

STM32 DMAMUX: маршрутизатор запросов DMA

Введение

Чтобы снять определенные обязанности по передаче данных с ЦП, микроконтроллеры STM32 (MCU) и микропроцессоры (MPU) встраивают контроллеры прямого доступа к памяти (DMA). DMA может выполнять блочную передачу данных по запросу периферийного устройства или программному триггеру.

Каждый канал прямого доступа к памяти имеет конфигурируемый программным обеспечением выбор периферийного устройства, запрашивающего его услуги. В устаревших продуктах STM32 выбор запроса канала реализован внутри контроллера DMA с ограниченным списком периферийных запросов для данного канала. Программное приложение не может свободно сопоставлять любой запрос периферийного устройства с любым каналом.

Возможности маршрутизации запросов DMA STM32 расширены за счет мультиплексора запросов DMA (периферийное устройство DMAMUX). DMAMUX добавляет больше гибкости, предоставляя полное динамическое сопоставление периферийных запросов DMA вместо псеводинамического сопоставления. Он предлагает полностью настраиваемую маршрутизацию любого запроса DMA от данного периферийного устройства к любому контроллеру DMA и/или каналу DMA контроллера.

В этом примечании по применению объясняются различные функции DMAMUX продуктов, перечисленных в таблице ниже: как настроить DMAMUX, а также даются рекомендации по использованию новых возможностей синхронизации и генерации запросов. Для получения дополнительной информации о DMAMUX в устройствах STM32 обратитесь к справочным руководствам по продуктам, доступным на сайте www.st.com.

Таблица 1. Применимые продукты

Тип	Серия продукта
Microcontrollers	STM32H7 Series
	STM32G0 Series
	STM32G4 Series
	STM32L4+ Series
	STM32L5 Series
	STM32WB Series
Microprocessors	STM32MP1 Series

1 Описание DMAMUX

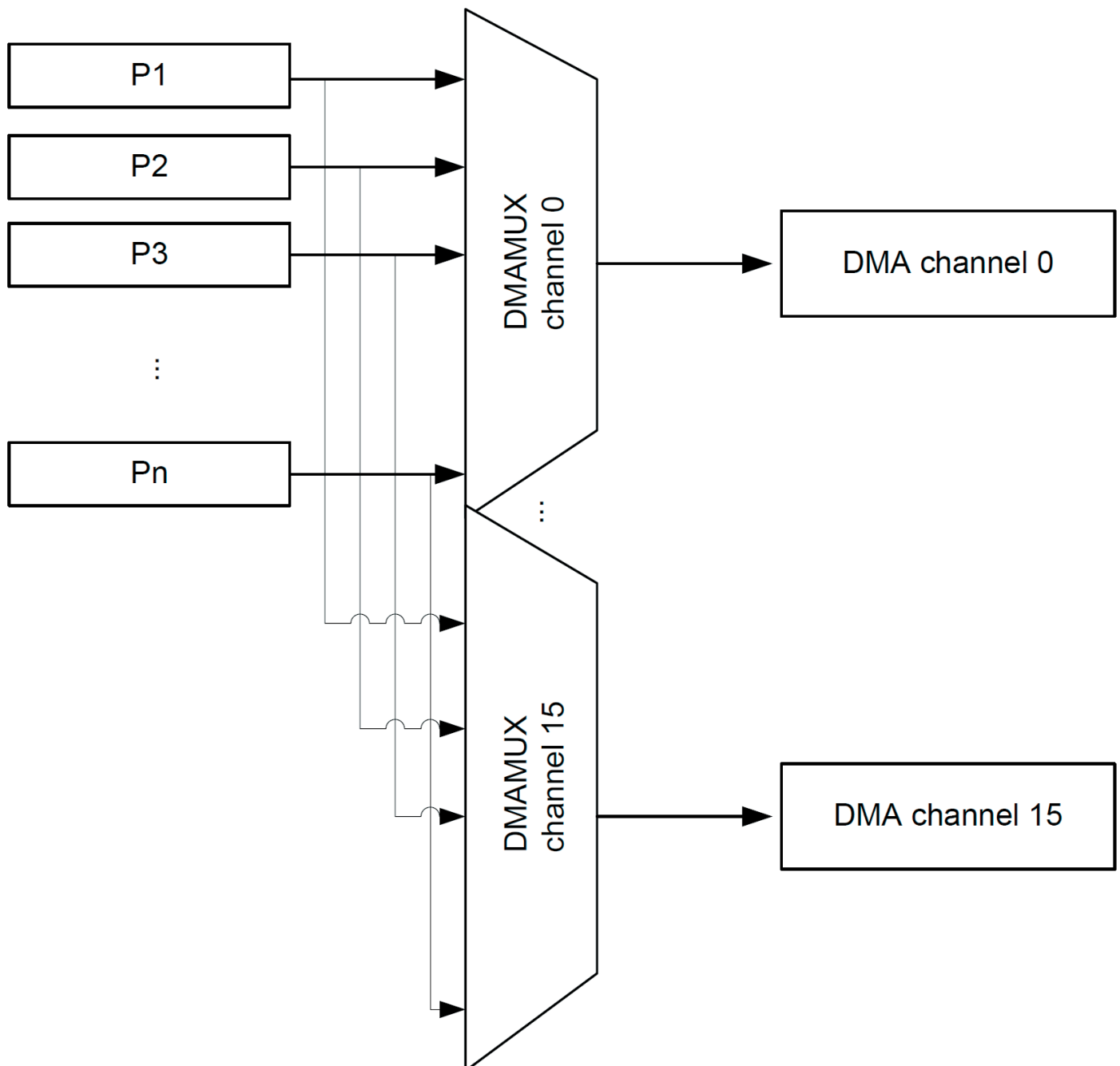
Периферийное устройство указывает запрос на передачу DMA, устанавливая свой сигнал запроса DMA. Запрос DMA находится в состоянии ожидания до тех пор, пока он не будет обслужен контроллером DMA, который генерирует сигнал подтверждения DMA, и соответствующий сигнал запроса DMA не будет деактивирован.

