратное событие регулярного триггера, АЦП выполняет однократное преобразование всех регулярных каналов, а затем автоматически повторно запускает и непрерывно преобразует каждое преобразование последовательности. Этот режим запускается с битом CONT в 1 либо внешним триггером, либо установкой бита ADSTART в регистре ADC\_CR.  Внутри обычной последовательности, после завершения каждого преобразования:  - преобразованные данные сохраняются в 32-битном регистре ADC\_DR  - Устанавливается флаг EOC (конец преобразования).  - Генерируется прерывание, если установлен бит EOCIE.  После завершения последовательности преобразований:  - Устанавливается флаг EOS (конец последовательности).  - Генерируется прерывание, если установлен бит EOSIE.  Затем немедленно начинается новая последовательность, и АЦП непрерывно повторяет последовательность преобразования.  Примечание: Для преобразования одного канала запрограммируйте последовательность длиной 1.  Невозможно включить как прерывистый, так и непрерывный режим: нельзя  запрещено устанавливать и DISCEN=1, и CONT=1.  Инжектированные каналы не могут быть преобразованы непрерывно. Единственным исключением является случай, когда инжектируемый канал настроен на автоматическое преобразование после обычных каналов в непрерывном режиме (с помощью бита JAUTO). (с помощью бита JAUTO), см. раздел "Режим автоинжекции"). |
| **25.4.16** Starting conversions (ADSTART, JADSTART)  Software starts ADC regular conversions by setting ADSTART=1.  When ADSTART is set, the conversion starts:  • Immediately: if EXTEN = 0x0 (software trigger)  • At the next active edge of the selected regular hardware trigger: if EXTEN /= 0x0  Software starts ADC injected conversions by setting JADSTART=1.  When JADSTART is set, the conversion starts:  • Immediately, if JEXTEN = 0x0 (software trigger)  • At the next active edge of the selected injected hardware trigger: if JEXTEN /= 0x0  Note: In auto-injection mode (JAUTO=1), use ADSTART bit to start the regular conversions followed by the auto-injected conversions (JADSTART must be kept cleared).  ADSTART and JADSTART also provide information on whether any ADC operation is  currently ongoing. It is possible to re-configure the ADC while ADSTART=0 and  JADSTART=0 are both true, indicating that the ADC is idle.  ADSTART is cleared by hardware:  • In single mode with software trigger (CONT=0, EXTEN=0x0)  – at any end of conversion sequence (EOS =1)  • In discontinuous mode with software trigger (CONT=0, DISCEN=1, EXTEN=0x0)  – at end of conversion (EOC=1)  • In all other cases (CONT=x, EXTEN=x)  – after execution of the ADSTP procedure asserted by the software.  Note: In continuous mode (CONT=1), ADSTART is not cleared by hardware with the assertion of EOS because the sequence is automatically relaunched.  When a hardware trigger is selected in single mode (CONT=0 and EXTEN /=0x00),  ADSTART is not cleared by hardware with the assertion of EOS to help the software which does not need to reset ADSTART again for the next hardware trigger event. This ensures that no further hardware triggers are missed.  JADSTART is cleared by hardware:  • in single mode with software injected trigger (JEXTEN=0x0)  – at any end of injected conversion sequence (JEOS assertion) or at any end of  sub-group processing if JDISCEN=1  • in all cases (JEXTEN=x)  – after execution of the JADSTP procedure asserted by the software.  Note: When the software trigger is selected, ADSTART bit should not be set if the EOC flag is still high. | 25.4.16 Запуск преобразований (ADSTART, JADSTART)  Программное обеспечение запускает регулярные преобразования АЦП, установив ADSTART=1.  Когда ADSTART установлен, начинается преобразование:  - Немедленно: если EXTEN = 0x0 (программный триггер)  - По следующему активному фронту выбранного регулярного аппаратного триггера: если EXTEN /= 0x0.  Программный запуск инжектированных преобразований АЦП осуществляется установкой JADSTART=1.  Когда JADSTART установлен, начинается преобразование:  - Немедленно, если JEXTEN = 0x0 (программный триггер)  - По следующему активному фронту выбранного инжектируемого аппаратного триггера: если JEXTEN /= 0x0  Примечание: В режиме автоинжекции (JAUTO=1) используйте бит ADSTART для запуска обычных преобразований за которым следуют автоинжектируемые преобразования (JADSTART должен быть очищен).  ADSTART и JADSTART также предоставляют информацию о том, выполняется ли в данный момент какая-либо операция АЦП.  в данный момент. Можно переконфигурировать АЦП, пока ADSTART=0 и  JADSTART=0, что указывает на то, что АЦП простаивает.  ADSTART очищается аппаратно:  - в одиночном режиме с программным триггером (CONT=0, EXTEN=0x0)  - в любом конце последовательности преобразования (EOS =1).  - В прерывистом режиме с программным триггером (CONT=0, DISCEN=1, EXTEN=0x0)  - в конце преобразования (EOC=1)  - Во всех остальных случаях (CONT=x, EXTEN=x)  - после выполнения процедуры ADSTP, утверждаемой программным обеспечением.  Примечание: В непрерывном режиме (CONT=1), ADSTART не очищается аппаратно с утверждением EOS, поскольку последовательность автоматически перезапускается.  Когда аппаратный триггер выбран в одиночном режиме (CONT=0 и EXTEN /=0x00),  ADSTART не очищается аппаратными средствами при утверждении EOS, чтобы помочь программному обеспечению, которому не нужно снова сбрасывать ADSTART для следующего события аппаратного триггера. Это гарантирует что никакие другие аппаратные триггеры не будут пропущены.  JADSTART сбрасывается аппаратно:  - в одиночном режиме с программным триггером (JEXTEN=0x0)  - в любом конце инжектированной последовательности преобразования (утверждение JEOS) или в любом конце  обработки подгруппы, если JDISCEN=1  - во всех случаях (JEXTEN=x)  - после выполнения процедуры JADSTP, утверждаемой программным обеспечением.  Примечание: Если выбран программный триггер, бит ADSTART не должен быть установлен, если флаг EOC все еще находится на высоком уровне. высоким. |
| 25.4.17 Timing  The elapsed time between the start of a conversion and the end of conversion is the sum of the configured sampling time plus the successive approximation time depending on data resolution: | 25.4.17 Время  Истекшее время между началом преобразования и его окончанием представляет собой сумму следующих величин настроенного времени выборки плюс время последовательного приближения в зависимости от данных разрешение: |
| 25.4.18 Stopping an ongoing conversion (ADSTP, JADSTP)  The software can decide to stop regular conversions ongoing by setting ADSTP=1 and injected conversions ongoing by setting JADSTP=1.  Stopping conversions will reset the ongoing ADC operation. Then the ADC can be reconfigured (ex: changing the channel selection or the trigger) ready for a new operation.  Note that it is possible to stop injected conversions while regular conversions are still operating and vice-versa. This allows, for instance, re-configuration of the injected conversion sequence and triggers while regular conversions are still operating (and vice-versa).  When the ADSTP bit is set by software, any ongoing regular conversion is aborted with partial result discarded (ADC\_DR register is not updated with the current conversion).  When the JADSTP bit is set by software, any ongoing injected conversion is aborted with partial result discarded (ADC\_JDRy register is not updated with the current conversion).  The scan sequence is also aborted and reset (meaning that relaunching the ADC would re-start a new sequence).  Once this procedure is complete, bits ADSTP/ADSTART (in case of regular conversion), or JADSTP/JADSTART (in case of injected conversion) are cleared by hardware and the  software must poll ADSTART (or JADSTART) until the bit is reset before assuming the ADC is completely stopped.  Note: In auto-injection mode (JAUTO=1), setting ADSTP bit aborts both regular and injected conversions (JADSTP must not be used). | 25.4.18 Остановка текущего преобразования (ADSTP, JADSTP)  Программное обеспечение может принять решение об остановке обычного преобразования, установив ADSTP=1 и  инжектированные преобразования, установив JADSTP=1.  Остановка преобразований приведет к сбросу текущей операции АЦП. После этого АЦП может быть  переконфигурировать (например, изменить выбор канала или триггера), чтобы он был готов к новой работе.  Обратите внимание, что можно остановить инжектированные преобразования, в то время как обычные преобразования продолжают работать.  и наоборот. Это позволяет, например, переконфигурировать последовательность инжектируемых преобразований и триггеры, в то время как обычные преобразования продолжают работать.  последовательность инжектируемых преобразований и триггеров, в то время как обычные преобразования продолжают работать (и наоборот).  Когда бит ADSTP установлен программно, любое текущее обычное преобразование прерывается с отбрасыванием частичного результата (ADC). частичный результат отбрасывается (регистр ADC\_DR не обновляется текущим преобразованием).  Если программно установлен бит JADSTP, любое текущее инжектированное преобразование прерывается с отбрасыванием части результата (регистр ADC\_DR не обновляется текущим преобразованием). частичный результат отбрасывается (регистр ADC\_JDRy не обновляется для текущего преобразования).  Последовательность сканирования также прерывается и сбрасывается (это означает, что повторный запуск АЦП заново запустит новую последовательность).  После завершения этой процедуры биты ADSTP/ADSTART (в случае обычного преобразования) или JADSTP/JADSTART (в случае обычного преобразования) очищаются аппаратными средствами, а программное обеспечение должно опросить ADSTART (или JADSTART).  программное обеспечение должно опрашивать ADSTART (или JADSTART) до тех пор, пока бит не будет сброшен, прежде чем считать, что АЦП полностью остановлен.  Примечание: В режиме автоинжекции (JAUTO=1) установка бита ADSTP прерывает как обычное, так и инжектированное преобразование (JADSTP не должен использоваться). преобразования (JADSTP не должен использоваться). |
