

10 Direct memory access controller (DMA)

RM0091 pg 188/1004

Контроллер прямого доступа к памяти (DMA)

10.1 Введение

Прямой доступ к памяти (DMA) используется для обеспечения высокоскоростной передачи данных между периферийными устройствами и памятью, а также памятью в память. Данные могут быть быстро перемещены DMA без каких-либо действий ЦП. Это освобождает ресурсы ЦП для других операций.

Контроллер DMA имеет до 12 каналов, каждый из которых предназначен для управления запросами на доступ к памяти от одного или нескольких периферийных устройств. В нем есть арбитр для обработки приоритета между запросами DMA.

ИЗМЕНЕНИЯ: 09-01-2014

DMA

- Измените 5 каналов на 7 каналов
- Обновлено Рисунок 21: Блок-схема DMA
- Обновлено Рисунок 23: Отображение запроса DMA
- Добавлена таблица 29: Сводка запросов DMA для каждого канала на STM32F03x, STM32F04x и STM32F05x и Таблица 30: Сводка запросов DMA для каждого канала на устройствах STM32F07x
- Добавлены биты 27-20 в разделе 11.5.1: регистр состояния прерывания DMA (DMA_ISR) и раздел 11.5.2: регистр сброса флага прерывания DMA (DMA_IFCR)

Functional overview STM32F051x

Pg 16/1053

8 Прямой контроллер доступа к памяти (DMA)

5-канальные DMA общего назначения управляют передачами между памятью, периферией и памятью и периферией.

DMA поддерживает циклическое управление буфером, устраняя необходимость вмешательства пользователя, когда контроллер достигает конца буфера.

Каждый канал подключается к выделенным аппаратным запросам DMA, поддерживая программный триггер на каждом канале. Конфигурация выполняется с помощью программного обеспечения, а размеры передачи между источником и местом назначения являются независимыми.

DMA может использоваться с основными периферийными устройствами: SPI, I2S, I2C, USART, все таймеры TIMx (кроме TIM14), ЦАП и АЦП.

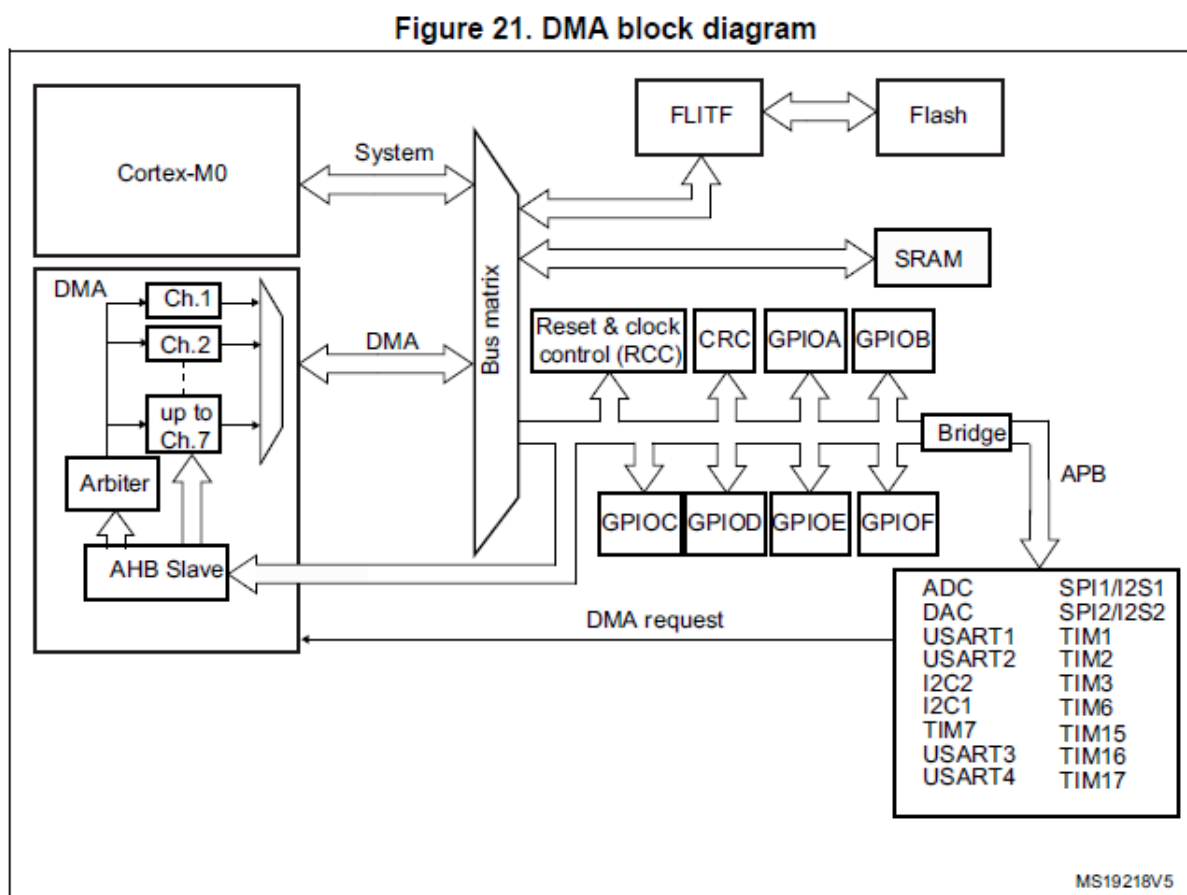
10.2 Основные функции DMA

- До 7 независимых настраиваемых каналов (запросов) на DMA
- STM32F09x предоставляет 5 дополнительных независимо настраиваемых каналов (запросов) на DMA2
- Каждый канал подключен к выделенным аппаратным запросам DMA, программный триггер также поддерживается на каждом канале. Эта конфигурация выполняется с помощью программного обеспечения.
- Приоритеты между запросами каналов DMA программируются программно (4 уровни, состоящие из очень высокого, высокого, среднего, низкого) или аппаратного обеспечения в случае равенства (запрос 1 имеет приоритет над запросом 2 и т. д.),
- Независимый размер передачи источника и назначения (байт, половина слова, слово), эмулирование упаковки и распаковки. Исходные / целевые адреса должны быть выровнены по размеру данных.
- Поддержка управления циклическим буфером

- 3 флага событий (DMA Half Transfer, DMA Transfer complete и DMA Transfer Error) логически ORed вместе в одном запросе прерывания для каждого канала
- Передача памяти на память
- Переключение между памятью и памятью на периферию и периферийные периферийные передачи
- Доступ к периферийным устройствам Flash, SRAM, APB и AHB в качестве источника и назначения
- Программируемое количество передаваемых данных: до 65535

10.3 Функциональное описание DMA

Структурная схема показана на следующем рисунке.



Контроллер DMA выполняет прямую передачу данных путем совместного использования системной шины с ядром Cortex®-M0. Запрос DMA может остановить доступ ЦП к системной шине для некоторых циклов шины, когда ЦП и DMA нацелены на один и тот же пункт назначения (память или периферийное устройство). Матрица шины реализует циклическое планирование, обеспечивая, по меньшей мере, половину пропускной способности системной шины (как для памяти, так и для периферии) для ЦП.

10.3.1 Операции DMA (DMA transactions)

После события периферийное устройство отправляет сигнал запроса в контроллер DMA. Контроллер DMA выполняет запрос в зависимости от приоритетов канала. Как только контроллер DMA обращается к периферийному устройству, он посылается периферийному устройству контроллером DMA. Периферийное устройство отправляет запрос, как только получает подтверждение от контроллера DMA. Как только запрос отменяется периферийным устройством, контроллер DMA освобождает подтверждение. Если запросов больше, периферийное устройство может инициировать следующую транзакцию.

Таким образом, каждая передача DMA состоит из трех операций:

- Загрузка данных из регистра периферийных данных или местоположение в памяти, адресованное внутренним регистром адресов периферийных устройств / памяти. Начальный адрес, используемый для первой передачи, является адресом базовой периферии / памяти, запрограммированным в регистре DMA_CPARx или DMA_CMARx.

- Хранение данных, загружаемых в регистр периферийных данных, или местоположение в памяти, адресованное внутренним регистром адресов периферийных устройств / памяти. Начальный адрес, используемый для первой передачи, является адресом базовой периферии / памяти, запрограммированным в регистре DMA_CPARx или DMA_CMARx.

- Пост декрементация регистра DMA_CNDTRx, который содержит количество транзакций, которые еще предстоит выполнить

10.3.2 Арбитр

Арбитр управляет запросами канала на основе их приоритета и запускает периферийных / последовательностей доступа к памяти.

Приоритеты управляются в два этапа:

- Программное обеспечение: каждый приоритет канала может быть настроен в регистре DMA_CCRx. Существует четыре уровня:

- Очень высокий приоритет
- Высокий приоритет
- Средний приоритет
- Низкий приоритет

- Аппаратное обеспечение: если 2 запроса имеют одинаковый уровень приоритета программного обеспечения, канал с самым низким номером получит приоритет по сравнению с каналом с наибольшим номером. Например, канал 2 получает приоритет над каналом 4.

10.3.3 Каналы DMA

Каждый канал может обрабатывать передачу DMA между периферийным регистром, расположенным по фиксированному адресу и адресом памяти. Объем данных, подлежащих передаче (до 65535), программируется. Регистр, который содержит количество передаваемых элементов данных, уменьшается после каждой транзакции.

Программируемые размеры данных

Размеры передаваемых данных периферийного устройства и памяти полностью программируются через биты PSIZE и MSIZE в регистре DMA_CCRx.

Инкремент указателя

Периферийные и запоминающие указатели могут быть автоматически добавлены после каждой транзакции в зависимости от битов PINC и MINC в регистре DMA_CCRx. Если активирован инкрементный режим, адрес следующей передачи будет адресом предыдущего, увеличенным на 1, 2 или 4 в зависимости от выбранного размера данных. Первым адресом передачи является тот, который запрограммирован в регистрах DMA_CPARx / DMA_CMARx. Во время операций переноса эти регистры сохраняют первоначально запрограммированное значение. Текущие адреса передачи (в текущем внутреннем регистре периферии / памяти) недоступны программным обеспечением.