В этом документе набор управляющих сигналов, необходимых для протокола запроса/подтверждения прямого доступа к памяти, явно не описывается, и он упоминается как периферийная линия запроса прямого доступа к памяти.

Маршрутизатор запросов DMA можно рассматривать как расширение контроллера DMA. Он направляет периферийные запросы DMA самому контроллеру DMA.

Мультиплексор запросов DMAMUX позволяет маршрутизировать линию запроса DMA от периферийных устройств к контроллерам DMA продукта. Функция маршрутизации обеспечивается программируемым многоканальным мультиплексором линии запроса DMA. Каждый канал (канал 0 DMAMUX в примере на рисунке ниже) выбирает уникальную строку запроса DMA для пересылки (безоговорочно или синхронно) на соответствующий канал контроллера DMA (канал 0 DMA в примере на рисунке ниже). Это позволяет управлять запросами DMA с высокой гибкостью, максимально увеличивая количество запросов DMA, которые выполняются одновременно.

Рисунок 1. Мультиплексор запросов DMAMUX

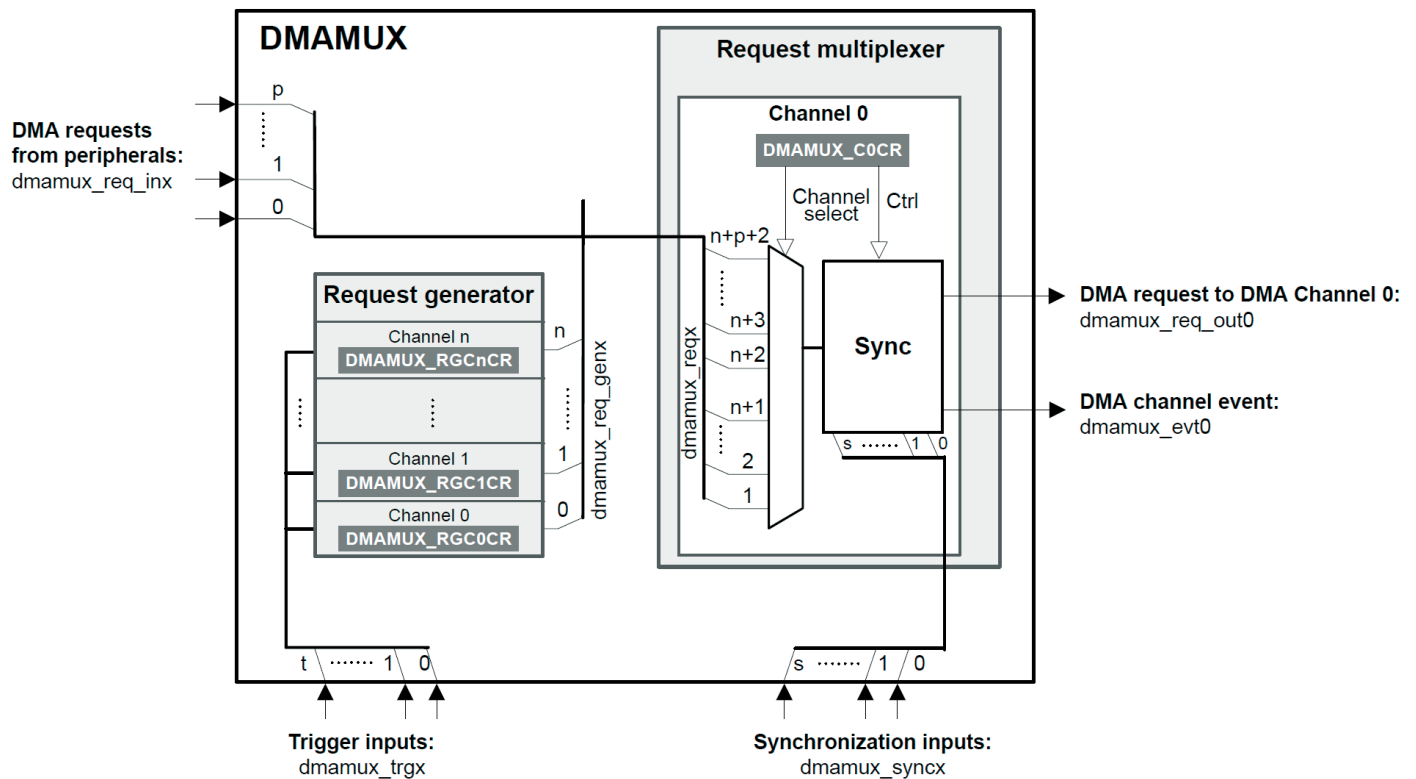


Где, Px - периферийный запрос x (пример LPUART1_TX или LPUART1_RX)

2 функции DMAMUX

На рисунке ниже представлена упрощенная блок-схема DMAMUX. Структура «Мультиплексор запроса» дублируется N раз в зависимости от количества каналов DMA, управляемых DMAMUX.

Рисунок 2. Упрощенная блок-схема DMAMUX



Примечание. Упрощенная блок-схема только с одним мультиплексором запросов.

DMAMUX в основном состоит из двух компонентов: мультиплексора запросов (или блока маршрутизатора) и генератора запросов.

Мультиплексор запросов включает в себя блок синхронизации на канал со следующими входами/выходами:

- Входы:

- $dmamux_reqx$: запрос DMA от периферийного устройства ($dmamux_req_inx$) или от генератора запросов ($dmamux_req_genx$)

$dmamux_req_gen[0..n]$ затрагивается соответственно $dmamux_req[1..n+1]$ и $dmamux_req_inx$ изменяется, начиная с $dmamux_req[n+2]$.

- $dmamux_syncx$: необязательное событие синхронизации

- Выходы:

- $dmamux_req_outx$: запрос DMA $dmamux_reqx$ перенаправляется с входа на выход