| 25.4.19 Conversion on external trigger and trigger polarity (EXTSEL, EXTEN,  JEXTSEL, JEXTEN)  A conversion or a sequence of conversions can be triggered either by software or by an external event (e.g. timer capture, input pins). If the EXTEN[1:0] control bits (for a regular conversion) or JEXTEN[1:0] bits (for an injected conversion) are different from 0b00, then external events are able to trigger a conversion with the selected polarity.  When the Injected Queue is enabled (bit JQDIS=0), injected software triggers are not possible.  The regular trigger selection is effective once software has set bit ADSTART=1 and the injected trigger selection is effective once software has set bit JADSTART=1.  Any hardware triggers which occur while a conversion is ongoing are ignored.  • If bit ADSTART=0, any regular hardware triggers which occur are ignored.  • If bit JADSTART=0, any injected hardware triggers which occur are ignored.  Table 206 provides the correspondence between the EXTEN[1:0] and JEXTEN[1:0] values and the trigger polarity. | 25.4.19 Преобразование внешнего триггера и полярности триггера (EXTSEL, EXTEN,  JEXTSEL, JEXTEN)  Преобразование или последовательность преобразований могут быть запущены либо программно, либо по внешним событием (например, захватом таймера, входными контактами). Если управляющие биты EXTEN[1:0] (для обычного преобразования) или биты управления JEXTEN[1:0] (для обычного преобразования) отличны от 0b00, то внешние события могут вызвать преобразование с выбранной полярностью.  Если очередь инжектирования включена (бит JQDIS=0), программные триггеры инжектирования невозможны.  Выбор обычного триггера вступает в силу, как только программа установит бит ADSTART=1 и выбор инжектируемого триггера действует, как только программа установит бит JADSTART=1.  Любые аппаратные триггеры, возникающие в процессе преобразования, игнорируются.  - Если бит ADSTART=0, все обычные аппаратные триггеры игнорируются.  - Если бит JADSTART=0, любые инжектированные аппаратные триггеры игнорируются.  В таблице 206 приведено соответствие между значениями EXTEN[1:0] и JEXTEN[1:0] и полярностью триггера. |
| 25.4.20 Injected channel management  Triggered injection mode  To use triggered injection, the JAUTO bit in the ADC\_CFGR register must be cleared.  1. Start the conversion of a group of regular channels either by an external trigger or by setting the ADSTART bit in the ADC\_CR register.  2. If an external injected trigger occurs, or if the JADSTART bit in the ADC\_CR register is set during the conversion of a regular group of channels, the current conversion is  reset and the injected channel sequence switches are launched (all the injected  channels are converted once).  3. Then, the regular conversion of the regular group of channels is resumed from the last interrupted regular conversion.  4. If a regular event occurs during an injected conversion, the injected conversion is not interrupted but the regular sequence is executed at the end of the injected sequence.  Figure 149 shows the corresponding timing diagram.  Note: When using triggered injection, one must ensure that the interval between trigger events is longer than the injection sequence. For instance, if the sequence length is 20 ADC clock cycles (that is two conversions with a sampling time of 1.5 clock periods), the minimum interval between triggers must be 21 ADC clock cycles.  Auto-injection mode  If the JAUTO bit in the ADC\_CFGR register is set, then the channels in the injected group are automatically converted after the regular group of channels. This can be used to convert a sequence of up to 20 conversions programmed in the ADC\_SQRy and ADC\_JSQR registers.  In this mode, the ADSTART bit in the ADC\_CR register must be set to start regular  conversions, followed by injected conversions (JADSTART must be kept cleared). Setting the ADSTP bit aborts both regular and injected conversions (JADSTP bit must not be used).  In this mode, external trigger on injected channels must be disabled.  If the CONT bit is also set in addition to the JAUTO bit, regular channels followed by injected channels are continuously converted.  Note: It is not possible to use both the auto-injected and discontinuous modes simultaneously.  When the DMA is used for exporting regular sequencer’s data in JAUTO mode, it is  necessary to program it in circular mode (CIRC bit set in DMA\_CCRx register). If the CIRC bit is reset (single-shot mode), the JAUTO sequence will be stopped upon DMA Transfer Complete event. | 25.4.20 Управление инжектированным каналом  Режим триггерного впрыска  Для использования триггерного впрыска бит JAUTO в регистре ADC\_CFGR должен быть очищен.  1. Запустите преобразование группы обычных каналов либо внешним триггером, либо путем установкой бита ADSTART в регистре ADC\_CR.  2. Если произошел внешний инжектированный триггер или если бит JADSTART в регистре ADC\_CR установлен во время преобразования обычной группы каналов, текущее преобразование сбрасывается  сбрасывается, и запускаются переключатели последовательности инжектируемых каналов (все инжектируемые каналы преобразуются один раз).  3. Затем регулярное преобразование регулярной группы каналов возобновляется с последнего прерванного регулярного преобразования.  4. Если регулярное событие происходит во время инжектированного преобразования, то инжектированное преобразование не прерывается прерывается, но регулярная последовательность выполняется в конце инжектированной последовательности.  На рисунке 149 показана соответствующая временная диаграмма.  Примечание: При использовании триггерной инъекции необходимо обеспечить, чтобы интервал между событиями триггера был больше, чем последовательность впрыска. Например, если длина последовательности составляет 20 тактов АЦП циклов (то есть два преобразования со временем выборки 1,5 тактовых периода), минимальный интервал между срабатываниями должен составлять 21 такт АЦП.  Режим автоинжекции  Если бит JAUTO в регистре ADC\_CFGR установлен, то каналы в инжектируемой группе автоматически преобразуются после обычной группы каналов. Это может быть использовано для преобразования последовательности до 20 преобразований, запрограммированных в ADC\_SQRy и ADC\_JSQR регистрах.  В этом режиме бит ADSTART в регистре ADC\_CR должен быть установлен для запуска обычного преобразования, за которым следует инжектированное преобразование.  преобразования, за которым следуют инжектированные преобразования (JADSTART должен быть сброшен). Установка бита ADSTP прерывает как обычное, так и инжектированное преобразование (бит JADSTP не должен использоваться).  В этом режиме внешний триггер на инжектированных каналах должен быть отключен.  Если в дополнение к биту JAUTO установлен бит CONT, то обычные каналы, за которыми следуют инжектированные, преобразуются непрерывно. каналы преобразуются непрерывно.  Примечание: Невозможно одновременно использовать и автоинжектируемый, и прерывистый режимы.  Когда DMA используется для экспорта данных регулярного секвенсора в режиме JAUTO, необходимо  необходимо запрограммировать его в круговом режиме (бит CIRC установлен в регистре DMA\_CCRx). Если бит CIRC бит сброшен (однократный режим), последовательность JAUTO будет остановлена по событию DMA Transfer Transfer Complete event. |