Если канал настроен в некруглом режиме, ни один запрос DMA не подается после последней передачи (то есть, когда количество передаваемых элементов данных достигнет нуля). Чтобы перезагрузить новое количество элементов данных, которые должны быть переданы в регистр DMA_CNDTRx, канал DMA должен быть отключен.

10.3.4. Программируемая ширина данных, выравнивание данных и континцы

Когда PSIZE и MSIZE не равны, DMA выполняет некоторые выравнивания данных, как описано в таблице 27:

Программируемая ширина данных и поведение endian (когда бит PINC = MINC = 1).

Table 27. Programmable data width & endian behavior (when bits PINC = MINC = 1)

Source port width	Destination port width	Number of data items to transfer (NDT)	Source content: address / data	Transfer operations	Destination content: address / data
8	8	4	@0x0 / B0 @0x1 / B1 @0x2 / B2 @0x3 / B3	1: READ B0[7:0] @0x0 then WRITE B0[7:0] @0x0 2: READ B1[7:0] @0x1 then WRITE B1[7:0] @0x1 3: READ B2[7:0] @0x2 then WRITE B2[7:0] @0x2 4: READ B3[7:0] @0x3 then WRITE B3[7:0] @0x3	@0x0 / B0 @0x1 / B1 @0x2 / B2 @0x3 / B3
8	16	4	@0x0 / B0 @0x1 / B1 @0x2 / B2 @0x3 / B3	1: READ B0[7:0] @0x0 then WRITE 00B0[15:0] @0x0 2: READ B1[7:0] @0x1 then WRITE 00B1[15:0] @0x2 3: READ B2[7:0] @0x2 then WRITE 00B2[15:0] @0x4 4: READ B3[7:0] @0x3 then WRITE 00B3[15:0] @0x6	@0x0 / 00B0 @0x2 / 00B1 @0x4 / 00B2 @0x6 / 00B3
8	32	4	@0x0 / B0 @0x1 / B1 @0x2 / B2 @0x3 / B3	1: READ B0[7:0] @0x0 then WRITE 000000B0[31:0] @0x0 2: READ B1[7:0] @0x1 then WRITE 000000B1[31:0] @0x4 3: READ B2[7:0] @0x2 then WRITE 000000B2[31:0] @0x8 4: READ B3[7:0] @0x3 then WRITE 000000B3[31:0] @0xC	@0x0 / 000000B0 @0x4 / 000000B1 @0x8 / 000000B2 @0xC / 000000B3
16	8	4	@0x0 / B1B0 @0x2 / B3B2 @0x4 / B5B4 @0x6 / B7B6	1: READ B1B0[15:0] @0x0 then WRITE B0[7:0] @0x0 2: READ B3B2[15:0] @0x2 then WRITE B2[7:0] @0x1 3: READ B5B4[15:0] @0x4 then WRITE B4[7:0] @0x2 4: READ B7B6[15:0] @0x6 then WRITE B6[7:0] @0x3	@0x0 / B0 @0x2 / B2 @0x4 / B4 @0x6 / B6
16	16	4	@0x0 / B1B0 @0x2 / B3B2 @0x4 / B5B4 @0x6 / B7B6	1: READ B1B0[15:0] @0x0 then WRITE B1B0[15:0] @0x0 2: READ B3B2[15:0] @0x2 then WRITE B3B2[15:0] @0x2 3: READ B5B4[15:0] @0x4 then WRITE B5B4[15:0] @0x4 4: READ B7B6[15:0] @0x6 then WRITE B7B6[15:0] @0x6	@0x0 / B1B0 @0x2 / B3B2 @0x4 / B5B4 @0x6 / B7B6
16	32	4	@0x0 / B1B0 @0x2 / B3B2 @0x4 / B5B4 @0x6 / B7B6	1: READ B1B0[15:0] @0x0 then WRITE 0000B1B0[31:0] @0x0 2: READ B3B2[15:0] @0x2 then WRITE 0000B3B2[31:0] @0x4 3: READ B5B4[15:0] @0x4 then WRITE 0000B5B4[31:0] @0x8 4: READ B7B6[15:0] @0x6 then WRITE 0000B7B6[31:0] @0xC	@0x0 / 0000B1B0 @0x4 / 0000B3B2 @0x8 / 0000B5B4 @0xC / 0000B7B6
32	8	4	@0x0 / B3B2B1B0 @0x4 / B7B6B5B4 @0x8 / BBBAB9B8 @0xC / BFBEBDBC	1: READ B3B2B1B0[31:0] @0x0 then WRITE B0[7:0] @0x0 2: READ B7B6B5B4[31:0] @0x4 then WRITE B4[7:0] @0x1 3: READ BBBAB9B8[31:0] @0x8 then WRITE B8[7:0] @0x2 4: READ BFBEBDBC[31:0] @0xC then WRITE BC[7:0] @0x3	@0x0 / B0 @0x1 / B4 @0x2 / B8 @0x3 / BC
32	16	4	@0x0 / B3B2B1B0 @0x4 / B7B6B5B4 @0x8 / BBBAB9B8 @0xC / BFBEBDBC	1: READ B3B2B1B0[31:0] @0x0 then WRITE B1B0[15:0] @0x0 2: READ B7B6B5B4[31:0] @0x4 then WRITE B5B4[15:0] @0x2 3: READ BBBAB9B8[31:0] @0x8 then WRITE B9B8[15:0] @0x4 4: READ BFBEBDBC[31:0] @0xC then WRITE BDBC[15:0] @0x6	@0x0 / B1B0 @0x2 / B5B4 @0x4 / B9B8 @0x6 / BDBC
32	32	4	@0x0 / B3B2B1B0 @0x4 / B7B6B5B4 @0x8 / BBBAB9B8 @0xC / BFBEBDBC	1: READ B3B2B1B0[31:0] @0x0 then WRITE B3B2B1B0[31:0] @0x0 2: READ B7B6B5B4[31:0] @0x4 then WRITE B7B6B5B4[31:0] @0x4 3: READ BBBAB9B8[31:0] @0x8 then WRITE BBBAB9B8[31:0] @0x8 4: READ BFBEBDBC[31:0] @0xC then WRITE BFBEBDBC[31:0] @0xC	@0x0 / B3B2B1B0 @0x4 / B7B6B5B4 @0x8 / BBBAB9B8 @0xC / BFBEBDBC

Адресация периферийного устройства АНВ, которое не поддерживает операции записи байтов или полуслов

Когда DMA инициирует операцию записи байта или полуслова АНВ, данные отражаются на неиспользуемых дорожках шины HWDATA [31: 0]. Поэтому, когда используемое вспомогательное периферийное устройство АНВ не поддерживает операции записи байтов или полуслов (когда HSIZE не используется периферийным устройством) и не генерирует никаких ошибок, DMA записывает 32 бита HWDATA, как показано в двух примерах ниже:

- Чтобы записать полуслова «0xABCD», DMA устанавливает шину HWDATA в «0xABCDABCD» с HSIZE = HalfWord
- Чтобы записать байт «0xAB», DMA устанавливает шину HWDATA в «0xABABABAB» с HSIZE = Byte. Предполагая, что мост АНВ / APB представляет собой 32-битное подчиненное периферийное устройство АНВ, которое не учитывает данные HSIZE, оно будет преобразует любую операцию байта или полуслова АНВ в 32-разрядную операцию APB следующим образом:
 - Операция записи байтов АНВ данных «0xB0» в 0x0 (или в 0x1, 0x2 или 0x3) будет преобразована в операцию записи слов APB данных «0xB0B0B0B0» в 0x0

- Операция записи ползучести АНВ данных «0xB1B0» в 0x0 (или до 0x2) будет преобразуется в запись слов APB для записи данных «0xB1B0B1B0» в 0x0

Например, для записи регистров резервного копирования APB (16-разрядные регистры, ориентированные на 32-разрядную границу адреса) программное обеспечение должно настроить размер источника памяти (MSIZE) на «16-разрядный» и размер периферийного адресата (PSIZE) на «32-бит».

10.3.5 Управление ошибками

Ошибка передачи DMA может быть сгенерирована путем чтения или записи в зарезервированное адресное пространство. Когда ошибка передачи DMA возникает во время чтения DMA или доступа к записи, неисправный канал автоматически отключается через аппаратное обеспечение, исключая его бит EN в соответствующем регистре конфигурации канала (DMA_CCRx). Флаг прерывания передачи канала (TEIF) в регистре DMA_IFR установлен, и прерывание генерируется, если установлен бит разрешения прерывания передачи (TEIE) в регистре DMA_CCRx.

10.3.6 Прерывания DMA

Прерывание может быть выполнено при передаче по полу-передаче, передаче или передаче для каждого канала DMA. Для гибкости доступны отдельные бит разрешения прерывания.

Table 28. DMA interrupt requests

Interrupt event	Event flag	Enable control bit
Half-transfer	HTIF	HTIE
Transfer complete	TCIF	TCIE
Transfer error	TEIF	TEIE

Контроллер DMA

Аппаратные запросы от периферийных устройств (TIMx, ADC, DAC, SPI, I2C и USARTx) просто логически ORed перед входом в DMA. Это означает, что на одном канале одновременно должен быть включен только один запрос.

Периферийные запросы DMA могут быть независимо активированы / деактивированы путем программирования бит управления DMA в регистрах соответствующего периферийного устройства.

В таблице 29 и таблице 30 перечислены запросы DMA для каждого канала.

