

Прямой доступ к памяти

Копирование данных из одной области памяти в другую — распространённая задача. Если их количество не велико, то проблемы никакой нет, но, если нам нужно перемещать целые массивы или очень быстро забирать данные с какого-нибудь регистра, операция копирования начинает сжирать заметную часть процессорного времени.

В большинстве современных микроконтроллеров, в том числе и в **stm32f103c8**, специально для таких случаев предусмотрен модуль прямого доступа к памяти, ПДП (англ. Direct Memory Access, DMA). Обобщая, в нашем микроконтроллере всего два вида памяти: регистры периферии и основная память. Соответственно ПДП может:

- либо копировать из памяти в периферию (M2P) / из периферии в память (P2M);
- либо из памяти в память (M2M).

Операция копирования из периферии в периферию не доступна.

Запустить копирование можно как программно (записью в определённый регистр), так и привязать запуск DMA к какому-нибудь событию, например к окончанию преобразования ADC или приёму символа по UART.

Задумывались как работает mp3-плеер? В его состав входит flash-память, где хранятся песни. Как минимум их нужно вытащить от туда в оперативную память для последующей декодировки. Далее результат декодировки нужно подать на цифро-аналоговый преобразователь, а это опять операция копирования. Если декодировщик программный, то тратить драгоценное процессорное время на тупое копирование -- непозволительная роскошь.

Допустим по UART приходят команды, которые нужно обрабатывать. Обычно команды — это последовательности символов (NMEA, AT-команды и т.д.), поэтому пытаться обработать один единственный прибывший символ не имеет смысла, их нужно сложить в массив, а затем проанализировать. DMA можно настроить так, что бы с каждым новым его вызовом он итерировал адрес в памяти на нужную длину: 1 (`uint8_t`); 2 (`uint16_t`); или 4 байта (`uint32_t`). Так как массив — линейный участок в памяти, все данные будут складываться друг за другом.

Буфер однако не может быть бесконечным, поэтому DMA можно настроить на работу в циклическом режиме. Когда индекс дойдёт до края массива, он просто запишет следующее значение в начало. Такая возможность удобна ещё и тогда, когда вы работаете с несколькими каналами ADC, которые работают по очереди и складывают результаты преобразования в один единственный регистр.

Как и любая другая периферия DMA может формировать запросы на прерывания.

- **Передача закончена на половину** (англ. Half Transfer). Вернёмся к mp3-плееру — если в выходном буфере (данные на ЦАП) нет места, то зачем считывать новую партию данных с flash-памяти?
- **Окончание передачи** (англ. Transfer Complete). Модуль DMA может работать, когда ядро находится в режиме сна. Один из таймеров можно привязать к ADC, который будет запускаться периодически. К ADC привязать DMA, который будет копировать данные результата в массив. Как только буфер заполнится, сработает прерывание от DMA и микроконтроллер проснётся, что бы обработать их.
- **Ошибка передачи** (англ. Transfer Error). Если запись или чтение было произведено из непредусмотренных областей памяти (зарезервированных), то контроллер прямого доступа к памяти

выдаст ошибку.

Конкретно в `stm32f103` присутствует два модуля DMA ¹, которые суммарно предоставляют 12 каналов (7 на DMA1 и 5 на DMA2). Все они могут быть настроены независимо друг от друга, но есть небольшое ограничение: нельзя повесить нужную периферию на произвольный канал.

Peripherals	Channel 1	Channel 2	Channel 3	Channel 4	Channel 5	Channel 6	Channel 7
ADC1	ADC1	-	-	-	-	-	-
SPI/I ² S	-	SPI1_RX	SPI1_TX	SPI2/I2S2_RX	SPI2/I2S2_TX	-	-
USART	-	USART3_TX	USART3_RX	USART1_TX	USART1_RX	USART2_RX	USART2_TX
I ² C	-	-	-	I2C2_TX	I2C2_RX	I2C1_TX	I2C1_RX
TIM1	-	TIM1_CH1	-	TIM1_CH4 TIM1_TRIG TIM1_COM	TIM1_UP	TIM1_CH3	-
TIM2	TIM2_CH3	TIM2_UP	-	-	TIM2_CH1	-	TIM2_CH2 TIM2_CH4
TIM3	-	TIM3_CH3	TIM3_CH4 TIM3_UP	-	-	TIM3_CH1 TIM3_TRIG	-
TIM4	TIM4_CH1	-	-	TIM4_CH2	TIM4_CH3	-	TIM4_UP

Таблица с каналами DMA1 из Reference Manual, стр. 282

В более продвинутых МК от ST наряду с каналами присутствуют потоки (англ. stream), но мы не будем их рассматривать.

Шины и DMA

Все данные передаются по шинам (англ. bus), которые пересекаются в матрице (англ. bus matrix). В STM32 модуль DMA и ядро Cortex-M3 используют одну и ту же шину, поэтому работа DMA может тормозить работу ядра. С одной стороны, это существенный недостаток, а с другой такой подход позволяет делать микроконтроллеры дешёвыми.