| 25.4.21 Discontinuous mode (DISCEN, DISCNUM, JDISCEN)  Regular group mode  This mode is enabled by setting the DISCEN bit in the ADC\_CFGR register.  It is used to convert a short sequence (sub-group) of n conversions (n ≤ 8) that is part of the sequence of conversions selected in the ADC\_SQRy registers. The value of n is specified by writing to the DISCNUM[2:0] bits in the ADC\_CFGR register.  When an external trigger occurs, it starts the next n conversions selected in the ADC\_SQR registers until all the conversions in the sequence are done. The total sequence length is defined by the L[3:0] bits in the ADC\_SQR1 register.  Example:  • DISCEN=1, n=3, channels to be converted = 1, 2, 3, 6, 7, 8, 9, 10, 11  – 1st trigger: channels converted are 1, 2, 3 (an EOC event is generated at each conversion).  – 2nd trigger: channels converted are 6, 7, 8 (an EOC event is generated at each conversion).  – 3rd trigger: channels converted are 9, 10, 11 (an EOC event is generated at each conversion) and an EOS event is generated after the conversion of channel 11.  – 4th trigger: channels converted are 1, 2, 3 (an EOC event is generated at each conversion).  –...  • DISCEN=0, channels to be converted = 1, 2, 3, 6, 7, 8, 9, 10,11  – 1st trigger: the complete sequence is converted: channel 1, then 2, 3, 6, 7, 8, 9, 10  and 11. Each conversion generates an EOC event and the last one also generates an EOS event.  – all the next trigger events will relaunch the complete sequence.  Note: When a regular group is converted in discontinuous mode, no rollover occurs (the last subgroup of the sequence can have less than n conversions).  When all subgroups are converted, the next trigger starts the conversion of the first subgroup. In the example above, the 4th trigger reconverts the channels 1, 2 and 3 in the 1st subgroup.  It is not possible to have both discontinuous mode and continuous mode enabled. In this case (if DISCEN=1, CONT=1), the ADC behaves as if continuous mode was disabled.  **Injected group mode**  This mode is enabled by setting the JDISCEN bit in the ADC\_CFGR register. It converts the sequence selected in the ADC\_JSQR register, channel by channel, after an external injected trigger event. This is equivalent to discontinuous mode for regular channels where  ‘n’ is fixed to 1.  When an external trigger occurs, it starts the next channel conversions selected in the ADC\_JSQR registers until all the conversions in the sequence are done. The total sequence  length is defined by the JL[1:0] bits in the ADC\_JSQR register.  Example:  • JDISCEN=1, channels to be converted = 1, 2, 3  – 1st trigger: channel 1 converted (a JEOC event is generated)  – 2nd trigger: channel 2 converted (a JEOC event is generated)  – 3rd trigger: channel 3 converted and a JEOC event + a JEOS event are generated  –...  Note: When all injected channels have been converted, the next trigger starts the conversion of the first injected channel. In the example above, the 4th trigger reconverts the 1st injected channel 1.  It is not possible to use both auto-injected mode and discontinuous mode simultaneously:  the bits DISCEN and JDISCEN must be kept cleared by software when JAUTO is set. | 25.4.21 Прерывистый режим (DISCEN, DISCNUM, JDISCEN)  Режим обычной группы  Этот режим включается установкой бита DISCEN в регистре ADC\_CFGR.  Он используется для преобразования короткой последовательности (подгруппы) из n преобразований (n ≤ 8), которая является частью последовательности преобразований, выбранной в регистрах ADC\_SQRy. Значение n задается путем записи в биты DISCNUM[2:0] в регистре ADC\_CFGR.  При возникновении внешнего триггера запускается следующее n преобразований, выбранных в регистрах ADC\_SQR регистров, пока не будут выполнены все преобразования в последовательности. Общая длина последовательности определяется битами L[3:0] в регистре ADC\_SQR1.  Пример:  - DISCEN=1, n=3, преобразуемые каналы = 1, 2, 3, 6, 7, 8, 9, 10, 11.  - 1-й триггер: преобразуются каналы 1, 2, 3 (событие EOC генерируется при каждом  преобразование).  - 2-й триггер: преобразуются каналы 6, 7, 8 (при каждом преобразовании генерируется событие EOC).  преобразование).  - 3-й триггер: преобразованы каналы 9, 10, 11 (событие EOC генерируется при каждом преобразовании) и событие EOS  и событие EOS генерируется после преобразования канала 11.  - 4-й триггер: преобразованы каналы 1, 2, 3 (событие EOC генерируется при каждом преобразовании).  -...  - DISCEN=0, преобразуемые каналы = 1, 2, 3, 6, 7, 8, 9, 10,11  - 1-е срабатывание: преобразуется полная последовательность: канал 1, затем 2, 3, 6, 7, 8, 9, 10 и 11. Каждое преобразование генерирует событие EOC, а последнее преобразование также генерирует событие EOS.  - Все следующие триггерные события перезапускают всю последовательность.  Примечание: Когда обычная группа преобразуется в прерывистом режиме, не происходит переворачивания (последняя подгруппа последовательности может иметь менее n преобразований).  Когда все подгруппы преобразованы, следующий триггер запускает преобразование первой подгруппы. В приведенном выше примере 4-й триггер повторно преобразует каналы 1, 2 и 3 в 1-й подгруппе.  Невозможно включить одновременно прерывистый и непрерывный режимы. В этом случае (если DISCEN=1, CONT=1), АЦП ведет себя так, как если бы непрерывный режим был отключен.  **Режим инжектированной группы**  Этот режим включается установкой бита JDISCEN в регистре ADC\_CFGR. Он преобразует последовательность, выбранную в регистре ADC\_JSQR, канал за каналом, после внешнего инжектированного триггера. Это эквивалентно прерывистому режиму для обычных каналов, где 'n' фиксировано на 1.  Когда возникает внешний триггер, он запускает преобразования следующего канала, выбранного в ADC\_JSQR, пока не будут выполнены все преобразования в последовательности. Общая последовательность определяется битами JL[1:0] в регистре ADC\_JSQR.  Пример:  - JDISCEN=1, преобразуемые каналы = 1, 2, 3  - 1-й триггер: канал 1 преобразован (генерируется событие JEOC)  - 2-й триггер: канал 2 преобразован (генерируется событие JEOC)  - 3-й триггер: канал 3 преобразован и генерируется событие JEOC + событие JEOS  -...  Примечание: Когда все инжектированные каналы преобразованы, следующий триггер запускает преобразование первого инжектированного канала. В приведенном выше примере 4-й триггер переконвертирует 1-й инжектированный канал 1.  Невозможно одновременно использовать режим автоинжекции и прерывистый режим:  биты DISCEN и JDISCEN должны быть программно очищены, когда JAUTO установлен. |
|  |  |
|  |  |
| 25.4.22 Queue of context for injected conversions  A queue of context is implemented to anticipate up to 2 contexts for the next injected sequence of conversions. JQDIS bit of ADC\_CFGR register must be reset to enable this feature.  Only hardware-triggered conversions are possible when the context queue is enabled.  This context consists of:  • Configuration of the injected triggers (bits JEXTEN[1:0] and JEXTSEL[4:0] in ADC\_JSQR register)  • Definition of the injected sequence (bits JSQx[4:0] and JL[1:0] in ADC\_JSQR register)  All the parameters of the context are defined into a single register ADC\_JSQR and this register implements a queue of 2 buffers, allowing the bufferization of up to 2 sets of parameters:  • The ADC\_JSQR register can be written at any moment even when injected conversions are ongoing.  • Each data written into the JSQR register is stored into the Queue of context.  • At the beginning, the Queue is empty and the first write access into the JSQR register immediately changes the context and the ADC is ready to receive injected triggers.  • Once an injected sequence is complete, the Queue is consumed and the context changes according to the next JSQR parameters stored in the Queue. This new context is applied for the next injected sequence of conversions.  • A Queue overflow occurs when writing into register JSQR while the Queue is full. This overflow is signaled by the assertion of the flag JQOVF. When an overflow occurs, the write access of JSQR register which has created the overflow is ignored and the queue of context is unchanged. An interrupt can be generated if bit JQOVFIE is set.  • Two possible behaviors are possible when the Queue becomes empty, depending on the value of the control bit JQM of register ADC\_CFGR:  – If JQM=0, the Queue is empty just after enabling the ADC, but then it can never be empty during run operations: the Queue always maintains the last active context and any further valid start of injected sequence will be served according to the last active context.  – If JQM=1, the Queue can be empty after the end of an injected sequence or if the Queue is flushed. When this occurs, there is no more context in the queue and  hardware triggers are disabled. Therefore, any further hardware injected triggers  are ignored until the software re-writes a new injected context into JSQR register.  • Reading JSQR register returns the current JSQR context which is active at that moment. When the JSQR context is empty, JSQR is read as 0x0000.  • The Queue is flushed when stopping injected conversions by setting JADSTP=1 or  when disabling the ADC by setting ADDIS=1:  – If JQM=0, the Queue is maintained with the last active context.  – If JQM=1, the Queue becomes empty and triggers are ignored.  Note: When configured in discontinuous mode (bit JDISCEN=1), only the last trigger of the injected sequence changes the context and consumes the Queue.The 1st trigger only consumes the queue but others are still valid triggers as shown by the discontinuous mode example below (length = 3 for both contexts):  • 1st trigger, discontinuous. Sequence 1: context 1 consumed, 1st conversion carried out  • 2nd trigger, disc. Sequence 1: 2nd conversion.  • 3rd trigger, discontinuous. Sequence 1: 3rd conversion.  • 4th trigger, discontinuous. Sequence 2: context 2 consumed, 1st conversion carried out.  • 5th trigger, discontinuous. Sequence 2: 2nd conversion.  • 6th trigger, discontinuous. Sequence 2: 3rd conversion.  Note: When queue of context enabled (bit JQDIS=0), only hardware trigger can be used.  Behavior when changing the trigger or sequence context  The Figure 150 and Figure 151 show the behavior of the context Queue when changing the sequence or the triggers.  **…пропущены пару страниц (см мануал)…**  Queue of context: Starting the ADC with an empty queue  The following procedure must be followed to start ADC operation with an empty queue, in case the first context is not known at the time the ADC is initialized. This procedure is only applicable when JQM bit is reset:  5. Write a dummy JSQR with JEXTEN not equal to 0 (otherwise triggering a software  conversion)  6. Set JADSTART  7. Set JADSTP  8. Wait until JADSTART is reset  9. Set JADSTART.  Disabling the queue  It is possible to disable the queue by setting bit JQDIS=1 into the ADC\_CFGR register.  Queue of context: Programming of the register ADC\_JSQR  When the injected conversion queue of context is enabled (JQDIS=0), the ADC\_JSQR must be programmed at one register write access. As JL[1:0] register define the number of the injected sequence, corresponding JSQ1 to JSQ4 must be written at same time. If ADC\_JSQR is reprogrammed before the injected conversion start, reprogrammed data is put on the queue. When queue of context is empty, ADC\_JSQR read back as 0x0000.  Register access should not use the ‘read modify write’ sequence.  When ADC\_JSQR is programmed when already 2 contexts are queued, it will raise JQOVF flag and generate the interrupt. | 25.4.22 Очередь режимов для инжектированных преобразований  Очередь режимов реализована для прогнозирования до 2 режимов для следующей инжектированной последовательности преобразований. Для включения этой функции бит JQDIS регистра ADC\_CFGR должен быть сброшен.  Только аппатно-триггерные преобразования возможны если очередь режимов на преобразование включена.  Этот режим состоит из:  - Конфигурация инжектируемых триггеров (биты JEXTEN[1:0] и JEXTSEL[4:0] в регистре ADC\_JSQR)  - Определение инжектируемой последовательности (биты JSQx[4:0] и JL[1:0] в регистре ADC\_JSQR).  Все параметры режима определены в одном регистре ADC\_JSQR и этот регистр реализует очередь из 2 буферов, позволяя буферизовать до 2 наборов  параметров:  - Регистр ADC\_JSQR может быть записан в любой момент, даже когда инжектированные преобразования продолжаются.  - Каждые данные, записанные в регистр JSQR, сохраняются в очереди режимов.  - В начале очередь пуста, и первый доступ к записи в регистр JSQR немедленно изменяет режим, и АЦП готов к приему инжектированных триггеров.  - После завершения инжектируемой последовательности очередь заполняется, а объект изменяется в соответствии со следующими параметрами JSQR, хранящимися в очереди. Этот новый объект применяется для следующей инжектированной последовательности преобразований.  - Переполнение очереди происходит при записи в регистр JSQR, когда очередь заполнена. Это переполнение сигнализируется установкой флага JQOVF. Когда происходит переполнение, то запись в регистр JSQR, создавшая переполнение, игнорируется, и очередь объектов не изменяется. Прерывание может быть сгенерировано, если установлен бит JQOVFIE.  - Возможны два варианта поведения, когда очередь становится пустой, в зависимости от значения управляющего бита JQM регистра ADC\_CFGR:  - Если JQM=0, очередь становится пустой сразу после включения АЦП, но затем она никогда не может быть пустой во время выполнения операций: очередь всегда содержит последний активный объект и любой последующий допустимый запуск инжектированной последовательности будет обслуживаться в соответствии с последним активным объектом.  - Если JQM=1, очередь может быть пустой после окончания инжектируемой последовательности или если очередь очищена. Когда это происходит, в очереди больше нет объектов и  аппаратные триггеры отключены. Таким образом, любые дальнейшие аппаратные триггеры игнорируются до тех пор, пока программное обеспечение не перезапишет новый инжектированный режим в регистр JSQR.  - Чтение регистра JSQR возвращает текущий режим JSQR, который активен в данный момент. Когда контекст JSQR пуст, JSQR читается как 0x0000.  - Очередь очищается при остановке инжектированных преобразований установкой JADSTP=1 или при отключении АЦП установкой ADDIS=1:  - Если JQM=0, очередь сохраняется с последним активным контекстом.  - Если JQM=1, очередь становится пустой и триггеры игнорируются.  Примечание: При конфигурации в прерывистом режиме (бит JDISCEN=1), только последний триггер из изменяет режим и расходует очередь. 1-е срабатывание только потребляет очередь, но остальные триггеры остаются действительными, как показано на примере прерывистого режима пример ниже (длина = 3 для обоих режимов):  - 1-й триггер, прерывистый. Последовательность 1: режим 1 потреблен, выполнено 1-е преобразование  - 2-й триггер, диск. Последовательность 1: 2-е преобразование.  - 3-й триггер, прерывистый. Последовательность 1: 3-е преобразование.  - 4-й триггер, прерывистый. Последовательность 2: контекст 2 израсходован, выполнено 1-е преобразование.  - 5-й триггер, прерывистый. Последовательность 2: 2-е преобразование.  - 6-й триггер, прерывистый. Последовательность 2: 3-е преобразование.  Примечание: Если очередь режимов разрешена (бит JQDIS=0), можно использовать только аппаратный триггер.  Поведение при изменении контекста триггера или последовательности  На Рисунке 150 и Рисунке 151 показано поведение очереди контекста при изменении последовательности или триггеров.  Очередь контекста: Запуск АЦП с пустой очередью  Для запуска работы АЦП с пустой очередью необходимо выполнить следующую процедуру, если если первый контекст не известен в момент инициализации АЦП. Эта процедура применима только применима только в том случае, если бит JQM сброшен:  5. Записать фиктивный JSQR с JEXTEN не равным 0 (в противном случае запускается программное  преобразование)  6. Установите JADSTART  7. Установите JADSTP  8. Подождите, пока JADSTART не будет сброшен  9. Установите JADSTART.  Отключение очереди  Очередь можно отключить, установив бит JQDIS=1 в регистре ADC\_CFGR.  Очередь контекста: Программирование регистра ADC\_JSQR  Когда очередь контекста инжектированного преобразования включена (JQDIS=0), ADC\_JSQR должен быть быть запрограммирован при одном доступе к записи в регистр. Поскольку регистр JL[1:0] определяет номер инжектируемой очереди, соответствующие JSQ1 - JSQ4 должны быть записаны одновременно. Если ADC\_JSQR перепрограммируется до начала инжектируемого преобразования, перепрограммированные данные помещаются в очередь. Когда очередь контекста пуста, ADC\_JSQR считывается обратно как 0x0000.  Доступ к регистру не должен использовать последовательность 'чтение-модификация-запись'.  Если ADC\_JSQR программируется, когда в очереди уже 2 контекста, это приведет к поднятию флага JQOVF и сгенерирует прерывание. |
| 25.4.23 Programmable resolution (RES) - fast conversion mode  It is possible to perform faster conversion by reducing the ADC resolution.  The resolution can be configured to be either 16, 14, 12, 10, 8 bits by programming the control bits RES[1:0]. Figure 166, Figure 167, Figure 168 and Figure 169 show the conversion result format with respect to the resolution as well as to the data alignment.  Lower resolution allows faster conversion time for applications where high-data precision is not required. It reduces the conversion time spent by the successive approximation steps according to Table 210. | 25.4.23 Программируемое разрешение (RES) - режим быстрого преобразования  Можно выполнить более быстрое преобразование, уменьшив разрешение АЦП.  Разрешение может быть настроено на 16, 14, 12, 10, 8 бит путем программирования управляющих битов RES[1:0]. На рисунке 166, рисунке 167, рисунке 168 и рисунке 169 показан формат результата преобразования в зависимости от разрешения, а также от выравнивания данных.  Более низкое разрешение позволяет ускорить время преобразования для приложений, где высокая точность данных не требуется. Это уменьшает время преобразования, затрачиваемое на последовательные шаги аппроксимации в соответствии с таблицей 210. |