Table 29. Summary of the DMA requests for each channel on STM32F03x, STM32F04x and STM32F05x devices

Peripherals	Channel 1	Channel 2	Channel 3	Channel 4	Channel 5
ADC	ADC ⁽¹⁾	ADC ⁽²⁾	-	-	-
SPI	-	SPI1_RX	SPI1_TX	SPI2_RX	SPI2_TX
USART	-	USART1_TX ⁽¹⁾	USART1_RX ⁽¹⁾	USART1_TX ⁽²⁾ USART2_TX	USART1_RX ⁽²⁾ USART2_RX
I2C	-	I2C1_TX	I2C1_RX	I2C2_TX	I2C2_RX
TIM1	-	TIM1_CH1	TIM1_CH2	TIM1_CH4 TIM1_TRIG TIM1_COM	TIM1_CH3 TIM1_UP
TIM2	TIM2_CH3	TIM2_UP	TIM2_CH2	TIM2_CH4	TIM2_CH1
TIM3	-	TIM3_CH3	TIM3_CH4 TIM3_UP	TIM3_CH1 TIM3_TRIG	-
TIM6 / DAC	-	-	TIM6_UP DAC_Channel1	-	-
TIM15	-	-	-	-	TIM15_CH1 TIM15_UP TIM15_TRIG TIM15_COM
TIM16	-	-	TIM16_CH1 ⁽¹⁾ TIM16_UP ⁽¹⁾	TIM16_CH1 ⁽²⁾ TIM16_UP ⁽²⁾	-
TIM17	TIM17_CH1 ⁽¹⁾ TIM17_UP ⁽¹⁾	TIM17_CH1 ⁽²⁾ TIM17_UP ⁽²⁾	-	-	-

1. Запрос DMA, отображаемый на этом канале DMA только в том случае, если соответствующий бит повторного набора очищается в регистре SYSCFG_CFGR1. Для получения дополнительной информации см. Раздел 9.1.1: регистр конфигурации SYSCFG 1 (SYSCFG_CFGR1) на стр. 165.

2. Запрос DMA отображается на этом канале DMA только в том случае, если соответствующий бит переназначения установлен в регистре SYSCFG_CFGR1.

Для получения дополнительной информации см. Раздел 9.1.1: регистр конфигурации SYSCFG 1 (SYSCFG_CFGR1) на стр. 165.

9.1 Регистр SYSCFG

Pg 165/1004

Этот регистр используется для конкретных конфигураций памяти и переназначения запросов DMA и для управления специальными функциями ввода-вывода.

Два бита используются для настройки типа памяти, доступной по адресу 0x0000 0000. Эти биты используются для выбора физического переназначения с помощью программного обеспечения и, следовательно, обхода выбора аппаратного BOOT.

После сброса эти биты принимают значение, выбранное по фактической конфигурации режима загрузки.

Смещение адреса: 0x00

Значение сброса: 0x0000 000X (X - это режим памяти, выбранный по фактической конфигурации режима загрузки)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	TIM3_DMA_RMP	TIM2_DMA_RMP	TIM1_DMA_RMP	I2C1_DMA_RMP	USART3_DMA_RMP	USART2_DMA_RMP	SPI2_DMA_RMP	I2C_PA10_FMP	I2C_PA9_FMP	I2C2_FMP	I2C1_FMP	I2C_PB9_FMP	I2C_PB8_FMP	I2C_PB7_FMP	I2C_PB6_FMP
	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	TIM17_DMA_RMP2	TIM16_DMA_RMP2	TIM17_DMA_RMP	TIM16_DMA_RMP	USART1_RX_DMA_RMP	USART1_TX_DMA_RMP	ADC_DMA_RMP	IR_MOD [1:0]		Res.	PA11_PA12_RMP	Res.	Res.	MEM_MODE [1:0]	
	rw	rw	rw	rw	rw	rw	rw	rw			rw			rw	rw

Бит **31** Зарезервирован, он должен храниться при значении сброса.

Бит **30** TIM3_DMA_RMP: бит 3 переназначения запроса DMA TIM3. Доступно только для устройств STM32F07x.

Этот бит устанавливается и очищается программным обеспечением. Он контролирует переназначение запросов TIM3 DMA.

0: нет переназначения (запросы TIM3_CH1 и TIM3_TRIG DMA отображаются на канале DMA 4)

1: Remap (запросы TIM3_CH1 и TIM3_TRIG DMA, отображаемые на канале DMA 6)

Бит **29** TIM2_DMA_RMP: бит переназначения запроса TIM2 DMA. Доступно только для устройств STM32F07x.

Этот бит устанавливается и очищается программным обеспечением. Он контролирует переназначение запросов TIM2 DMA.

0: нет переназначения (запросы TIM2_CH2 и TIM2_CH4 DMA отображаются на DMA-канал 3 и 4 соответственно)

1: Remap (запросы TIM2_CH2 и TIM2_CH4 DMA, отображаемые на канале DMA 7)

Бит **28** TIM1_DMA_RMP: бит повторного набора запроса TIM1 DMA. Доступно только для устройств STM32F07x.

Этот бит устанавливается и очищается программным обеспечением. Он контролирует переназначение запросов TIM1 DMA.

0: нет переназначения (запросы TIM1_CH1, TIM1_CH2 и TIM1_CH3 DMA, отображаемые на каналах DMA 2, 3 и 4 соответственно)

1: Remap (TIM1_CH1, TIM1_CH2 и TIM1_CH3 запросы DMA, отображаемые на канале DMA 6)

Бит **27** I2C1_DMA_RMP: бит переадресации запроса DMA I2C1. Доступно только для устройств STM32F07x.

Этот бит устанавливается и очищается программным обеспечением. Он контролирует переназначение запросов DMA I2C1.

0: Нет переназначения (запросы I2C1_RX и I2C1_TX DMA отображаются на DMA-канал 3 и 2 соответственно)

1: Remap (I2C1_RX и I2C1_TX запросы DMA отображаются на DMA-канал 7 и 6 соответственно)

Бит **26** USART3_DMA_RMP: бит переназначения запроса USART3 DMA. Доступно только для устройств STM32F07x.

Этот бит устанавливается и очищается программным обеспечением. Он контролирует переназначение запросов USART3 DMA.

0: (USART3_RX и USART3_TX запросы DMA отображаются на канале DMA 6 и 7 соответственно)

1: Remap (USART3_RX и USART3_TX запросы DMA отображаются на каналах DMA 3 и 2 соответственно)

Бит **25** USART2_DMA_RMP: бит переназначения запроса USART2 DMA. Доступно только для устройств STM32F07x.

Этот бит устанавливается и очищается программным обеспечением. Он контролирует переназначение запросов USART2 DMA.

0: Нет переназначения (запросы USART2_RX и USART2_TX DMA отображаются на DMA-канале 5 и 4 соответственно)

1: Запросы Remap (USART2_RX и USART2_TX DMA, отображаемые на каналах 6 и 7 DMA соответственно)

Бит **24** SPI2_DMA_RMP: бит повторного преобразования запроса DMA SPI2. Доступно только для устройств STM32F07x.

Этот бит устанавливается и очищается программным обеспечением. Он контролирует переназначение запросов SPMA2 DMA.

0: Нет переназначения (запросы SPI2_RX и SPI2_TX DMA отображаются на DMA-канале 4 и 5 соответственно)

1: Remap (запросы SPI2_RX и SPI2_TX DMA, отображаемые на канале DMA 6 и 7 соответственно)

Биты **23:22** I2C_PAx_FMP: бит активации быстрого режима Plus (FM +). Доступно только для устройств STM32F03x, STM32F04x и STM32F09x.

Эти биты устанавливаются и очищаются программным обеспечением. Каждый бит позволяет использовать режим I2C FM + для ввода / вывода PA10 и PA9.

0: контакт PAx работает в стандартном режиме.

1: режим I2C FM + включен на выводе PAx, а управление скоростью отключено.

Бит **21** I2C2_FMP: активация FM + вождения для I2C2. Доступно только для устройств STM32F07x и STM32F09x.

Этот бит устанавливается и очищается программным обеспечением. Этот бит OR-ed с битами I2C_Pxx_FM +.

0: режим FM + управляется только битами I2C_Pxx_FM +.

1: Режим FM + включен на всех выводах I2C2, выбранных через биты выбора в регистре GPIOx_AFR. Это единственный способ включить режим FM + для пэдов без специального бита управления I2C_Pxx_FM +.

Бит **20** I2C1_FMP: активация возможностей FM + для I2C1. Недоступно на устройствах STM32F05x.

Этот бит устанавливается и очищается программным обеспечением. Этот бит OR-ed с битами I2C_Pxx_FM +.

0: режим FM + управляется только битами I2C_Pxx_FM +.

1: Режим FM + включен на всех выводах I2C1, выбранных с помощью битов выбора в регистрах GPIOx_AFR. Это единственный способ включить режим FM + для пэдов без специального бита управления I2C_Pxx_FM +.

Биты **19:16** I2C_PBx_FMP: бит активации быстрого режима Plus (FM +).

Эти биты устанавливаются и очищаются программным обеспечением. Каждый бит позволяет использовать режим I2C FM + для ввода / вывода PB6, PB7, PB8 и PB9.

0: контакт PBx работает в стандартном режиме.

1: режим I2C FM + включен на выводе PBx, а управление скоростью отключено.

Бит **15** Зарезервирован, должен храниться при значении сброса.

Бит **14** TIM17_DMA_RMP2: бит переназначения запроса DMA с запросом TIM17. Доступно только для устройств STM32F07x.

Этот бит устанавливается и очищается программным обеспечением. Он контролирует альтернативное переназначение запросов TIM17 DMA.

0: нет альтернативного переназначения (запросы TIM17 DMA отображаются в соответствии с бит TIM17_DMA_RMP)

1: альтернативный переназначение (запросы TIM17_CH1 и TIM17_UP DMA, отображаемые на канале DMA 7)

Бит **13** TIM16_DMA_RMP2: бит повторного набора запроса DMA с чередованием TIM16. Доступно только для устройств STM32F07x.

Этот бит устанавливается и очищается программным обеспечением. Он управляет альтернативным переназначением запросов TIM16 DMA.

0: нет альтернативного переназначения (запросы TIM16 DMA отображаются в соответствии с битом TIM16_DMA_RMP)

1: альтернативный переназначение (запросы TIM16_CH1 и TIM16_UP DMA, отображаемые на канале DMA 6)

Бит **12** TIM17_DMA_RMP: бит переназначения запроса TIM17 DMA. Доступно только для устройств STM32F03x, STM32F04x, STM32F05x и STM32F07x.