- $dmamux_evtx$: необязательное сгенерированное событие, которое можно использовать для запуска/синхронизации других каналов DMAMUX.

Генератор запросов позволяет генерировать запросы DMA по сигналам прерывания или событиям с вводом/выводом следующим образом:

- Вход: $dmamux_trgx$, триггерные входы событий в подблок генератора запросов.

- Вывод: `dmamux_req_genx`, запрос DMA от подблока генератора запросов к каналам мультиплексора запросов DMAMUX.

Количество блоков мультиплексора запросов зависит от количества каналов DMA, управляемых DMAMUX.

Например:

- Для 8-канального прямого доступа к памяти должны быть доступны 8 каналов мультиплексора запроса.
- Для продукта с двумя контроллерами прямого доступа к памяти по 8 каналов в каждом должно быть доступно 16 каналов мультиплексора запросов.

Генератор запросов создается один раз DMAMUX. Он содержит N каналов (в зависимости от продукта), способных генерировать запросы DMA. Дополнительные сведения см. в разделе «Реализация DMAMUX» в справочном руководстве по продукту.

Благодаря блоку генератора запросов пользовательское программное обеспечение может запускать передачи DMA на основе сигналов от периферийных устройств, которые не реализуют запросы DMA.

2.1 Маршрутизация запросов и синхронизация

2.1.1 Безусловная переадресация запроса

Для выполнения передачи из периферийного устройства в память или из памяти в периферийное устройство каналу контроллера прямого доступа к памяти каждый раз требуется строка запроса периферийного прямого доступа к памяти. Каждый раз, когда происходит запрос, канал DMA передает данные с/на периферийное устройство. Канал x мультиплексора запросов DMAMUX позволяет выбирать/маршрутизировать линию запроса периферийного DMA на канал DMA x.

Когда установлен мультиплекс (DMAREQ_ID не равен нулю), он обеспечивает фактическую маршрутизацию линии запроса DMA. Подключение периферийного запроса DMA к выходу канала мультиплексора выбирается через запрограммированный ID в битах DMAREQ_ID регистра управления каналом (DMAMUX_CxCR).