| 25.4.24 End of conversion, end of sampling phase (EOC, JEOC, EOSMP)  The ADC notifies the application for each end of regular conversion (EOC) event and each injected conversion (JEOC) event.  The ADC sets the EOC flag as soon as a new regular conversion data is available in the  ADC\_DR register. An interrupt can be generated if bit EOCIE is set. EOC flag is cleared by the software either by writing 1 to it or by reading ADC\_DR.  The ADC sets the JEOC flag as soon as a new injected conversion data is available in one of the ADC\_JDRy register. An interrupt can be generated if bit JEOCIE is set. JEOC flag is cleared by the software either by writing 1 to it or by reading the corresponding ADC\_JDRy register.  The ADC also notifies the end of Sampling phase by setting the status bit EOSMP (for  regular conversions only). EOSMP flag is cleared by software by writing 1 to it. An interrupt can be generated if bit EOSMPIE is set. | 25.4.24 Конец конверсии, конец фазы выборки (EOC, JEOC, EOSMP)  АЦП уведомляет приложение о каждом событии окончания обычного преобразования (EOC) и каждом инжектированного преобразования (JEOC).  АЦП устанавливает флаг EOC, как только новые данные регулярного преобразования становятся доступными в регистре ADC\_DR. Прерывание может быть сгенерировано, если установлен бит EOCIE. Флаг EOC очищается программное обеспечение либо путем записи в него 1, либо путем чтения ADC\_DR.  АЦП устанавливает флаг JEOC, как только новые введенные данные преобразования становятся доступными в одном из регистров ADC\_JDRy. регистре ADC\_JDRy. Если установлен бит JEOCIE, может быть сгенерировано прерывание. Флаг JEOC снимается программным обеспечением либо путем записи в него 1, либо путем чтения соответствующего регистра ADC\_JDRy регистра.  АЦП также уведомляет об окончании фазы выборки установкой бита состояния EOSMP (только для только для регулярных преобразований). Флаг EOSMP снимается программно путем записи в него 1. Прерывание может быть сгенерировано, если установлен бит EOSMPIE. |
| 25.4.25 End of conversion sequence (EOS, JEOS)  The ADC notifies the application for each end of regular sequence (EOS) and for each end of injected sequence (JEOS) event.  The ADC sets the EOS flag as soon as the last data of the regular conversion sequence is available in the ADC\_DR register. An interrupt can be generated if bit EOSIE is set. EOS flag is cleared by the software either by writing 1 to it.  The ADC sets the JEOS flag as soon as the last data of the injected conversion sequence is complete. An interrupt can be generated if bit JEOSIE is set. JEOS flag is cleared by the software either by writing 1 to it. | 25.4.25 Конец последовательности преобразования (EOS, JEOS)  АЦП уведомляет приложение о каждом событии конца регулярной последовательности (EOS) и о каждом событии конца инжектированной последовательности (JEOS).  АЦП устанавливает флаг EOS, как только последние данные регулярной последовательности преобразования становятся доступны в регистре ADC\_DR. Если установлен бит EOSIE, может быть сгенерировано прерывание. Флаг EOS очищается программным обеспечением либо путем записи в него 1.  АЦП устанавливает флаг JEOS, как только завершаются последние данные введенной последовательности преобразования. Если бит JEOSIE установлен, может быть сгенерировано прерывание. Флаг JEOS очищается программным обеспечением либо путем записи в него 1. |
| 25.4.27 Data management  Data register, data alignment and offset (ADC\_DR, ADC\_JDRy, OFFSETy, OFFSETy\_CH, OVSS, LSHIFT, RSHIFT, SSATE)  Data and alignment  At the end of each regular conversion channel (when EOC event occurs), the result of the converted data is stored into the ADC\_DR data register which is 32 bits wide.  At the end of each injected conversion channel (when JEOC event occurs), the result of the converted data is stored into the corresponding ADC\_JDRy data register which is 32 bits wide.  The OVSS[3:0] and LSHIFT[3:0] bitfields in the ADC\_CFGR2 register selects the alignment of the data stored after conversion. Data can be right- or left-aligned as shown in Figure 166, Figure 167, Figure 168 and Figure 169.  Note: The data can be re-aligned in normal and in oversampling mode.  Offset  An offset y (y=1, 2, 3, 4) can be applied to a channel by programming a value different from 0 in OFFSETy[25:0] bitfield into ADC\_OFRy register. The channel to which the offset will be applied is programmed into the bits OFFSETy\_CH[4:0] of ADC\_OFRy register. In this case, the converted value is decreased by the user-defined offset written in the bits OFFSETy[25:0]. The result may be a negative value so the read data is signed and the SEXT bit represents the extended sign value.  The offset value should be lower than the max conversion value (ex. 16bit mode, offset  value max is 0xFFFF).  The offset correction is also supported in oversampling mode. For the oversampling mode, offset is subtracted before OVSS right shift applied.  Table 211 describes how the comparison is performed for all the possible resolutions for analog watchdog 1, 2, 3.  16-bit and 8-bit signed format management: RSHIFTx,SSATE  The offset correction sign-extends the data format, resulting in an unsigned 16-bit conversion being extended to 17-bit signed format, for instance.  Three options are offered for formatting 8-bit and 16-bit conversion results.  For each offset correction channel 1 to 4, a RSHIFT1..4 bit in the ADC\_CFGR2 register  allows to have the result right-shifted 1-bit and have it fitting a standard 8 or 16-bit format.  Another option is to have the result saturated to the 16-bit and 8-bit signed formats, for the  following cases only: RES[2:0] = 000 (16-bit format) and RES[2:0] = 111 (8-bit format).  This mode is enabled with the SSATE bit in the ADC\_OFRy register.  The table below summarizes the 3 available use case for 16-bit format.  ADC overrun (OVR, OVRMOD)  The overrun flag (OVR) notifies of a buffer overrun event when the regular converted data has not been read (by the CPU or the DMA) before ADC\_DR FIFO (eight stages) is  overflowed.  The OVR flag is set when a new conversion completes while ADC\_CR register FIFO was  full. An interrupt is generated if OVRIE bit is set to 1.  When an overrun condition occurs, the ADC is still operating and can continue to convert unless the software decides to stop and reset the sequence by setting ADSTP to 1.  OVR flag is cleared by software by writing 1 to it.  Data can be configured to be preserved or overwritten when an overrun event occurs by programming the OVRMOD control bit of the ADC\_CFGR register:  • OVRMOD = 0  The overrun event preserves the data register from being overwritten: the old data is  maintained up to ADC\_DR FIFO depth (8 data) and the new conversion is discarded  and lost. If OVR remains at 1, any further conversion is be performed but the resulting  data is also dicarded.  • OVRMOD = 1  The data register is overwritten with the last conversion result and the previous unread  data is lost. In this mode, ADC\_DR FIFO is disabled. If OVR remains at 1, any further  conversion is performed normally and the ADC\_DR register always contains the latest  converted data.  Note: There is no overrun detection on the injected channels since there is a dedicated data  register for each of the four injected channels.  Managing a sequence of conversion without using the DMA  If the conversions are slow enough, the conversion sequence can be handled by the  software. In this case the software must use the EOC flag and its associated interrupt to handle each data. Each time a conversion is complete, EOC is set and the ADC\_DR  register can be read. OVRMOD should be configured to 0 to manage overrun events or FIFO overflows as errors.  Managing conversions without using the DMA and without overrun  It may be useful to let the ADC convert one or more channels without reading the data each time (if there is an analog watchdog for instance). In this case, the OVRMOD bit must be configured to 1 and OVR flag should be ignored by the software. An overrun event will not prevent the ADC from continuing to convert and the ADC\_DR register will always contain the latest conversion.  Managing conversions using the DMA  Since converted channel values are stored into a unique data register, it is useful to use DMA for conversion of more than one channel. This avoids the loss of the data already stored in the ADC\_DR register.  When the DMA mode is enabled (DMNGT bit = 01 or 11 in the ADC\_CFGR register in single ADC mode or MDMA different from 0b00 in dual ADC mode), a DMA request is generated after each conversion of a channel. This allows the transfer of the converted data from the ADC\_DR register to the destination location selected by the software.  Despite this, if an overrun occurs (OVR=1) because the DMA could not serve the DMA transfer request in time, the ADC stops generating DMA requests and the data corresponding to the new conversion is not transferred by the DMA. Which means that all the data transferred to the RAM can be considered as valid.  Depending on the configuration of OVRMOD bit, the data is either preserved or overwritten (refer to Section : ADC overrun (OVR, OVRMOD)).  The DMA transfer requests are blocked until the software clears the OVR bit.  Two different DMA modes are proposed depending on the application use and are  configured with bit DMNGT of the ADC\_CFGR register in single ADC mode, or with bit  DAMDF of the ADCx\_CCR register in dual ADC mode:  • DMA one shot mode (DMNGT bit = 01).  