Этот бит устанавливается и очищается программным обеспечением. Он контролирует переназначение запросов TIM17 DMA.

0: нет переназначения (запросы TIM17_CH1 и TIM17_UP DMA отображаются на канале DMA 1)

1: Ремар (запросы TIM17_CH1 и TIM17_UP DMA, отображаемые на канале DMA 2)

Бит **11** TIM16_DMA_RMP: бит повторного набора запроса TIM16 DMA. Доступно только для устройств STM32F03x, STM32F04x, STM32F05x и STM32F07x.

Этот бит устанавливается и очищается программным обеспечением. Он контролирует переназначение запросов TIM16 DMA.

0: нет переназначения (TIM16_CH1 и TIM16_UP запросы DMA отображаются на канале DMA 3)

1: Ремар (запросы TIM16_CH1 и TIM16_UP DMA, отображаемые на канале DMA 4)

Бит **10** USART1_RX_DMA_RMP: USART1_RX бит переназначения запроса DMA. Доступно только для устройств STM32F03x, STM32F04x, STM32F05x и STM32F07x.

Этот бит устанавливается и очищается программным обеспечением. Он контролирует переназначение USART1_RX DMA

Запросы.

0: нет переназначения (запрос USART1_RX DMA, отображаемый на канале DMA 3)

1: Ремар (запрос USART1_RX DMA, отображаемый на канале DMA 5)

Бит **9** USART1_TX_DMA_RMP: USART1_TX бит повторного набора запроса DMA. Доступно только для устройств STM32F03x, STM32F04x, STM32F05x и STM32F07x.

Этот бит устанавливается и очищается программным обеспечением. Он бит контролирует переназначение запросов USART1_TX DMA.

0: Нет переназначения (запрос USART1_TX DMA, отображаемый на канале DMA 2)

1: Ремар (запрос USART1_TX DMA, отображаемый на канале DMA 4)

Бит **8** ADC_DMA_RMP: бит переназначения запроса DMA ADC. Доступно только для устройств STM32F03x, STM32F04x, STM32F05x и STM32F07x.

Этот бит устанавливается и очищается программным обеспечением. Он управляет переназначением запросов ADM DMA.

0: Нет переназначения (запрос ADMA DMA, отображаемый на канале DMA 1)

1: Ремар (запрос ADMA DMA, отображаемый на канале DMA 2)

Биты **7: 6** IR_MOD [1: 0]: ИК-модуляция Выбор сигнала конверта. Доступно только для устройств STM32F09x.

Эти биты позволяют выбирать сигнал огибающей модуляции между TIM16, USART1 и USART4:

00: выбран TIM16

01: выбран USART1

10: выбран USART4

11: Зарезервировано

Бит **5** Зарезервирован, он должен храниться при значении сброса.

Бит **4** PA11_PA12_RMP: бит для переназначения PA11 и PA12 для небольших пакетов (28 и 20 контактов).

Доступно только для устройств STM32F04x.

Этот бит устанавливается и очищается программным обеспечением. Он управляет отображением пары пассивов PA9 / 10 или PA11 / 12 в небольших пакетах с подсчетом.

0: нет ремар (палец PA9 / 10, нанесенный на контакты)

1: Ремар (палец PA11 / 12, отображаемый вместо PA9 / 10)

Биты **3: 2** Зарезервировано, должно храниться при значении сброса.

Биты **1: 0** MEM_MODE [1: 0]: бит выбора отображения памяти

Эти биты устанавливаются и очищаются программным обеспечением. Они управляют внутренним отображением памяти по адресу 0x0000 0000. После сброса эти биты принимают значение, выбранное по фактической конфигурации режима загрузки. Подробнее см. Главу 2.5: Конфигурация загрузки.

x0: основная флэш-память отображается на 0x0000 0000

01: Системная флэш-память, отображаемая на 0x0000 0000

11: встроенный SRAM, отображаемый на 0x0000 0000

Pg 168/1004

Table 30. Summary of the DMA requests for each channel on STM32F07x devices

Peripherals	Channel 1	Channel 2	Channel 3	Channel 4	Channel 5	Channel 6	Channel 7
ADC	ADC ⁽¹⁾	ADC ⁽²⁾	Reserved	Reserved	Reserved	Reserved	Reserved
SPI	Reserved	SPI1_RX	SPI1_TX	SPI2_RX ⁽¹⁾	SPI2_TX ⁽¹⁾	SPI2_RX ⁽²⁾	SPI2_TX ⁽²⁾
USART	Reserved	USART1_TX ⁽¹⁾ USART3_TX ⁽²⁾	USART1_RX ⁽¹⁾ USART3_RX ⁽²⁾	USART1_TX ⁽²⁾ USART2_TX ⁽¹⁾	USART1_RX ⁽²⁾ USART2_RX ⁽¹⁾	USART2_RX ⁽²⁾ USART3_RX ⁽¹⁾ USART4_RX	USART2_TX ⁽²⁾ USART3_TX ⁽¹⁾ USART4_TX
I2C	Reserved	I2C1_TX ⁽¹⁾	I2C1_RX ⁽¹⁾	I2C2_TX	I2C2_RX	I2C1_TX ⁽²⁾	I2C1_RX ⁽²⁾
TIM1	Reserved	TIM1_CH1 ⁽¹⁾	TIM1_CH2 ⁽¹⁾	TIM1_CH4 TIM1_TRIG TIM1_COM	TIM1_CH3 ⁽¹⁾ TIM1_UP	TIM1_CH1 ⁽²⁾ TIM1_CH2 ⁽²⁾ TIM1_CH3 ⁽²⁾	Reserved
TIM2	TIM2_CH3	TIM2_UP	TIM2_CH2 ⁽¹⁾	TIM2_CH4 ⁽¹⁾	TIM2_CH1	Reserved	TIM2_CH2 ⁽²⁾ TIM2_CH4 ⁽²⁾
TIM3	Reserved	TIM3_CH3	TIM3_CH4 TIM3_UP	TIM3_CH1 ⁽¹⁾ TIM3_TRIG ⁽¹⁾	Reserved	TIM3_CH1 ⁽²⁾ TIM3_TRIG ⁽²⁾	Reserved

Table 30. Summary of the DMA requests for each channel on STM32F07x devices (continued)

Peripherals	Channel 1	Channel 2	Channel 3	Channel 4	Channel 5	Channel 6	Channel 7
TIM6 / DAC	Reserved	Reserved	TIM6_UP DAC_Channel1	Reserved	Reserved	Reserved	Reserved
TIM7 / DAC	Reserved	Reserved	Reserved	TIM7_UP DAC_Channel2	Reserved	Reserved	Reserved
TIM15	Reserved	Reserved	Reserved	Reserved	TIM15_CH1 TIM15_UP TIM15_TRIG TIM15_COM	Reserved	Reserved
TIM16	Reserved	Reserved	TIM16_CH1 ⁽¹⁾ TIM16_UP ⁽¹⁾	TIM16_CH1 ⁽²⁾ TIM16_UP ⁽²⁾	Reserved	TIM16_CH1 ⁽³⁾ TIM16_UP ⁽³⁾	Reserved
TIM17	TIM17_CH1 ⁽¹⁾ TIM17_UP ⁽¹⁾	TIM17_CH1 ⁽²⁾ TIM17_UP ⁽²⁾	Reserved	Reserved	Reserved	Reserved	TIM17_CH1 ⁽³⁾ TIM17_UP ⁽³⁾

Контроллеры DMA1 / DMA2 на устройствах STM32F09x

Эта глава действительна только для устройств STM32F09x.

STM32F09x включает в себя два независимых DMA-контроллера DMA1 и DMA2.

Запросы аппаратного обеспечения от периферийных устройств (TIMx, ADC, DAC, SPI, I2C и USARTx) отображаются в DMAx-каналах (DMA1 от 1 до 7 и DMA2 от 1 до 5) через регистры выбора канала DMAx. На одном канале одновременно должен быть включен только один запрос. См. Рис. 22: Архитектура маршрутизации запроса DMAx на устройствах STM32F09x.

Периферийные запросы DMA могут быть независимо активированы / деактивированы путем программирования бит управления DMA в регистрах соответствующего периферийного устройства.