Для каждой строки запроса периферийного DMA в продукте применяется уникальный идентификатор. Нулевое значение (DMAREQ_ID = 0x00) соответствует тому, что строка запроса DMA не выбрана.

После настройки канала DMAMUX можно в свою очередь настроить соответствующий канал контроллера DMA. Два разных канала DMAMUX не могут быть настроены для выбора одной и той же периферийной линии запроса DMA в качестве источника.

2.1.2 Переадресация условного запроса

В дополнение к безусловной переадресации запросов модуль синхронизации позволяет программному обеспечению реализовывать условную переадресацию запросов. Маршрутизация эффективно выполняется только при обнаружении определенного условия. Передачи DMA могут быть синхронизированы с внутренними или внешними сигналами.

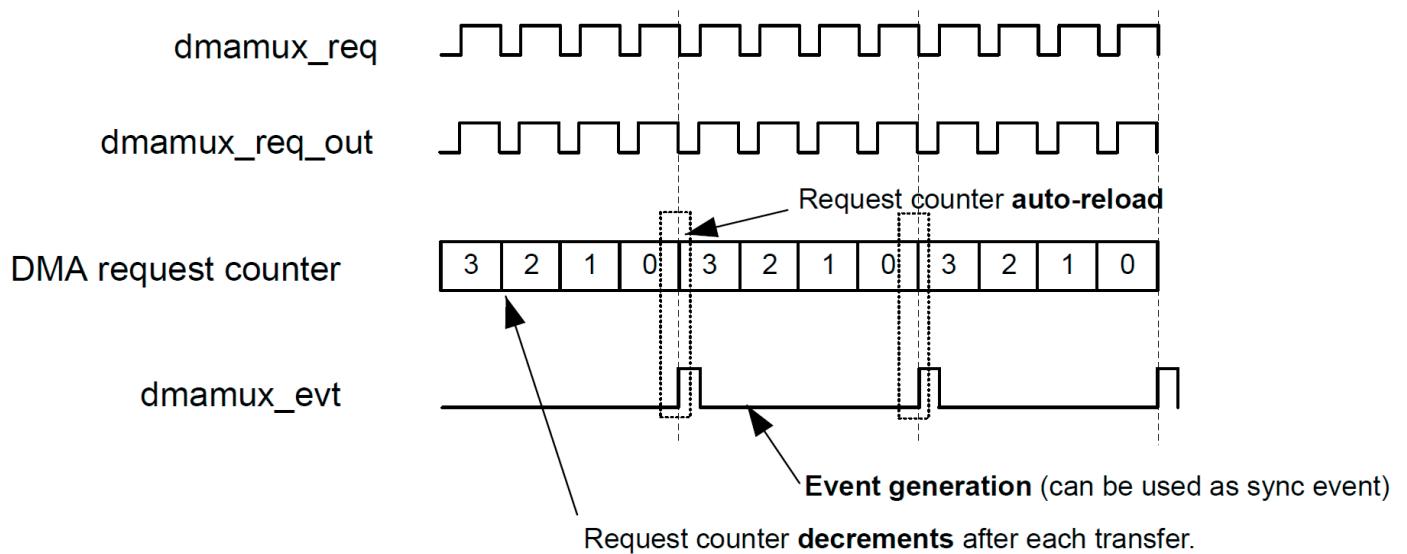
Например, пользовательское программное обеспечение может использовать блок синхронизации, чтобы инициировать или регулировать пропускную спо-

способность передачи данных. Запрос DMA может быть перенаправлен одним из следующих способов:

- каждый раз, когда на выводе GPIO (EXTI) обнаруживается фронт.
- в ответ на периодическое событие от таймера
- в ответ на асинхронное событие от периферии
- в ответ на событие от другого маршрутизатора запросов (цепочка запросов)

Помимо обработки запроса DMA, блок синхронизации позволяет генерировать события, которые могут использоваться другими подблоками DMAMUX (например, генератором запросов или другим каналом мультиплексора запросов DMAMUX).

Рис. 3. Канал мультиплексора линии запроса DMA — генерация события



Когда канал DMAMUX настроен в синхронном режиме, его поведение выглядит следующим образом:

1. Вход мультиплексора запросов (запрос DMA от периферийного устройства) может стать активным, но он не перенаправляется на выход мультиплексора запросов DMAMUX до тех пор, пока не будет получен сигнал синхронизации.

2. Когда событие синхронизации получено, мультиплексор запросов соединяет свой вход и выход, и ожидающий запрос периферийного устройства, если таковой имеется, перенаправляется.