This mode is suitable when the DMA is programmed to transfer a fixed number of data.  • DMA circular mode (DMNGT bit = 11)  This mode is suitable when programming the DMA in circular mode.  DMA one shot mode (DMNGT=01)  In this mode, the ADC generates a DMA transfer request each time a new conversion data is available and stops generating DMA requests once the DMA has reached the last DMA transfer (when DMA\_EOT interrupt occurs - refer to DMA paragraph) even if a conversion has been started again.  When the DMA transfer is complete (all the transfers configured in the DMA controller have been done):  • The content of the ADC data register is frozen.  • Any ongoing conversion is aborted with partial result discarded.  • No new DMA request is issued to the DMA controller. This avoids generating an  overrun error if there are still conversions which are started.  • Scan sequence is stopped and reset.  • The DMA is stopped.  DMA circular mode (DMNGT=11)  In this mode, the ADC generates a DMA transfer request each time a new conversion data is available in the data register, even if the DMA has reached the last DMA transfer. This allows configuring the DMA in circular mode to handle a continuous analog input data stream.  DMA with FIFO  The output data register has eight-stage FIFO. Two different DMA requests are generated parallel. When a data is available, “SREQ single request” generated, when 4 data are available, “BREQ burst request” generated. DMA2 can be programmed either single transfer mode or incremental burst mode(4 beats), according to this mode, correct request line is selected by the DMA2. Please refer to the DMA2 chapter for further information. | 25.4.27 Управление данными  Регистр данных, выравнивание и смещение данных (ADC\_DR, ADC\_JDRy, OFFSETy, OFFSETy\_CH, OVSS, LSHIFT, RSHIFT, SSATE)  Данные и выравнивание  В конце каждого регулярного канала преобразования (когда происходит событие EOC), результат преобразованных данных сохраняется в регистре данных ADC\_DR шириной 32 бита.  В конце каждого инжектированного канала преобразования (при возникновении события JEOC), результат преобразованных данных сохраняется в соответствующем регистре данных ADC\_JDRy шириной 32 бита. шириной 32 бита.  Битовые поля OVSS[3:0] и LSHIFT[3:0] в регистре ADC\_CFGR2 выбирают выравнивание данных, сохраняемых после преобразования. Данные могут быть выровнены вправо или влево, как показано на Рисунок 166, Рисунок 167, Рисунок 168 и Рисунок 169.  Примечание: Данные могут быть выровнены в нормальном режиме и в режиме передискретизации.  Смещение  Смещение y (y=1, 2, 3, 4) может быть применено к каналу путем программирования значения, отличного от 0 в битовом поле OFFSETy[25:0] в регистре ADC\_OFRy. Канал, к которому будет применено смещение программируется в битах OFFSETy\_CH[4:0] регистра ADC\_OFRy. В этом случае, преобразованное значение уменьшается на заданное пользователем смещение, записанное в битах OFFSETy[25:0]. Результатом может быть отрицательное значение, поэтому считанные данные подписываются, а бит SEXT представляет собой расширенное значение знака.  Значение смещения должно быть меньше, чем максимальное значение преобразования (например, 16-битный режим, смещение  максимальное значение - 0xFFFF).  Коррекция смещения также поддерживается в режиме передискретизации. Для режима передискретизации смещение вычитается перед применением правого сдвига OVSS.  В таблице 211 описано, как выполняется сравнение для всех возможных разрешений для аналогового сторожевого таймера 1, 2, 3.  Управление 16- и 8-битными знаковыми форматами: RSHIFTx,SSATE  Коррекция смещения расширяет формат данных по знаку, в результате чего беззнаковое 16-битное преобразование, например, в 17-битный знаковый формат.  Для форматирования результатов 8- и 16-битного преобразования предлагаются три варианта.  Для каждого канала коррекции смещения от 1 до 4, бит RSHIFT1..4 в регистре ADC\_CFGR2  позволяет сдвинуть результат вправо на 1 бит и привести его к стандартному 8- или 16-битному формату.  Другой возможностью является насыщение результата 16- и 8-битными знаковыми форматами, для  только в следующих случаях: RES[2:0] = 000 (16-битный формат) и RES[2:0] = 111 (8-битный формат).  Этот режим включается битом SSATE в регистре ADC\_OFRy.  В таблице ниже приведены 3 доступных варианта использования 16-битного формата.  Выход за пределы АЦП (OVR, OVRMOD)  Флаг превышения (OVR) уведомляет о событии превышения буфера, когда обычные преобразованные данные не были прочитаны (процессором или DMA) до того, как FIFO ADC\_DR (восемь этапов) будет переполнился.  Флаг OVR устанавливается, когда новое преобразование завершается, а FIFO регистра ADC\_CR был переполнен.  переполнен. Прерывание генерируется, если бит OVRIE установлен в 1.  Когда возникает условие превышения, АЦП продолжает работать и может продолжать преобразование. пока программное обеспечение не решит остановить и сбросить последовательность путем установки ADSTP в 1.  Флаг OVR сбрасывается программно путем записи в него 1.  Данные могут быть настроены на сохранение или перезапись при возникновении события переполнения путем программирования управляющего бита OVRMOD регистра ADC\_CFGR:  - OVRMOD = 0  Событие выхода за границы сохраняет регистр данных от перезаписи: старые данные сохраняются до глубины FIFO ADC\_DR (8 данных), а новое преобразование отбрасывается и теряется. Если OVR остается в 1, любое дальнейшее преобразование будет выполнено, но результирующие данные также отбрасываются.  - OVRMOD = 1  Регистр данных перезаписывается последним результатом преобразования, а предыдущие непрочитанные данные теряются. В этом режиме FIFO ADC\_DR отключен. Если OVR остается равным 1, любое дальнейшее преобразование выполняется нормально, и регистр ADC\_DR всегда содержит последние преобразованные данные.  Примечание: В инжектируемых каналах нет обнаружения превышения, так как для каждого из четырех каналов имеется отдельный регистр данных.  для каждого из четырех инжектируемых каналов.  Управление последовательностью преобразований без использования DMA  Если преобразования выполняются достаточно медленно, последовательность преобразований может быть обработана с помощью  программным обеспечением. В этом случае программное обеспечение должно использовать флаг EOC и связанное с ним прерывание для обработки каждых данных. Каждый раз, когда преобразование завершается, EOC устанавливается, и регистр ADC\_DR  может быть прочитан. OVRMOD должен быть сконфигурирован на 0, чтобы управлять событиями превышения или переполнения FIFO как ошибки.  Управление преобразованиями без использования DMA и без разгона  Может оказаться полезным позволить АЦП преобразовывать один или несколько каналов без чтения данных каждый (например, при наличии аналогового сторожевого таймера). В этом случае бит OVRMOD должен быть сконфигурирован в 1, а флаг OVR должен игнорироваться программным обеспечением. Событие превышения не помешает АЦП продолжить преобразование, и регистр ADC\_DR всегда будет содержать последнее преобразование.  Управление преобразованиями с помощью DMA  Поскольку преобразованные значения каналов хранятся в уникальном регистре данных, полезно использовать DMA для преобразования более чем одного канала. Это позволяет избежать потери данных, уже хранящихся в регистре ADC\_DR.  Когда режим DMA включен (бит DMNGT = 01 или 11 в регистре ADC\_CFGR в режиме одиночного АЦП или MDMA, отличном от режима АЦП. или MDMA отличен от 0b00 в режиме двойного АЦП), запрос DMA генерируется после каждого преобразования канала. Это позволяет передавать преобразованные данные из регистра ADC\_DR в место назначения, выбранное программным обеспечением.  Несмотря на это, если произойдет выход за границы (OVR=1), потому что DMA не смог вовремя обслужить запрос DMA запрос на передачу, АЦП прекращает генерировать запросы DMA и данные соответствующее новому преобразованию, не передается DMA. Это означает, что все данные, переданные в ОЗУ, можно считать действительными.  В зависимости от конфигурации бита OVRMOD, данные либо сохраняются, либо перезаписываются (см. раздел : Выход за пределы АЦП (OVR, OVRMOD)).  Запросы на передачу данных DMA блокируются до тех пор, пока программное обеспечение не очистит бит OVR.  Два различных режима DMA предлагаются в зависимости от использования приложения и конфигурируются с помощью бита DMNGT.  конфигурируются с помощью бита DMNGT регистра ADC\_CFGR в режиме одиночного АЦП или с помощью бита  DAMDF регистра ADCx\_CCR в режиме двойного АЦП:  - Режим DMA one shot (бит DMNGT = 01).  Этот режим подходит, когда DMA запрограммирован на передачу фиксированного количества данных.  - Циркулярный режим DMA (бит DMNGT = 11).  Этот режим подходит для программирования DMA в круговом режиме.  Режим DMA one shot (DMNGT=01)  В этом режиме АЦП генерирует запрос на передачу DMA каждый раз, когда доступны новые данные преобразования и прекращает генерировать запросы DMA, как только DMA достигнет последнего DMA передачи (когда возникает прерывание DMA\_EOT - см. параграф DMA), даже если преобразование было начато снова.  