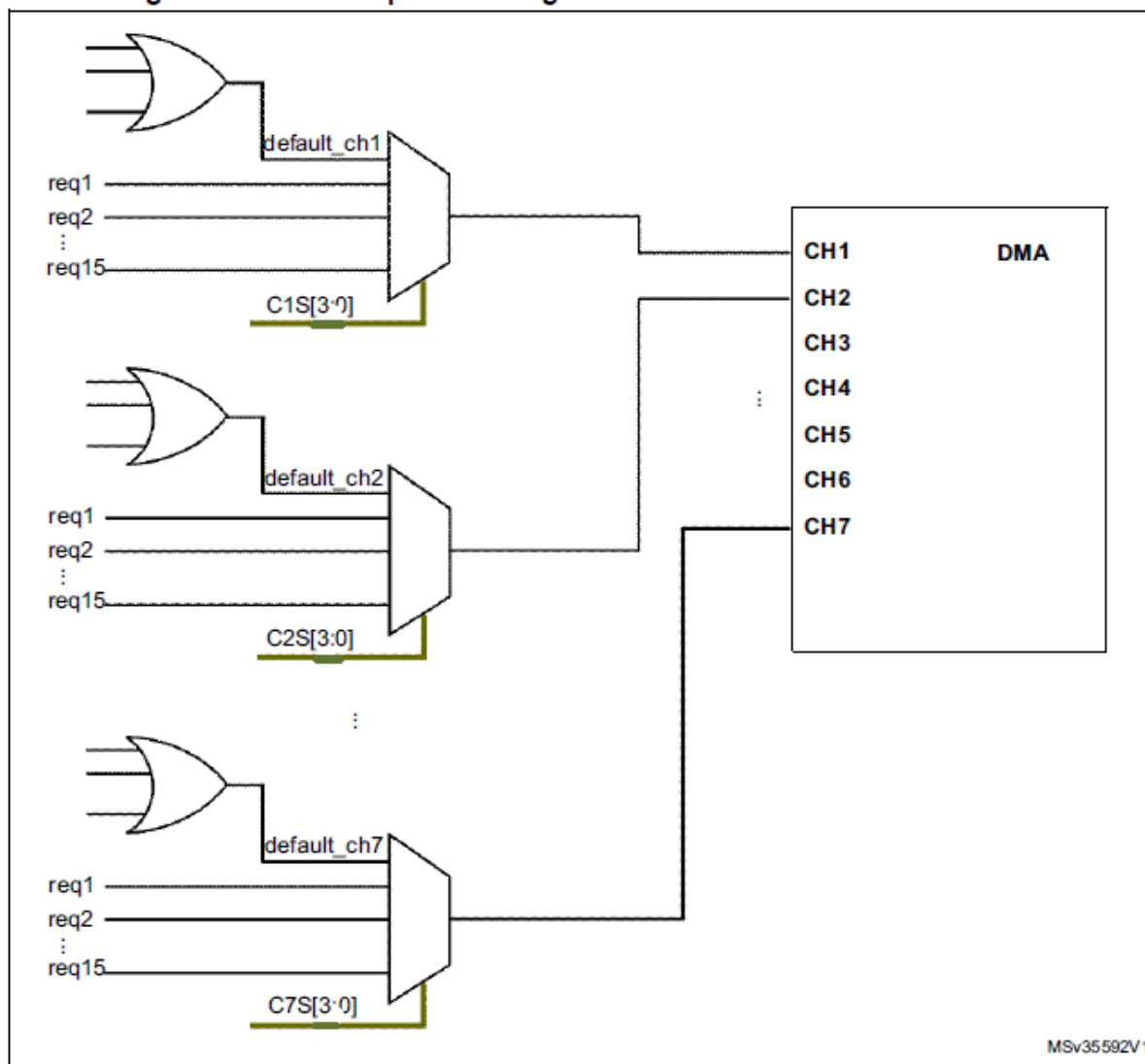
Позиция отображения по умолчанию 0 обеспечивает совместимость с отображением DMA, используемым в других продуктах STM32F0xx. Аппаратные запросы от периферийных устройств

(TIMx, ADC, DAC, SPI, I2C и USARTx) просто логически ORed перед входом в DMA. Это означает, что на одном канале одновременно должен быть включен только один запрос.

Альтернативные позиции сопоставления с 1 по 15 обеспечивают большую гибкость для сопоставления запросов аппаратного обеспечения по каналам DMA. Когда альтернативная позиция сопоставления используется для некоторых периферийных устройств, один и тот же запрос удаляется из позиции отображения по умолчанию, чтобы избежать конфликтов.

В таблице 31 и таблице 32 перечислены запросы DMA для каждого канала и альтернативной позиции.

Figure 22. DMAx request routing architecture on STM32F09x devices



1. Каналы 6 и 7 недоступны на DMA2.
2. Как только какой-либо запрос DMA выбран в позиции 1-15, он исчезает из местоположения по умолчанию в позиции 0.

Table 31. Summary of the DMA1 requests for each channel on STM32F09x devices

CxS [3:0]	Channel 1	Channel 2	Channel 3	Channel 4	Channel 5	Channel 6	Channel 7
0000	TIM2_CH3	TIM2_UP TIM3_CH3	TIM3_CH4 TIM3_UP	TIM1_CH4 TIM1_TRIG TIM1_COM	TIM1_UP	-	-
	-				TIM2_CH1	-	-
	-				TIM15_CH1 TIM15_UP TIM15_TRIG TIM15_COM	-	-
	ADC	-	TIM6_UP DAC_ Channel1	TIM7_UP DAC_ Channel2	-	-	-
	-	USART1_TX	USART1_RX	USART2_TX	USART2_RX	USART3_RX	USART3_TX
	-	-	-	-	-	USART4_RX	USART4_TX
	-	SPI1_RX	SPI1_TX	SPI2_RX	SPI2_TX	-	-
	-	I2C1_TX	I2C1_RX	I2C2_TX	I2C2_RX	-	-
	-	TIM1_CH1	TIM1_CH2	-	TIM1_CH3	-	-
	-	-	TIM2_CH2	TIM2_CH4	-	-	-
	TIM17_CH1 TIM17_UP	-	TIM16_CH1 TIM16_UP	TIM3_CH1 TIM3_TRIG	-	-	-
0001	ADC	ADC	TIM6_UP DAC_ Channel1	TIM7_UP DAC_ Channel2	-	-	-
0010	-	I2C1_TX	I2C1_RX	I2C2_TX	I2C2_RX	I2C1_TX	I2C1_RX
0011	-	SPI1_RX	SPI1_TX	SPI2_RX	SPI2_TX	SPI2_RX	SPI2_TX
0100	-	TIM1_CH1	TIM1_CH2	-	TIM1_CH3	TIM1_CH1 TIM1_CH2 TIM1_CH3	-
0101	-	-	TIM2_CH2	TIM2_CH4	-	-	TIM2_CH2 TIM2_CH4
0110	-	-	-	TIM3_CH1 TIM3_TRIG	-	TIM3_CH1 TIM3_TRIG	-
0111	TIM17_CH1 TIM17_UP	TIM17_CH1 TIM17_UP	TIM16_CH1 TIM16_UP	TIM16_CH1 TIM16_UP	-	TIM16_CH1 TIM16_UP	TIM17_CH1 TIM17_UP
1000	USART1_ RX	USART1_TX	USART1_RX	USART1_TX	USART1_RX	USART1_RX	USART1_TX
1001	USART2_ RX	USART2_TX	USART2_RX	USART2_TX	USART2_RX	USART2_RX	USART2_TX
1010	USART3_ RX	USART3_TX	USART3_RX	USART3_TX	USART3_RX	USART3_RX	USART3_TX
1011	USART4_ RX	USART4_TX	USART4_RX	USART4_TX	USART4_RX	USART4_RX	USART4_TX
1100	USART5_ RX	USART5_TX	USART5_RX	USART5_TX	USART5_RX	USART5_RX	USART5_TX
1101	USART6_ RX	USART6_TX	USART6_RX	USART6_TX	USART6_RX	USART6_RX	USART6_TX

Table 31. Summary of the DMA1 requests for each channel on STM32F09x devices (continued)

CxS [3:0]	Channel 1	Channel 2	Channel 3	Channel 4	Channel 5	Channel 6	Channel 7
1110	USART7_RX	USART7_TX	USART7_RX	USART7_TX	USART7_RX	USART7_RX	USART7_TX
1111	USART8_RX	USART8_TX	USART8_RX	USART8_TX	USART8_RX	USART8_RX	USART8_TX

Table 32. Summary of the DMA2 requests for each channel on STM32F09x devices

CxS [3:0]	Channel 1	Channel 2	Channel 3	Channel 4	Channel 5
0000	none				
0001	-	-	TIM6_UP DAC_Channel1	TIM7_UP DAC_Channel2	ADC
0010	I2C2_TX	I2C2_RX	-	-	-
0011	-	-	SPI1_RX	SPI1_TX	-
0100	-	-	-	-	-
0101	-	-	-	-	-
0110	-	-	-	-	-
0111	-	-	-	-	-
1000	USART1_TX	USART1_RX	USART1_RX	USART1_TX	USART1_TX
1001	USART2_TX	USART2_RX	USART2_RX	USART2_TX	USART2_TX
1010	USART3_TX	USART3_RX	USART3_RX	USART3_TX	USART3_TX
1011	USART4_TX	USART4_RX	USART4_RX	USART4_TX	USART4_TX
1100	USART5_TX	USART5_RX	USART5_RX	USART5_TX	USART5_TX
1101	USART6_TX	USART6_RX	USART6_RX	USART6_TX	USART6_TX
1110	USART7_TX	USART7_RX	USART7_RX	USART7_TX	USART7_TX
1111	USART8_TX	USART8_RX	USART8_RX	USART8_TX	USART8_TX

10.4 Регистры DMA

Обратитесь к разделу 1.1 на стр. 42 за списком сокращений, используемых в описаниях регистров.

К периферийным регистрам можно обращаться по байтам (8 бит), полусловам (16 бит) или словам (32 бит).

10.4.1 Регистр состояния прерывания DMA (DMA_ISR и DMA2_ISR)

Смещение адреса: 0x00

Значение сброса: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	TEIF7	HTIF7	TCIF7	GIF7	TEIF6	HTIF6	TCIF6	GIF6	TEIF5	HTIF5	TCIF5	GIF5
				r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TEIF4	HTIF4	TCIF4	GIF4	TEIF3	HTIF3	TCIF3	GIF3	TEIF2	HTIF2	TCIF2	GIF2	TEIF1	HTIF1	TCIF1	GIF1
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

Биты **31: 28** Зарезервировано, должно храниться при значении сброса.

Биты **27, 23, 19, 15, 11, 7, 3**

TEIFx: флаг ошибки передачи канала x (x = 1..7 для DMA и x = 1..5 для DMA2). Этот бит устанавливается аппаратным обеспечением. Он очищается с помощью программного обеспечения, записывающего 1, в соответствующий бит в регистре DMA_IFCR.

0: Ошибка передачи (TE) на канале x

1: Ошибка передачи (TE) произошла на канале x

Биты **26, 22, 18, 14, 10, 6, 2**

HTIFx: флажок передачи половины канала x (x = 1..7 для DMA и x = 1..5 для DMA2). Этот бит устанавливается с помощью аппаратного обеспечения. Он очищается с помощью программного обеспечения, записывающего 1, в соответствующий бит в регистре DMA_IFCR.