3. Каждый пересылаемый запрос DMA уменьшает счетчик мультиплексора запросов (значение, запрограммированное пользователем). Когда счетчик достигает нуля и контроллер DMA подтверждает последний перенаправленный запрос, соединение между контроллером DMA и периферийным устройством отключается (не перенаправляется) в ожидании нового события синхронизации.

Для каждого опустошения счетчика линия мультиплексора запроса может генерировать необязательное событие для синхронизации со второй линией DMAMUX. Это же событие можно использовать в некоторых сценариях с низким энергопотреблением для переключения системы обратно в режим остановки без вмешательства ЦП.

Режим синхронизации можно использовать, например, для автоматической синхронизации передачи данных с таймером или для запуска передачи по событию периферийного устройства.

Сигнал синхронизации (SYNC_ID), полярность сигнала синхронизации (SPOL) и количество запросов на пересылку (NBREQ+1) настраиваются в регистре конфигурации канала мультиплексора линии запроса (DMAMUX_CxCR).

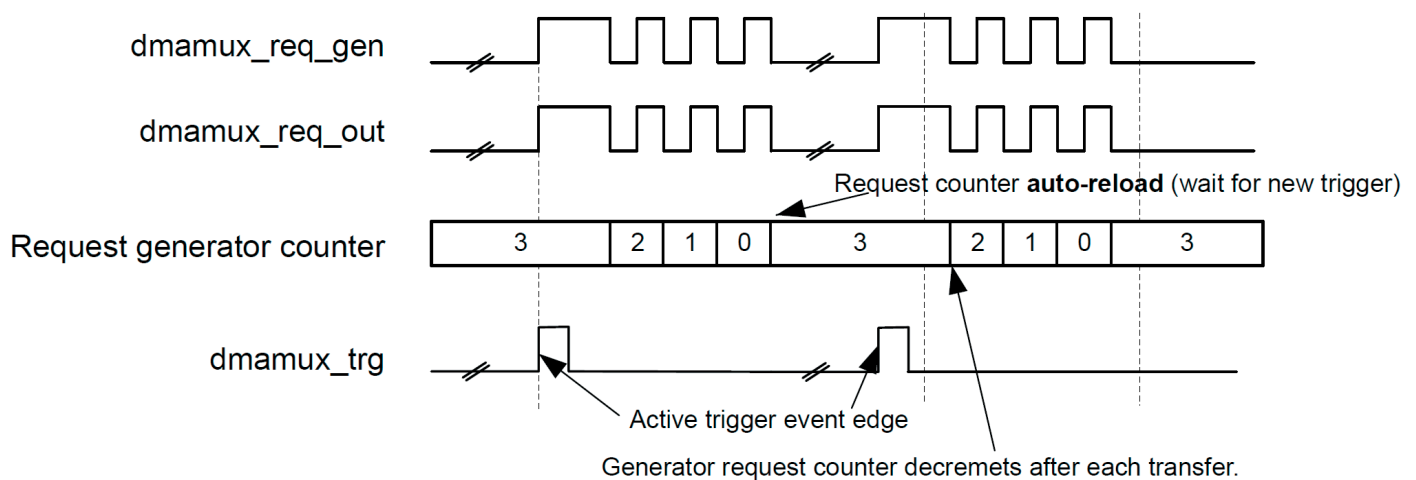
2.2 Генерация запроса

Генератор запросов можно рассматривать как посредника между периферийным устройством и контроллерами прямого доступа к памяти. Это позволяет периферийным устройствам без возможности прямого доступа к памяти (таким как сигнализация RTC или компараторы) генерировать программируемое количество запросов прямого доступа к памяти по событию. Сигнал триггера (SIG_ID), полярность триггера (GPOL) и количество запросов минус 1 для генерации (GNBREQ) настраиваются в регистре конфигурации генератора запросов (DMAMUX_RGxCR).

При получении триггерного события соответствующий канал генератора начинает генерировать запросы DMA на своем выходе. Каждый раз, когда сгенерированный DMAMUX запрос обслуживается подключенным контроллером DMA, встроенный счетчик запросов DMA (один счетчик на канал генератора запросов) уменьшается.

При недогрузке канал генератора запросов перестает генерировать запросы DMA, а счетчик запросов DMA автоматически перезагружается до своего запрограммированного значения при следующем событии триггера.

Рисунок 4. Генерация запроса DMA



Если новое событие триггера получено в то время, когда генератор управляет предыдущей инициированной последовательностью запроса DMA, то бит флага переполнения события триггера запроса OFx устанавливается аппаратно в регистре состояния DMAMUX_RGSR.

2.3 Генерация запроса и синхронизация

Для реализации сценариев автономной передачи и управления DMAMUX предлагает возможность объединения функции генерации запросов и синхронизации запросов в одной конфигурации.