Когда передача DMA завершена (все передачи, сконфигурированные в контроллере DMA, были были выполнены):  - Содержимое регистра данных АЦП замораживается.  - Любое текущее преобразование прерывается с отбрасыванием частичного результата.  - Новый запрос DMA контроллеру DMA не выдается. Это позволяет избежать генерации ошибки превышения, если все еще есть начатые преобразования.  - Последовательность сканирования останавливается и сбрасывается.  - DMA останавливается.  Циркулярный режим DMA (DMNGT=11)  В этом режиме АЦП генерирует запрос на передачу DMA каждый раз, когда новые данные преобразования доступны в регистре данных, даже если DMA достиг последней передачи DMA. Этот позволяет конфигурировать DMA в круговом режиме для обработки непрерывного потока аналоговых входных данных. поток.  DMA с FIFO  Выходной регистр данных имеет восьмиступенчатый FIFO. Два различных запроса DMA генерируются параллельный. Когда доступны данные, генерируется "SREQ single request", когда доступны 4 данных, генерируется "BREQ burst request". доступны, генерируется "BREQ burst request". DMA2 может быть запрограммирован либо в однократном режим передачи или инкрементный пакетный режим (4 такта), в соответствии с этим режимом, правильная линия запроса выбирается DMA2. Пожалуйста, обратитесь к главе DMA2 для получения дополнительной информации. |
| 25.4.28 Managing conversions using the DFSDM  The ADC conversion results can be transferred directly to the Digital Filter for Sigma Delta  Modulators (DFSDM).  In this case, the DMNGT[1:0] bits must be set to 10.  The ADC transfers 16 least significant bits of the regular data register data to the DFSDM,  which in turns will reset the EOC flag once the transfer is effective.  The data format must be 16-bit signed:  ADC\_DR[31:16] = don’t care  ADC\_DR[15] = sign  ADC\_DR[14:0] = data  Any value above 16-bit signed format will be runcated. | 25.4.28 Управление преобразованиями с помощью DFSDM  Результаты преобразования АЦП могут быть переданы непосредственно в цифровой фильтр для сигма-дельта  модуляторов (DFSDM).  В этом случае биты DMNGT[1:0] должны быть установлены в 10.  АЦП передает 16 младших битов данных обычного регистра данных в DFSDM, который, в свою очередь, сбросит флаг EOC, как только передача станет эффективной.  Формат данных должен быть 16-битным знаковым:  ADC\_DR[31:16] = don't care  ADC\_DR[15] = знак  ADC\_DR[14:0] = данные  Любое значение, превышающее 16-битный знаковый формат, будет усечено. |
| 25.4.29 Dynamic low-power features  Auto-delayed conversion mode (AUTDLY)  The ADC implements an auto-delayed conversion mode controlled by the AUTDLY  configuration bit. Auto-delayed conversions are useful to simplify the software as well as to optimize performance of an application clocked at low frequency where there would be risk of encountering an ADC overrun.  When AUTDLY=1, a new conversion can start only if all the previous data of the same group has been treated:  • For a regular conversion: once the ADC\_DR register has been read or if the EOC bit  has been cleared (see Figure 172).  • For an injected conversion: when the JEOS bit has been cleared (see Figure 173).  This is a way to automatically adapt the speed of the ADC to the speed of the system which will read the data.  The delay is inserted after each regular conversion (whatever DISCEN=0 or 1) and after  each sequence of injected conversions (whatever JDISCEN=0 or 1).  Note: There is no delay inserted between each conversions of the injected sequence, except after the last one.  During a conversion, a hardware trigger event (for the same group of conversions) occurring during this delay is ignored.  Note: This is not true for software triggers where it remains possible during this delay to set the bits ADSTART or JADSTART to re-start a conversion: it is up to the software to read the data before launching a new conversion.  No delay is inserted between conversions of different groups (a regular conversion followed by an injected conversion or conversely):  • If an injected trigger occurs during the automatic delay of a regular conversion, the  injected conversion starts immediately (see Figure 173).  • Once the injected sequence is complete, the ADC waits for the delay (if not ended) of the previous regular conversion before launching a new regular conversion (see Figure 175).  The behavior is slightly different in auto-injected mode (JAUTO=1) where a new regular conversion can start only when the automatic delay of the previous injected sequence of conversion has ended (when JEOS has been cleared). This is to ensure that the software can read all the data of a given sequence before starting a new sequence (see Figure 176).  To stop a conversion in continuous auto-injection mode combined with autodelay mode  (JAUTO=1, CONT=1 and AUTDLY=1), follow the following procedure:  1. Wait until JEOS=1 (no more conversions are restarted)  2. Clear JEOS,  3. Set ADSTP=1  4. Read the regular data.  If this procedure is not respected, a new regular sequence can re-start if JEOS is cleared after ADSTP has been set.  In AUTDLY mode, a hardware regular trigger event is ignored if it occurs during an already ongoing regular sequence or during the delay that follows the last regular conversion of the sequence. It is however considered pending if it occurs after this delay, even if it occurs during an injected sequence of the delay that follows it. The conversion then starts at the end of the delay of the injected sequence.  In AUTDLY mode, a hardware injected trigger event is ignored if it occurs during an already ongoing injected sequence or during the delay that follows the last injected conversion of the sequence. | 25.4.29 Динамические функции низкого энергопотребления  Режим преобразования с автоматической задержкой (AUTDLY)  АЦП реализует режим преобразования с автоматической задержкой, управляемый битом конфигурации AUTDLY  битом конфигурации. Преобразование с автоматической задержкой полезно для упрощения программного обеспечения, а также для оптимизировать работу приложения, работающего на низкой частоте, где существует риск столкнуться с превышением АЦП.  Когда AUTDLY=1, новое преобразование может начаться только в том случае, если все предыдущие данные той же группы были обработаны:  - Для обычного преобразования: после считывания регистра ADC\_DR или если бит EOC был очищен (см. Рисунок 172).  - Для инжектированного преобразования: когда бит JEOS очищен (см. Рисунок 173).  Это способ автоматической адаптации скорости АЦП к скорости системы, которая будет считывать данные.  Задержка вставляется после каждого обычного преобразования (независимо от того, DISCEN=0 или 1) и после каждой последовательности инжектированных преобразований (независимо от того, JDISCEN=0 или 1).  Примечание: Задержка не вставляется между каждым преобразованием инжектированной последовательности, за исключением последнего. последнего.  Во время преобразования событие аппаратного триггера (для той же группы преобразований), происходящее во время этой задержки игнорируется.  Примечание: Это не относится к программным триггерам, где во время этой задержки сохраняется возможность установить биты ADSTART или JADART. биты ADSTART или JADSTART для повторного запуска преобразования: считывание данных перед запуском нового преобразования зависит от программного обеспечения. данных перед запуском нового преобразования.  Между преобразованиями разных групп задержка не вводится (обычное преобразование с последующим инжектированным преобразованием или наоборот):  - Если инжектированный триггер происходит во время автоматической задержки обычного преобразования, то  инжектированное преобразование запускается немедленно (см. Рисунок 173).  - После завершения инжектированной последовательности АЦП ожидает задержки (если она не закончилась) предыдущего обычного преобразования, прежде чем запустить его. предыдущего регулярного преобразования перед запуском нового регулярного преобразования (см. Рисунок 175).  Поведение немного отличается в режиме автоинжекции (JAUTO=1), где новое регулярное преобразование может начаться только тогда, когда автоматическая задержка предыдущей инжектированной последовательности преобразования закончилась (когда JEOS был сброшен). Это необходимо для того, чтобы программное обеспечение успеет прочитать все данные данной последовательности перед началом новой последовательности (см. Рисунок 176).  Чтобы остановить преобразование в режиме непрерывного автоматического впрыска в сочетании с режимом автодолива  (JAUTO=1, CONT=1 и AUTDLY=1), выполните следующую процедуру:  1. Подождите, пока JEOS=1 (больше преобразования не перезапускаются).  2. Очистите JEOS,  3. Установить ADSTP=1  4. Считайте регулярные данные.  Если эта процедура не соблюдается, новая регулярная последовательность может быть запущена заново, если очистить JEOS после установки ADSTP.  В режиме AUTDLY событие аппаратного регулярного триггера игнорируется, если оно происходит во время уже текущей регулярной последовательности или во время задержки, следующей за последним регулярным преобразованием в последовательности последовательности. Однако оно считается ожидаемым, если возникает после этой задержки, даже если оно возникает во время инжектированной последовательности в следующей за ней задержке. Тогда преобразование начинается в окончании задержки инжектированной последовательности.  В режиме AUTDLY инжектированное аппаратное триггерное событие игнорируется, если оно происходит во время уже текущей инжектированной последовательности или во время задержки, которая следует за последним инжектированным преобразованием последовательности. |