0: нет события с половинной передачей (HT) на канале x

1: Событие с половинной передачей (HT) произошло на канале x

Биты **25, 21, 17, 13, 9, 5, 1**

TCIFx: флаг полного переноса канала x (x = 1..7 для DMA и x = 1..5 для DMA2). Этот бит устанавливается с помощью аппаратного обеспечения. Он очищается с помощью программного обеспечения, записывающего 1, в соответствующий бит в регистре DMA_IFCR.

0: нет события завершения передачи (TC) на канале x

1: Событие передачи завершено (TC) произошло на канале x

Биты **24, 20, 16, 12, 8, 4, 0**

GIFx: Глобальный флаг прерывания канала x (x = 1..7 для DMA и x = 1..5 для DMA2) Этот бит устанавливается аппаратным обеспечением. Он очищается с помощью программного обеспечения, записывающего 1, в соответствующий бит в регистре DMA_IFCR.

0: нет событий TE, HT или TC на канале x

1: Событие TE, HT или TC произошло на канале x

10.4.2. Регистр очистки флага прерывания DMA (DMA_IFCR и DMA2_IFCR)

Смещение адреса: 0x04

Значение сброса: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	CTEIF7	CHTIF7	CTCIF7	CGIF7	CTEIF6	CHTIF6	CTCIF6	CGIF6	CTEIF5	CHTIF5	CTCIF5	CGIF5
				w	w	w	w	w	w	w	w	w	w	w	w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CTEIF4	CHTIF4	CTCIF4	CGIF4	CTEIF3	CHTIF3	CTCIF3	CGIF3	CTEIF2	CHTIF2	CTCIF2	CGIF2	CTEIF1	CHTIF1	CTCIF1	CGIF1
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

Биты **31: 28** Зарезервировано, должно храниться при значении сброса.

Биты **27, 23, 19, 15, 11, 7, 3**

CTEIFx: Ошибка передачи канала x (x = 1..7 для DMA и x = 1..5 для DMA2)

Этот бит устанавливается программным обеспечением.

0: нет эффекта

1: очищает соответствующий флаг TEIF в регистре DMA_ISR. Биты 26, 22, 18, 14, 10, 6, 2

CHTIFx: передача x x x x x x x x (x = 1..7 для DMA и x = 1..5 для DMA2)

Этот бит устанавливается программным обеспечением.

0: нет эффекта

1: очищает соответствующий флаг HTIF в регистре DMA_ISR

Биты **25, 21, 17, 13, 9, 5, 1**

CTCIFx: передача канала x прозрачна (x = 1..7 для DMA и x = 1..5 для DMA2)

Этот бит устанавливается программным обеспечением.

0: нет эффекта

1: очищает соответствующий флаг TCIF в регистре DMA_ISR

Биты **24, 20, 16, 12, 8, 4, 0**

CGIFx: очистить глобальное прерывание канала x (x = 1..7 для DMA и x = 1..5 для DMA2)

Этот бит устанавливается программным обеспечением.

0: нет эффекта

1: очищает флаги GIF, TEIF, HTIF и TCIF в регистре DMA_ISR

10.4.3 Регистр конфигурации DMA-канала x (DMA_CCRx и DMA2_CCRx)

(x = 1..7 для DMA и x = 1..5 для DMA2, где x = номер канала)

Смещение адреса: $0x08 + 0d20 \times (\text{номер канала} - 1)$

Значение сброса: $0x0000\ 0000$

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	MEM2 MEM	PL[1:0]		MSIZE[1:0]		PSIZE[1:0]		MINC	PINC	CIRC	DIR	TEIE	HTIE	TCIE	EN
	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Биты **31:15** Зарезервировано, должно храниться при значении сброса.

Бит **14**

MEM2MEM: режим памяти в память

Этот бит устанавливается и очищается программным обеспечением.

0: режим памяти в память отключен

1: включен режим памяти в память.

Биты **13:12**

PL [1: 0]: уровень приоритета канала

Эти биты устанавливаются и очищаются программным обеспечением.

00: Низкий

01: Средний

10: Высокий

11: Очень высокий

Биты **11:10**

MSIZE [1: 0]: Размер памяти

Эти биты устанавливаются и очищаются программным обеспечением.

00: 8 бит

01: 16 бит

10: 32 бит

11: Зарезервировано

Биты **9: 8**

PSIZE [1: 0]: размер периферии

Эти биты устанавливаются и очищаются программным обеспечением.

00: 8 бит

01: 16 бит

10: 32 бит

11: Зарезервировано

Бит **7**

MINC: Режим увеличения памяти

Этот бит устанавливается и очищается программным обеспечением.

0: Режим увеличения памяти отключен

1: включен режим увеличения памяти

Бит **6**

PINC: Периферийный инкрементный режим

Этот бит устанавливается и очищается программным обеспечением.

0: отключен периферийный режим увеличения

1: включен режим расширенного периферийного устройства

Бит 5

CIRC: круговой режим

Этот бит устанавливается и очищается программным обеспечением.

0: отключен круговой режим

1: включен круговой режим

Бит 4

DIR: направление передачи данных

Этот бит устанавливается и очищается программным обеспечением.

0: чтение с периферийных устройств

1: чтение из памяти

Бит 3

TEIE: Разрешение прерывания передачи ошибок

Этот бит устанавливается и очищается программным обеспечением.

0: прерывание TE отключено

1: прерывание TE

Бит 2

HTIE: разрешение прерывания посылки

Этот бит устанавливается и очищается программным обеспечением.

0: прерывание HT отключено

1: Прерывание HT включено

Бит 1

TCIE: передача полного прерывания

Этот бит устанавливается и очищается программным обеспечением.

0: прерывание TC отключено

1: прерывание TC

Бит 0

EN: Разрешение канала

Этот бит устанавливается и очищается программным обеспечением.

0: отключен канал

1: Канал включен

10.4.4 DMA-канал x число регистра данных (DMA_CNDTRx и DMA2_CNDTRx) (x = 1..7 для DMA и x = 1..5 для DMA2, где x = номер канала)

Смещение адреса: $0x0C + 0d20 \times (\text{номер канала} - 1)$

Значение сброса: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NDT[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Биты **31: 16** Сохраняются, должны сохраняться при значении сброса.

Биты **15: 0**

NDT [15: 0]: количество передаваемых данных

Количество передаваемых данных (от 0 до 65535). Этот регистр может быть записан только в том случае, если канал отключен. Как только канал включен, этот регистр доступен только для

чтения, указывая оставшиеся байты, которые должны быть переданы. Этот регистр уменьшается после каждой передачи DMA.

Как только передача завершена, этот регистр может либо оставаться на нуле, либо перезагружаться автоматически с помощью ранее запрограммированного значения, если канал настроен в круговом режиме.

Если этот регистр равен нулю, транзакция не может быть выполнена независимо от того, включен ли канал или нет.

10.4.5. Адрес периферийного адреса DMA-канала x (DMA_CPARx и DMA2_CPARx) (x = 1..7 для DMA и x = 1..5 для DMA2, где x = номер канала)

Смещение адреса: $0x10 + 0d20 \times (\text{номер канала} - 1)$

Значение сброса: 0x0000 0000

Этот регистр не должен быть записан, когда канал включен.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PA [31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PA [15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Биты 31: 0

PA [31: 0]: Периферийный адрес

Базовый адрес регистра периферийных данных, с которого данные будут считаны / записаны.

Когда PSIZE равно 01 (16 бит), бит PA [0] игнорируется. Доступ автоматически выравнивается по полуслову.

Когда PSIZE равно 10 (32 бит), PA [1: 0] игнорируются. Доступ автоматически выравнивается со словом.

10.4.6 Регистр адресов памяти DMA-канала x (DMA_CMARx и DMA2_CMARx) (x = 1..7 для DMA и x = 1..5 для DMA2, где x = номер канала)

Смещение адреса: $0x14 + 0d20 \times (\text{номер канала} - 1)$

Значение сброса: 0x0000 0000

Этот регистр не должен быть записан, когда канал включен.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MA [31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MA [15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Биты 31: 0

MA [31: 0]: адрес памяти

Базовый адрес области памяти, с которой данные будут считаны / записаны.

Когда MSIZE равно 01 (16 бит), бит MA [0] игнорируется. Доступ автоматически выравнивается по полуслову.

Когда MSIZE равно 10 (32 бит), MA [1: 0] игнорируются. Доступ автоматически выравнивается со словом.

10.4.7 Регистр выбора канала DMA (DMA_CSELR и DMA2_CSELR)

Этот регистр присутствует только на устройствах STM32F09x.

Смещение адреса: 0xA8

Значение сброса: 0x00000000

Этот регистр используется для управления переназначением каналов DMA (см. Рис. 22).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	C7S [3:0]				C6S [3:0]				C5S [3:0]			
				rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
C4S [3:0]				C3S [3:0]				C2S [3:0]				C1S [3:0]			
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Биты 31: 28 Зарезервировано, должно храниться при значении сброса.

Биты 27:24 C7S [3: 0]: выбор канала DMA 7

DMA-запрос для канала 7 (1). Недоступно на DMA2

Биты 23:20 C6S [3: 0]: выбор канала DMA 6

Отображение запроса DMA для канала 6 (1). Недоступно на DMA2

Биты 19:16 C5S [3: 0]: выбор канала DMA 5

Отображение запроса DMA для канала 5 (1)

Биты 15:12 C4S [3: 0]: выбор канала DMA 4

Отображение запроса DMA для канала 4 (1)

Биты 11: 8 C3S [3: 0]: выбор канала DMA 3

Отображение запроса DMA для канала 3 (1)

Биты 7: 4 C2S [3: 0]: выбор канала DMA

Отображение запроса DMA для канала 2 (1)

Биты 3: 0 C1S [3: 0]: выбор канала DMA 1

Отображение запроса DMA для канала 1 (1)

1. Для конкретного сопоставления запросов DMA см. Таблицу 31: Сводка запросов DMA1 для каждого канала на устройствах STM32F09x и Table32: Сводка запросов DMA2 для каждого канала на устройствах STM32F09x.

10.4.8 DMA register map

The following table gives the DMA register map and the reset values.