3 Примеры DMAMUX

В этих примерах используется инструмент STM32CubeMX версии 4.26.1, работающий на микроконтроллерах STM32 (на базе ядер Arm®).

Примечание. Arm является зарегистрированным товарным знаком Arm Limited (или ее дочерних компаний) в США и/или других странах.

3.1 Пример использования синхронизации

После настройки канала DMA для обслуживания периферийной линии запроса DMA (пример SPI6_TX),

Блок синхронизации можно включить, как показано на рисунке ниже. В этом случае передний фронт сигнала LPTIM3_out используется для управления периодами передачи.

Рисунок 5. Пример использования синхронизации (на базе STM32CubeH7)

The screenshot shows the 'BDMA Configuration' dialog box. At the top, there are checkboxes for 'BDMA' and 'MemToMem', both of which are checked. Below this is a table with the following data:

DMA Request	Channel	Direction	Priority
SPI6_TX	BDMA Channel 0	Memory To Peripheral	Low
BDMA_GENERATOR0	BDMA Channel 1	Peripheral To Memory	Low

Below the table are 'Add' and 'Delete' buttons. The 'DMA Request Settings' section includes a 'Mode' dropdown set to 'Normal', an 'Increment Address' checkbox (unchecked), and 'Data Width' dropdowns for both 'Peripheral' and 'Memory' set to 'Half Word'. The 'DMA Request Generator Settings' section includes fields for 'Request generation Signal', 'Signal polarity', and 'Request number'. The 'DMA Request Synchronization Settings' section includes a checked 'Enable synchronization' checkbox, a 'Synchronization signal' dropdown set to 'LPTIM3 output', a 'Signal polarity' dropdown set to 'Synchronize on rising edge events', a checked 'Enable event' checkbox, and a 'Request number' field set to '1'. At the bottom are 'Apply', 'Ok', and 'Cancel' buttons.

Annotations on the right side of the dialog box:

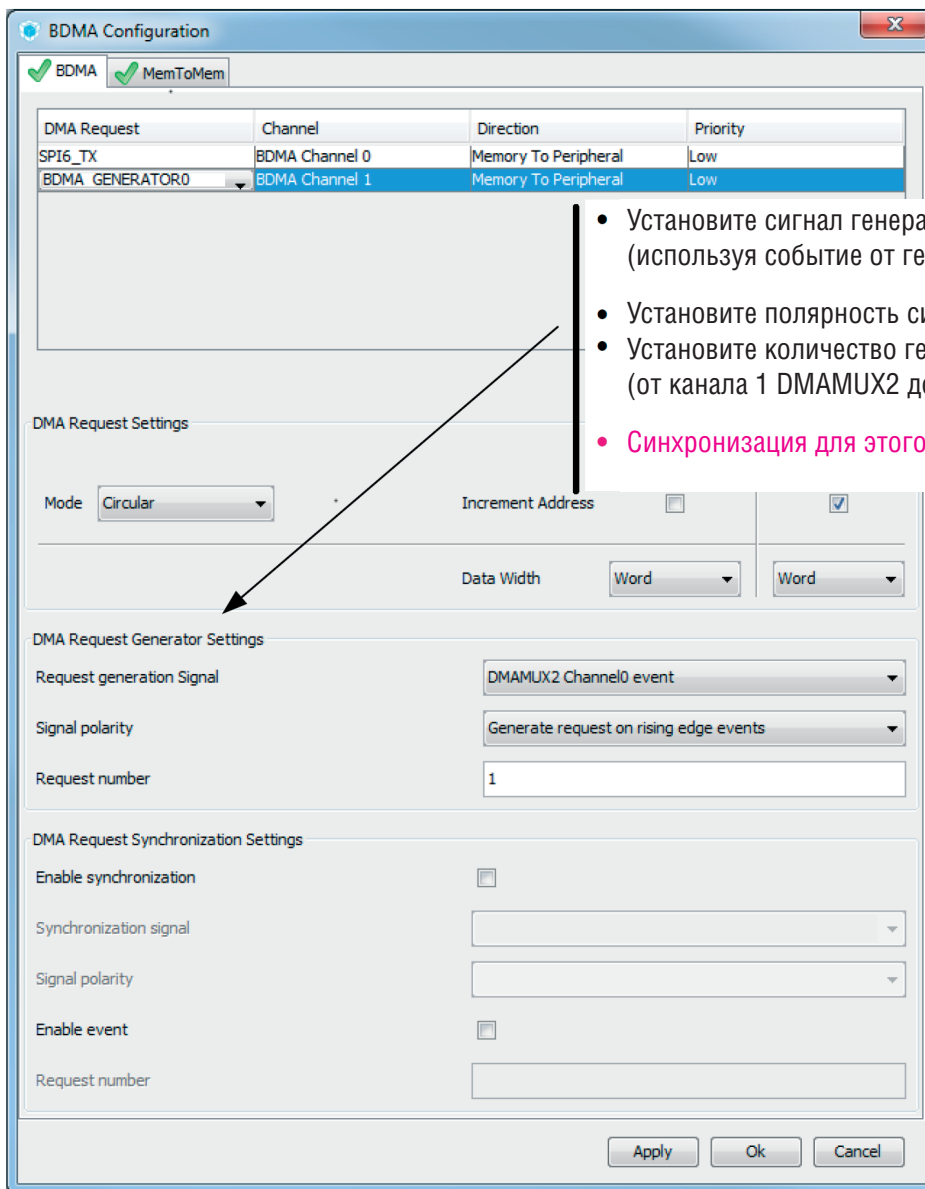
- Установите флажок Включить синхронизацию
- Установить сигнал синхронизации из списка (настраивается в приложении)
- Установить полярность сигнала синхронизации
- Установите флажок Включить событие (включить генерацию события)
- Установить количество запросов для пересылки