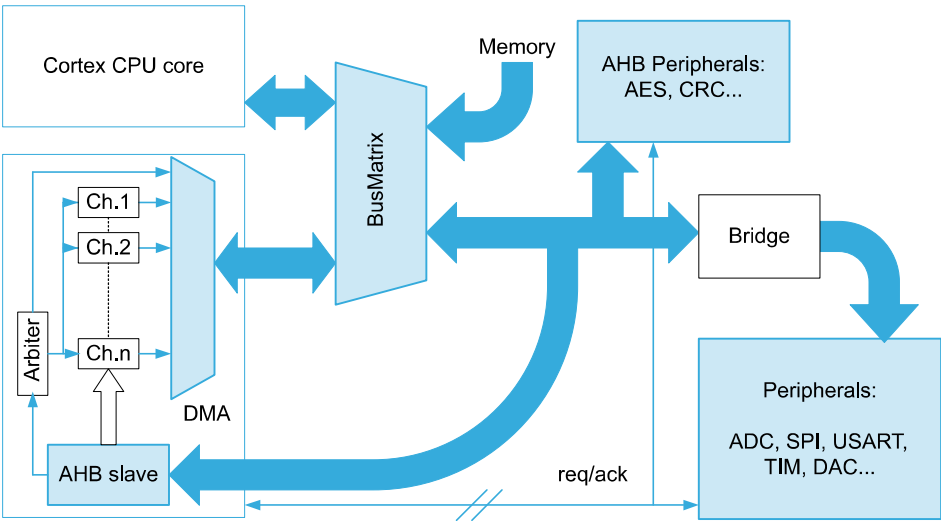


Диаграмма шин данных, документ [AN2548](#), стр. 7.

Когда ядро считывает инструкции из flash-памяти, а DMA работает с периферией, они не мешают друг другу.

Матрица шин реализует алгоритм **round-robin** для распределения нагрузки. Проще говоря шина циклически перебирает всех потребителей и даёт им доступ по очереди.

Здесь сразу же может возникнуть вопрос, — «А что если несколько каналов DMA захотят работать одновременно?» Ответ довольно прост: они не смогут этого сделать. Каждому «объекту» (периферии) программно задаётся приоритет (очень высокий, высокий, средний и низкий), ориентируясь на который специальная схема-арбитр (англ. arbiter) позволит работать тому или иному каналу. В случае, если программные приоритеты равны у разных каналов, арбитр смотрит на их номера. Чем меньше номер, тем выше приоритет. Если одновременно нужно забрать данные с **ADC1** и считать символ с **SPI2**, то последний будет ожидать освобождения шины первым каналом.

Если в МК присутствует два модуля, то **DMA1** имеет приоритет над **DMA2**.

После возникновения события, периферия отправляет сигнал контроллеру ПДП. Контроллер, ориентируясь на приоритетность выбирает самый срочный запрос и отправляет сигнал одобрения (англ. acknowledge), в ответ периферия сбрасывает запрос и контроллер забирает или отсылает данные.

Регистры DMA

Для работы с прерываниями отведено два регистра. Устроены они аналогично, однако первый, **ISR**, служит для индикации, т.е. его можно только считывать, а второй, **IFCR** для сброса.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved				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

- **TEIFx** сокращение от Transfer Error, т.е. отвечает за событие ошибки передачи.
- **HTIFx** сокращение от Half Transfer, т.е. отвечает за событие половинной передачи.
- **TCIFx** сокращение от Transfer Complete, т.е. отвечает за событие окончания передачи.
- **GIFx** сокращение от Global, т.е. сообщает, что одно из трёх событий в принципе произошло.

Следующая группа регистров (**CCRx**) позволяет настроить режим работы канала.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
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

- **MEM2MEM** — включение (**1**) или отключение (**0**) режима Память-Память.
- **PL** — установка приоритета канала: низкий (**00**); средний (**01**); высокий (**10**); или очень высокий (**11**).
- **MSIZE** — размер данных в памяти: 8 бит (**00**); 16 бит (**01**); или 32 бит (**10**).
- **PSIZE** — размер данных в периферии: 8 бит (**00**); 16 бит (**01**); или 32 бит (**10**).
- **MINC** — разрешает (**1**) или запрещает (**0**) инкрементировать адрес памяти.
- **PINC** — разрешает (**1**) или запрещает (**0**) инкрементировать адрес периферии.
- **CIRC** — разрешает (**1**) или запрещает (**0**) циклический режим работы.
- **DIR** — задаёт направление передачи: **0** чтение из периферии; **1** чтение из памяти.

- **TEIE** — разрешает (1) или запрещает (0) прерывание ошибки передачи.
- **HTIE** — разрешает (1) или запрещает (0) прерывание по передаче половины.
- **TCIE** — разрешает (1) или запрещает (0) прерывание по окончанию передачи.
- **EN** — включает (1) или отключает (0) канал.

Группа регистров **CNDTRx** задаёт количество передаваемых данных для каждого канала, которое должна находиться в диапазоне от 0 до 65535.

И последние два регистра **CPARx** и **CMARx** задают начальный адрес (32 бита) периферии и памяти соответственно.

[Назад](#) | [Оглавление](#) | [Дальше](#)

1. Два DMA доступны только в high-density МК, а **stm32f103c8** относится к medium-density. [↪](#)