Table 33. DMA register map and reset values

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	DMA_ISR	Res	Res	Res	Res	TEIF7	HTIF7	TCIF7	GIF7	TEIF6	HTIF6	TCIF6	GIF6	TEIF5	HTIF5	TCIF5	GIF5	TEIF4	HTIF4	TCIF4	GIF4	TEIF3	HTIF3	TCIF3	GIF3	TEIF2	HTIF2	TCIF2	GIF2	TEIF1	HTIF1	TCIF1	GIF1
	Reset value					0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x04	DMA_IFCR	Res	Res	Res	Res	CTEIF7	CHTIF7	CTCIF7	CGIF7	CTEIF6	CHTIF6	CTCIF6	CGIF6	CTEIF5	CHTIF5	CTCIF5	CGIF5	CTEIF4	CHTIF4	CTCIF4	CGIF4	CTEIF3	CHTIF3	CTCIF3	CGIF3	CTEIF2	CHTIF2	CTCIF2	CGIF2	CTEIF1	CHTIF1	CTCIF1	CGIF1
	Reset value					0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x08	DMA_CCR1	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	MEM2MEM	PL [1:0]	MSIZE [1:0]	PSIZE [1:0]	MINC	PINC	CIRC	DIR	TEIE	HTIE	TCIE	EN				
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x0C	DMA_CNDTR1	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	NDT[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x10	DMA_CPAR1	PA[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x14	DMA_CMAR1	MA[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x18	Reserved	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res
0x1C	DMA_CCR2	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	MEM2MEM	PL [1:0]	MSIZE [1:0]	PSIZE [1:0]	MINC	PINC	CIRC	DIR	TEIE	HTIE	TCIE	EN				
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x20	DMA_CNDTR2	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	NDT[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x24	DMA_CPAR2	PA[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x28	DMA_CMAR2	MA[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x2C	Reserved	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res
0x30	DMA_CCR3	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	MEM2MEM	PL [1:0]	MSIZE [1:0]	PSIZE [1:0]	MINC	PINC	CIRC	DIR	TEIE	HTIE	TCIE	EN				
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x34	DMA_CNDTR3	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	NDT[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x38	DMA_CPAR3	PA[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x3C	DMA_CMAR3	MA[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 33. DMA register map and reset values (continued)

Table 34. DMA register map and reset values (registers available on STM32F07x and STM32F09x devices only) (continued)

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
0x84	DMA_CNDTR7	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	NDT[15:0]																
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x88	DMA_CPAR7	PA[31:0]																																
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x8C	DMA_CMAR7	MA[31:0]																																
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x90	Reserved																																	

Table 35. DMA register map and reset values (register available on STM32F09x devices only)

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0xA8	DMA_CSELR	Res	Res	Res	Res	C7S[3:0]				C6S[3:0]				C5S[3:0]				C4S[3:0]				C3S[3:0]				C2S[3:0]				C1S[3:0]			
	Reset value					0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Обратитесь к разделу 2.2.2 на стр. 46 для адресов границ регистров.

A.5.1 Пример кода последовательности конфигурации DMA-канала

/ * Следующий пример приведен для АЦП. Он может быть легко перенесен на любое периферийное устройство, поддерживающее передачу DMA связанного канала с периферийным устройством, это должно быть проверено в таблице данных. * /

/ * (1) Включить периферийные часы на DMA * /

/ * (2) Включить передачу DMA на АЦП * /

/ * (3) Настроить адрес регистра периферийных данных * /

/ * (4) Настройте адрес памяти * /

/ * (5) Настройте количество каналов DMA, которые будут выполняться на канале 1 * /

/ * (6) Настроить инкремент, размер и прерывания * /

/ * (7) Включить канал DMA 1 * /

RCC->AHBENR |= RCC_AHBENR_DMA1EN; /* (1) */

ADC1->CFGR1 |= ADC_CFGR1_DMAEN; /* (2) */

DMA1_Channel1->CPAR = (uint32_t)(&(ADC1->DR)); /* (3) */

DMA1_Channel1->CMAR = (uint32_t)(ADC_array); /* (4) */

DMA1_Channel1->CNDTR = 3; /* (5) */

DMA1_Channel1->CCR |= DMA_CCR_MINC | DMA_CCR_MSIZE_0 | DMA_CCR_PSIZE_0
| DMA_CCR_TEIE | DMA_CCR_TCIE; /* (6) */

DMA1_Channel1->CCR |= DMA_CCR_EN; /* (7) */

/ * Настройка NVIC для DMA * /

/ * (1) Включить прерывание по каналу DMA 1 * /

/ * (2) Установить приоритет для канала DMA 1 * /

NVIC_EnableIRQ(DMA1_Channel1_IRQn); /* (1) */

NVIC_SetPriority(DMA1_Channel1_IRQn,0); /* (2) */

A.7.9 Пример кода последовательности режима DMA с одним кадром


```

/* (1) Включить периферийные часы на DMA */
/* (2) Включить передачу DMA на АЦП - DMACFG поддерживается на 0 для одного режима
съемки */
/* (3) Настроить адрес регистра периферийных данных */
/* (4) Настройте адрес памяти */
/* (5) Настроить количество DMA-передачи, которое должно выполняться по каналу DMA 1 */
/* (6) Настроить инкремент, размер и прерывания */
/* (7) Включить канал DMA 1 */ RCC->AHBENR |= RCC_AHBENR_DMA1EN; /* (1) */
ADC1->CFGR1 |= ADC_CFGR1_DMAEN; /* (2) */
DMA1_Channel1->CPAR = (uint32_t)(&(ADC1->DR)); /* (3) */
DMA1_Channel1->CMAR = (uint32_t)(ADC_array); /* (4) */
DMA1_Channel1->CNDTR = NUMBER_OF_ADC_CHANNEL; /* (5) */
DMA1_Channel1->CCR |= DMA_CCR_MINC | DMA_CCR_MSIZE_0 | DMA_CCR_PSIZE_0
| DMA_CCR_TEIE | DMA_CCR_TCIE; /* (6) */
DMA1_Channel1->CCR |= DMA_CCR_EN; /* (7) */

```

A.7.10 Пример последовательности кода круговых режимов DMA

```

/* (1) Включить периферийные часы на DMA */
/* (2) Включить передачу DMA на АЦП и круговом режиме */
/* (3) Настроить адрес регистра периферийных данных */
/* (4) Настройте адрес памяти */
/* (5) Настроить количество DMA-передачи, которое должно выполняться по каналу DMA 1 */
/* (6) Настройка инкремента, размера, прерываний и кругового режима */
/* (7) Включить канал DMA 1 */
RCC->AHBENR |= RCC_AHBENR_DMA1EN; /* (1) */
ADC1->CFGR1 |= ADC_CFGR1_DMAEN | ADC_CFGR1_DMCFG; /* (2) */
DMA1_Channel1->CPAR = (uint32_t)(&(ADC1->DR)); /* (3) */
DMA1_Channel1->CMAR = (uint32_t)(ADC_array); /* (4) */
DMA1_Channel1->CNDTR = NUMBER_OF_ADC_CHANNEL; /* (5) */
DMA1_Channel1->CCR |= DMA_CCR_MINC | DMA_CCR_MSIZE_0 | DMA_CCR_PSIZE_0
| DMA_CCR_TEIE | DMA_CCR_CIRC; /* (6) */
DMA1_Channel1->CCR |= DMA_CCR_EN; /* (7) */

```

A.7.11 Пример последовательности кода ожидания

```

/* (1) Выберите HSI14, записав 00 в CKMODE (значение сброса) */
/* (2) Выберите непрерывный режим и режим ожидания */
/* (3) Выберите CHSEL1 / 2/3 */
ADC1->CFGR2 &= ~ADC_CFGR2_CKMODE; /* (1) */
ADC1->CFGR1 |= ADC_CFGR1_CONT | ADC_CFGR1_WAIT; /* (2) */
ADC1->CHSELR = ADC_CHSELR_CHSEL1 | ADC_CHSELR_CHSEL2
| ADC_CHSELR_CHSEL3; /* (3) */
ADC1->CR |= ADC_CR_ADSTART; /* start the ADC conversions */

```

A.8.12 Пример кода инициализации DMA

/* (1) Включить передачу DMA на ЦАП ch1 для обоих каналов, разрешить прерывание на DAC под управлением ЦАП, включить ЦАП ch1 и ch2, выбрать TIM7 в качестве триггера, записав 010 в TSEL1 и TSEL2 */

```
DAC->CR |= DAC_CR_TSEL1_1 | DAC_CR_TEN2 | DAC_CR_EN2
| DAC_CR_TSEL2_1 | DAC_CR_DMAUDRIE1 | DAC_CR_DMAEN1
| DAC_CR_TEN1 | DAC_CR_EN1; /* (1) */
/* (1) Включить периферийные часы на DMA */
/* (2) Настроить адрес регистра периферийных данных */
/* (3) Настроить адрес памяти */
/* (4) Настройте количество транков DMA, которое будет выполняться на канале 3 */
/* (5) Настроить инкремент, размер (32 бита), прерывания, передать из
память в периферийный и круговой режим */
/* (6) Включить канал DMA 3 */
RCC->AHBENR |= RCC_AHBENR_DMA1EN; /* (1) */
DMA1_Channel3->CPAR = (uint32_t) (&(DAC->DHR12RD)); /* (2) */
DMA1_Channel3->CMAR = (uint32_t) signal_data; /* (3) */
DMA1_Channel3->CNDTR = SIGNAL_ARRAY_SIZE; /* (4) */
DMA1_Channel3->CCR |= DMA_CCR_MINC | DMA_CCR_MSIZE_1 | DMA_CCR_PSIZE_1
| DMA_CCR_TEIE | DMA_CCR_CIRC; /* (5) */
DMA1_Channel3->CCR |= DMA_CCR_EN; /* (6) */
```