3.2 Пример использования подблока генератора запросов DMAMUX

Чтобы иметь некоторую автоматизацию, новые передачи DMA могут быть сгенерированы после передачи DMA на SPI6.

Благодаря генерации событий DMAMUX Channel 0 можно запустить генератор запросов. Сценарий можно настроить, как показано на рисунке ниже.

Рис. 6. Пример использования генератора запросов (на базе STM32CubeH7)



- Установите сигнал генерации запроса из списка (используя событие от генератора запросов DMAMUX2 0)
- Установите полярность сигнала (используя нарастающий фронт)
- Установите количество генерируемых запросов (от канала 1 DMAMUX2 до канала 1 BDMA)
- Синхронизация для этого канала не включена.

3.3 Примеры STM32CubeH7

Следующие примеры взяты из STM32CubeH7:

- DMAMUX_RequestGen

В этом примере линия EXTI0 используется для запуска генератора запросов DMAMUX и для выполнения передачи данных DMA из буфера SRAM в регистр выходных данных GPIO, изменяя состояние выходного вывода при каждом появлении нарастающего фронта EXTI0.

- DMAMUX_SYNC

В этом примере используется USART1 в режиме синхронизации DMA для отправки обратного отсчета от 10 до 00 с периодом 2 секунды. Блок синхронизации DMAMUX настроен на синхронизацию передачи DMA с выходным сигналом LPTIM1. Каждый нарастающий фронт сигнала синхронизации (выходной сигнал LPTIM1) разрешает передачу четырех запросов USART1 на периферийное устройство USART1 с использованием прямого доступа к памяти. Эти четыре запроса представляют собой два символа '\n\r' плюс два символа обратного отсчета от 10 до 00. LPTIM1 настроен на генерацию ШИМ с периодом 2 секунды.