| 25.4.30 Analog window watchdog (AWD1EN, JAWD1EN, AWD1SGL,  AWD1CH, AWD2CH, AWD3CH, AWD\_HTRy, AWD\_LTRy, AWDy)  The three AWD analog watchdogs monitor whether some channels remain within a configured voltage range (window).  AWDx flag and interrupt  An interrupt can be enabled for each of the 3 analog watchdogs by setting AWDyIE in the ADC\_IER register (x=1,2,3).  AWDy (y=1,2,3) flag is cleared by software by writing 1 to it.  The ADC conversion result is compared to the lower and higher thresholds before  alignment.  Description of analog watchdog 1  The AWD analog watchdog 1 is enabled by setting the AWD1EN bit in the ADC\_CFGR  register. This watchdog monitors whether either one selected channel or all enabled  channels(1) remain within a configured voltage range (window).  Table 214 shows how the ADC\_CFGRy registers should be configured to enable the analog watchdog on one or more channels.  These thresholds are programmed in bits HTR1[25:0] of the ADC\_HTR1 register and  LTR1[25:0] of the ADC\_LTR1 register for the analog watchdog 1.  The threshold can be up to 26-bits (16-bit resolution with oversampling, OSVR[9:0]=1024).  When converting data with a resolution of less than 16 bits (according to bits RES[2:0]), the LSBs of the programmed thresholds must be kept cleared, the internal comparison being performed on the full 16-bit converted data (left aligned to the half-word boundary).  Table 215 describes how the comparison is performed for all the possible resolutions for analog watchdog 1,2,3. | 25.4.30 Сторожевой таймер аналогового окна (AWD1EN, JAWD1EN, AWD1SGL,  AWD1CH, AWD2CH, AWD3CH, AWD\_HTRy, AWD\_LTRy, AWDy)  Три аналоговых сторожевых устройства AWD контролируют, остаются ли некоторые каналы в пределах сконфигурированного диапазона напряжения (окна).  Флаг AWDx и прерывание  Прерывание может быть включено для каждого из 3 аналоговых сторожевых устройств путем установки AWDyIE в регистре ADC\_IER (x=1,2,3).  Флаг AWDy (y=1,2,3) снимается программно путем записи в него 1.  Результат преобразования АЦП сравнивается с нижним и верхним порогами перед выравниванием.  выравнивания.  Описание аналогового сторожевого таймера 1  Аналоговый сторожевой таймер AWD 1 включается установкой бита AWD1EN в ADC\_CFGR  регистра. Этот сторожевой таймер отслеживает, находится ли один выбранный канал или все включенные  (1) остаются в пределах сконфигурированного диапазона напряжения (окна).  В таблице 214 показано, как должны быть настроены регистры ADC\_CFGRy, чтобы включить аналоговый сторожевой таймер на одном или нескольких каналах. сторожевой таймер на одном или нескольких каналах.  Эти пороговые значения программируются в битах HTR1[25:0] регистра ADC\_HTR1 и  LTR1[25:0] регистра ADC\_LTR1 для аналогового сторожевого таймера 1.  Порог может достигать 26 бит (16-битное разрешение с передискретизацией, OSVR[9:0]=1024).  При преобразовании данных с разрешением менее 16 бит (в соответствии с битами RES[2:0]) LSB запрограммированных порогов должны быть очищены, при этом внутреннее сравнение будет выполняется на полных 16-битных преобразованных данных (выровненных влево по границе полуслова).  Таблица 215 описывает, как выполняется сравнение для всех возможных разрешений для аналогового сторожевого таймера 1,2,3. |
| 25.4.31 Oversampler  The oversampling unit performs data preprocessing to offload the CPU. It is able to handle multiple conversions and average them into a single data with increased data width, up to 26-bit (16-bit values and OSVR[9:0] = 1024).  It provides a result with the following form, where N and M can be adjusted:  It allows to perform by hardware the following functions: averaging, data rate reduction,  SNR improvement, basic filtering.  The oversampling ratio N is defined using the OSVR[9:0] bits in the ADC\_CFGR2 register, and can range from 2x to 1024x. The division coefficient M consists of a right bit shift up to 10 bits, and is defined using the OVSS[3:0] bits in the ADC\_CFGR2 register.  The summation unit can yield a result up to 26 bits (1024 x 16-bit results), which can be left or right shifted. When right shifting is selected, it is rounded to the nearest value using the least significant bits left apart by the shifting, before being transferred into the ADC\_DR data register.  The Table 182 gives a numerical example of the processing, from a raw 26-bit accumulated data to the final 16-bit result.  There are no changes for conversion timings in oversampled mode: the sample time is maintained equal during the whole oversampling sequence. A new data is provided every N conversions, with an equivalent delay equal to N x TCONV = N x (tSMPL + tSAR). The flags are set as follow:  • the end of the sampling phase (EOSMP) is set after each sampling phase  • the end of conversion (EOC) occurs once every N conversions, when the oversampled result is available  • the end of sequence (EOS) occurs once the sequence of oversampled data is  completed (i.e. after N x sequence length conversions total) | 25.4.31 Передискретизация  Блок передискретизации выполняет предварительную обработку данных, чтобы разгрузить центральный процессор. Он способен обрабатывать несколько преобразований и усреднять их в единые данные с увеличенной шириной данных, вплоть до 26-битных (16-битные значения и OSVR[9:0] = 1024).  Он выдает результат следующей формы, где N и M могут быть отрегулированы:  Он позволяет аппаратно выполнять следующие функции: усреднение, снижение скорости передачи данных,  улучшение SNR, базовая фильтрация.  Коэффициент передискретизации N определяется с помощью битов OSVR[9:0] в регистре ADC\_CFGR2, и может варьироваться от 2x до 1024x. Коэффициент деления M состоит из сдвига вправо до 10 бит, и определяется с помощью битов OVSS[3:0] в регистре ADC\_CFGR2.  Блок суммирования может выдавать результат до 26 бит (1024 x 16-битных результата), который может быть сдвинут влево или сдвинут вправо. При выборе сдвига вправо результат округляется до ближайшего значения с использованием наименьшие значащие биты, оставшиеся после сдвига, перед передачей в регистр данных ADC\_DR регистр.  В таблице 182 приведен числовой пример обработки от исходных 26-битных накопленных данных до окончательного 16-битного результата.  Время преобразования в режиме передискретизации не меняется: время дискретизации поддерживается одинаковым в течение всей последовательности передискретизации. Новые данные предоставляются каждые N преобразований, с эквивалентной задержкой, равной N x TCONV = N x (tSMPL + tSAR). Флаги устанавливаются следующим образом:  - конец фазы выборки (EOSMP) устанавливается после каждой фазы выборки  - конец преобразования (EOC) происходит раз в N преобразований, когда передискретизированный результат  - конец последовательности (EOS) происходит после того, как последовательность передискретизированных данных будет  (т.е. после N x длина последовательности преобразований) |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
| Introduction This application note provides an overview of the on-chip hardware Analog-to-Digital Converter (ADC) oversampling engine integrated in microcontrollers belonging to the STM32 L0 and L4 series. The main benefit the user can get from the hardware oversampling is increased SNR (signal-to-noise ratio) with less CPU interaction, resulting in overall lower power consumption compared with the software-based implementation. | Введение  В данном руководстве  [..\dm00150423-adc-hardware-oversampling-for-microcontrollers-of-the-stm32-l0-and-l4-series-stmicroelectronics.pdf](../dm00150423-adc-hardware-oversampling-for-microcontrollers-of-the-stm32-l0-and-l4-series-stmicroelectronics.pdf)  по применению представлен обзор аппаратного аналого-цифрового преобразователя на кристалле.  преобразователя (АЦП) с передискретизацией, встроенного в микроконтроллеры серий  STM32 серий L0 и L4.  Для Н7 найден следующий документ [..\dm00628458-getting-started-with-the-stm32h7-series-mcu-16bit-adc-stmicroelectronics.pdf](../dm00628458-getting-started-with-the-stm32h7-series-mcu-16bit-adc-stmicroelectronics.pdf)  Основное преимущество, которое пользователь может получить от аппаратной передискретизации, заключается в увеличении SNR (отношение сигнал/шум) при меньшем взаимодействии с процессором, что приводит к общему снижению энергопотребления  по сравнению с программной реализацией. |
|  |  |