A.9.6 Вход PWM с примером кода конфигурации DMA

```
/* (1) Включить периферийные часы на DMA */
/* (2) Настроить адрес регистра периферийных данных для канала DMA x */
/* (3) Настройте адрес памяти для канала DMA x */
/* (4) Настроить количество передаваемых DMA-каналов
на канале DMA x */
/* (5) Не настраивайте прирост (значение сброса), размер (16 бит), прерывания, переход от
периферийного к памяти и круговой режим для канала DMA x */
/* (6) Включить канал DMA x */
RCC->AHBENR |= RCC_AHBENR_DMA1EN; /* (1) */
DMA1_Channel2->CPAR = (uint32_t) (&(TIM1->CCR1)); /* (2) */
DMA1_Channel2->CMAR = (uint32_t) (&Period); /* (3) */
DMA1_Channel2->CNDTR = 1; /* (4) */
DMA1_Channel2->CCR |= DMA_CCR_MSIZE_0 | DMA_CCR_PSIZE_0
| DMA_CCR_TEIE | DMA_CCR_CIRC; /* (5) */
DMA1_Channel2->CCR |= DMA_CCR_EN; /* (6) */
/* повторить (2) - (6) для канала 3 */
DMA1_Channel3->CPAR = (uint32_t) (&(TIM1->CCR2)); /* (2) */
DMA1_Channel3->CMAR = (uint32_t) (&DutyCycle); /* (3) */
DMA1_Channel3->CNDTR = 1; /* (4) */
DMA1_Channel3->CCR |= DMA_CCR_MSIZE_0 | DMA_CCR_PSIZE_0
| DMA_CCR_TEIE | DMA_CCR_CIRC; /* (5) */
DMA1_Channel3->CCR |= DMA_CCR_EN; /* (6) */
/* Настройка NVIC для DMA */
/* (7) Включить прерывание по каналам DMA x */
/* (8) Установить приоритет для каналов DMA x */
NVIC_EnableIRQ(DMA1_Channel2_3_IRQn); /* (7) */
```

```
NVIC_SetPriority(DMA1_Channel2_3_IRQn,3); /* (8) */
```

A.9.21 Пример кода функции всплеска DMA

```
/* В этом примере TIMx ранее был сконфигурирован в центрированном PWM */  
/* Настройка функции DMA Burst */  
/* Настроить соответствующий канал DMA */  
/* (1) Устанавливать периферийный адрес канала DMA является адресом регистра DMAR */  
/* (2) Установить адрес памяти канала DMA - адрес буфера в ОЗУ, содержащий данные,  
которые должны быть переданы DMA в регистры CCRx */  
/* (3) Задайте количество передач данных sizeof (Duty_Cycle_Table) */  
/* (4) Настроить передачу DMA в регистре CCR, включить круговой режим, установив бит CIRC  
(необязательно), установить размер памяти до 16_бит MSIZE = 01, установить размер периферии  
до 32_бит PSIZE = 10, включить режим увеличения памяти, установив MINC, задайте  
направление передачи данных, считанное из памяти, установив DIR. */  
/* (5) Настроить регистр TIMx_DCR с помощью DBL = 3 и DBA = (@ TIMx-> CCR2 - @ TIMx-> CR1)  
>> 2 = 0xE */  
/* (6) Включить запрос DMA для обновления TIMX, установив бит UDE в DIER  
регистр */  
/* (7) Включить TIMx */  
/* (8) Включить канал DMA */  
DMA1_Channel2->CPAR = (uint32_t)(TIMx->DMAR); /* (1) */  
DMA1_Channel2->CMAR = (uint32_t)(Duty_Cycle_Table); /* (2) */  
DMA1_Channel2->CNDTR = 10*3; /* (3) */  
DMA1_Channel2->CCR |= DMA_CCR_CIRC | DMA_CCR_MSIZ_0 | DMA_CCR_PSIZ_1  
| DMA_CCR_MINC | DMA_CCR_DIR; /* (4) */  
TIMx->DCR = (3 << 8)  
+ (((uint32_t)(TIMx->CCR2))  
- ((uint32_t)(TIMx->CR1))) >> 2); /* (5) */  
TIMx->DIER |= TIM_DIER_UDER; /* (6) */  
TIMx->CR1 |= TIM_CR1_CEN; /* (7) */  
DMA1_Channel2->CCR |= DMA_CCR_EN; /* (8) */
```

ИЗМЕНЕНИЯ

- Помечается как «зарезервировано»: биты 14 и 6 в разделе 2.6.3: TIM16 и TIM17
DMA / разрешения прерывания (TIM16_DIER и TIM17_DIER), бит 6 и 0
в разделе 2.6.4: регистр статуса TIM16 и TIM17 (TIM16_SR и
TIM17_SR) и бит 6 в разделе 2.6.5: генерация событий TIM16 и TIM17
(TIM16_EGR и TIM17_EGR).

A.14.8 I2C, сконфигурированный в основном режиме для передачи с использованием кода DMA

```
/* (1) Значение регистра синхронизации рассчитывается с помощью файла AN4235 xls,  
быстрый режим @ 400 кГц с I2CCLK = 48 МГц, время нарастания = 140 нс, время падения = 40 нс */  
/* (2) Periph enable */  
/* (3) Адрес ведомого = 0x5A, передача записи, 2 байта для передачи, автоподключение */  
I2C2->TIMINGR = (uint32_t)0x00B01A4B; /* (1) */  
I2C2->CR1 = I2C_CR1_PE | I2C_CR1_TXDMAEN; /* (2) */
```

```
I2C2->CR2 = I2C_CR2_AUTOEND | (SIZE_OF_DATA << 16)
| (I2C1_OWN_ADDRESS << 1); /* (3) */
```

A.14.9 I2C, сконфигурированный в подчиненном режиме для приема с примером кода DMA

```
/* (1) Значение регистра синхронизации рассчитывается с помощью файла AN4235 xls,
быстрый режим @ 400 кГц с I2CCLK = 48 МГц, время нарастания = 140 нс, время падения = 40 нс */
/* (2) Включить Periph, получить разрешение DMA */
/* (3) 7-битный адрес = 0x5A */
/* (4) Включить собственный адрес 1 */
/* (4) Включить собственный адрес 1 */
I2C1->TIMINGR = (uint32_t)0x00B00000; /* (1) */
I2C1->CR1 = I2C_CR1_PE | I2C_CR1_RXDMAEN | I2C_CR1_ADDRIE; /* (2) */
I2C1->OAR1 |= (uint32_t)(I2C1_OWN_ADDRESS << 1); /* (3) */
I2C1->OAR1 |= I2C_OAR1_OA1EN; /* (4) */
```

A.17.5. Конфигурация мастера SPI с примером кода DMA.

```
/* (1) Master selection, BR: Fpclk/256 (due to C27 on the board, SPI_CLK is set to the minimum)
CPOL and CPHA at zero (rising first edge) */
/* (2) TX и RX с DMA, включить вывод выбора подчиненного устройства, включить
прерывание RXNE, выбрать 8-битный Rx fifo */
/* (3) Включить SPI1 */
SPI1->CR1 = SPI_CR1_MSTR | SPI_CR1_BR; /* (1) */
SPI1->CR2 = SPI_CR2_TXDMAEN | SPI_CR2_RXDMAEN | SPI_CR2_SSOE
| SPI_CR2_RXNEIE | SPI_CR2_FRXTH
| SPI_CR2_DS_2 | SPI_CR2_DS_1 | SPI_CR2_DS_0; /* (2) */
SPI1->CR1 |= SPI_CR1_SPE; /* (3) */
```

A.17.6 Конфигурация slave-устройства SPI с примером кода DMA

```
/* NSS hard, slave, CPOL и CPHA на ноль (восходящий первый край) */
/* (1) Выберите TX и RX с DMA, включите прерывание RXNE, выберите 8-бит Rx fifo */
/* (2) Включить SPI2 */
SPI2->CR2 = SPI_CR2_TXDMAEN | SPI_CR2_RXDMAEN
| SPI_CR2_RXNEIE | SPI_CR2_FRXTH
| SPI_CR2_DS_2 | SPI_CR2_DS_1 | SPI_CR2_DS_0; /* (1) */
SPI2->CR1 |= SPI_CR1_SPE; /* (2) */
```

ИЗМЕНЕНИЯ:

Исправлены и обновлены процедуры отключения SPI и обработки DMA.

Исправлено управление CRC DMA.

A.19.10 Пример кода режима IrDA в USART /

```
/* (1) Перевыборка на 16, 9600 бод */
/* (2) Разделить на 24 для достижения низкой частоты мощности */
```

```

/* (3) Включить IrDA */
/* (4) 8 бит данных, 1 стартовый бит, 1 стоповый бит, без четности */
USART1->BRR = 480000 / 96; /* (1) */
USART1->GTPR = 24; /* (2) */
USART1->CR3 = USART_CR3_IREN; /* (3) */
USART1->CR1 = USART_CR1_TE | USART_CR1_UE; /* (4) */
/* Опрос простоя кадра Передача */
while((USART1->ISR & USART_ISR_TC) != USART_ISR_TC)
{
/* добавьте время для надежного приложения */
}
USART1->ICR |= USART_ICR_TCCF; /* Clear TC flag */
USART1->CR1 |= USART_CR1_TCIE; /* Enable TC interrupt */

```

А.19.11 Пример кода USART DMA

```

/* (1) Перевыборка на 16, 9600 бод */
/* (2) Включить DMA при приеме и передаче */
/* (3) 8 бит данных, 1 стартовый бит, 1 стоповый бит, без контроля четности, приема и
передачи */
USART1->BRR = 480000 / 96; /* (1) */
USART1->CR3 = USART_CR3_DMAT | USART_CR3_DMAR; /* (2) */
USART1->CR1 = USART_CR1_TE | USART_CR1_RE | USART_CR1_UE; /* (3) */
/* Опрос простоя кадра Передача */
while ((USART1->ISR & USART_ISR_TC) != USART_ISR_TC)
{
/* добавьте время для надежного приложения */
}
USART1->ICR |= USART_ICR_TCCF; /* Clear TC flag */
USART1->CR1 |= USART_CR1_TCIE; /* Enable TC interrupt */

```