Рисунок 7. DMAMUX_RequestGen

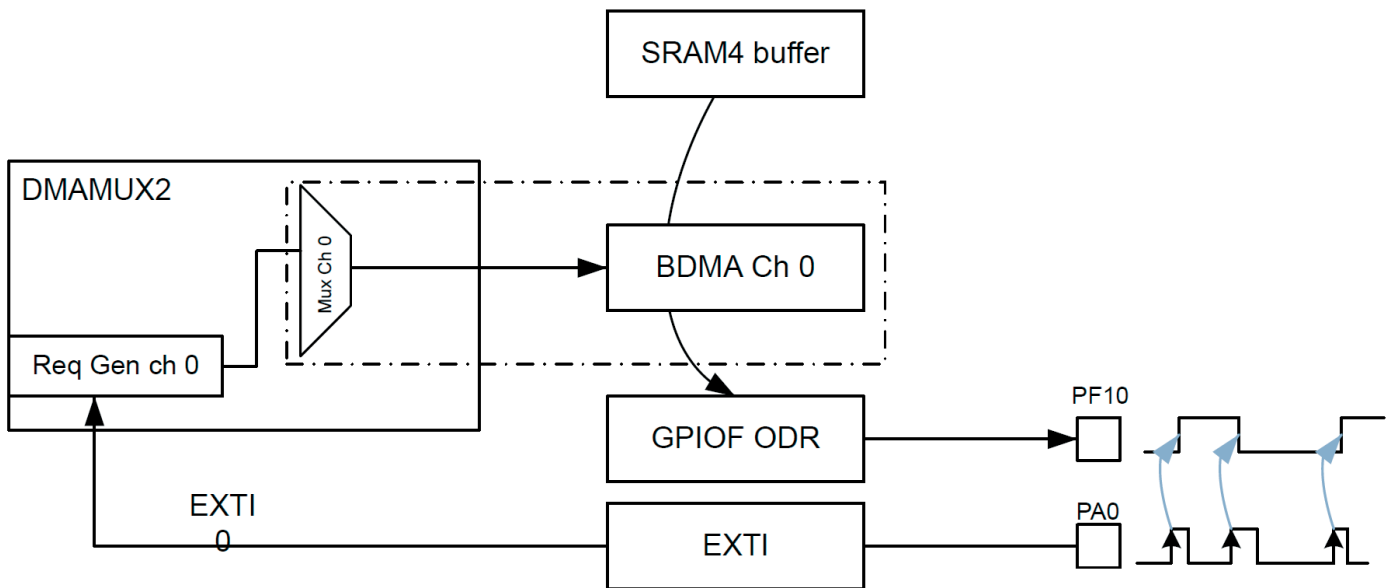
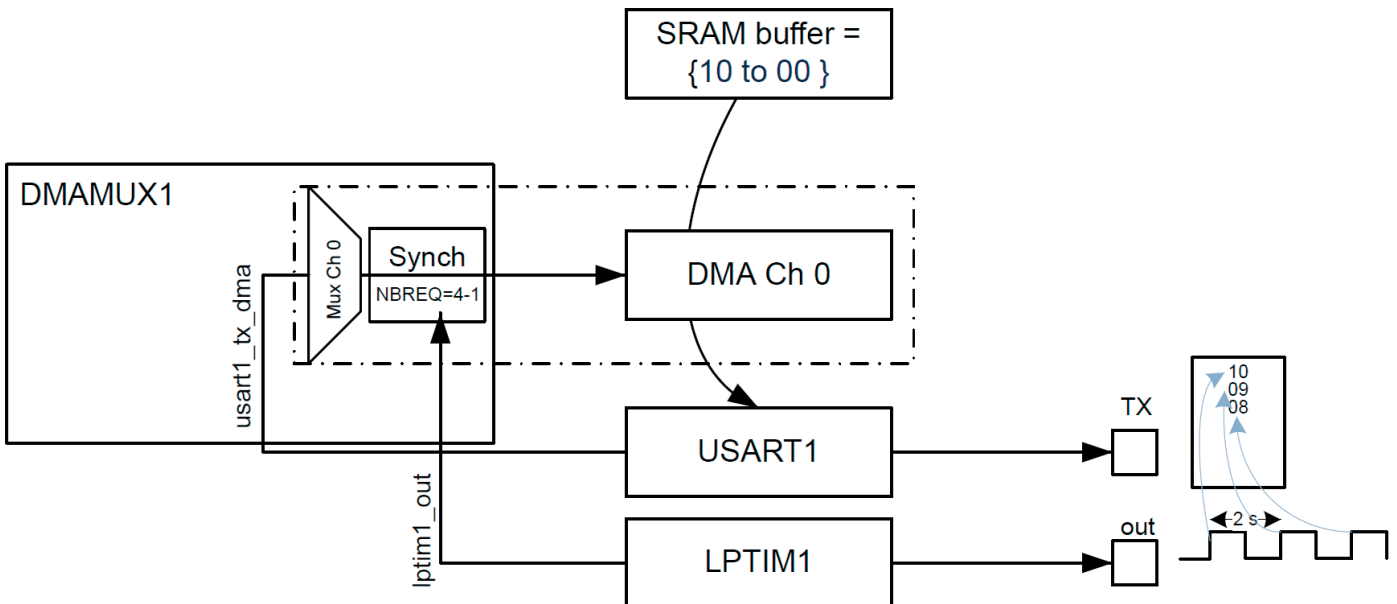


Рисунок 8. DMAMUX_SYNC



4. Вывод

Контроллер DMAMUX предназначен для упрощения распределения ресурсов встроенных приложений, поскольку он обеспечивает гибкость динамического выделения периферийного устройства для канала DMA. Кроме того, он расширяет возможности прямого доступа к памяти, предлагая механизм синхронизации, который позволяет разгрузить ЦП от управления передачей и синхронизации. Также комбинация синхронизации и генерации запросов может быть использована для реализации оптимизированной по энергопотреблению передачи данных (в автономном режиме без участия ЦП).

Лист регистраций изменений

Таблица 2. История изменений документа

Дата	Изменения версии
16 октября 2018 г.	1 Первоначальный выпуск.
20 ноября 2018 г.	2 Обновленная таблица 1. Применимые продукты.
16 января 2019 г.	3 Обновлен раздел 3. Примеры DMAMUX.
8 июня 2020 г.	4 Обновлено введение с новыми продуктами серий STM32G4, STM32L5 и STM32MP1