

STM32 DMAMUX: DMA 请求路由器

引言

为了从 CPU 减荷某些数据传输任务, STM32 微控制器嵌入了直接存储器访问 (DMA) 控制器。DMA 可以在外设请求或软件触发时执行面向块的数据传输。

每个 DMA 通道都有一个软件可配置的外设选择, 用于请求其服务。在传统 STM32 产品上, 通道请求选择在 DMA 控制器内实现, 该控制器带有面向给定通道的受限外设请求列表。软件应用程序不能自由地将任何外设请求映射到任何通道。

DMA 请求复用器 (DMAMUX 外设) 增强了 STM32 DMA 请求路由功能。DMAMUX 增加了更多灵活性, 可以提供完全动态的 DMA 外设请求映射, 而不是伪动态映射。它提供从给定外设到任何 DMA 控制器和/或控制器 DMA 通道的任何 DMA 请求完全可配置路由。

本应用说明解释了下表中所列产品的各种 DMAMUX 特性: 如何配置 DMAMUX, 以及如何使用新的同步和请求生成功能。

如需关于 STM32 设备中 DMAMUX 的详细信息, 请参阅 www.st.com 上提供的产品参考手册。

表 1. 适用产品

类型	产品系列
STM32 微控制器	STM32H7 系列
	STM32G0 系列
	STM32L4+ 系列
	STM32WB 系列

1 DMAMUX 描述

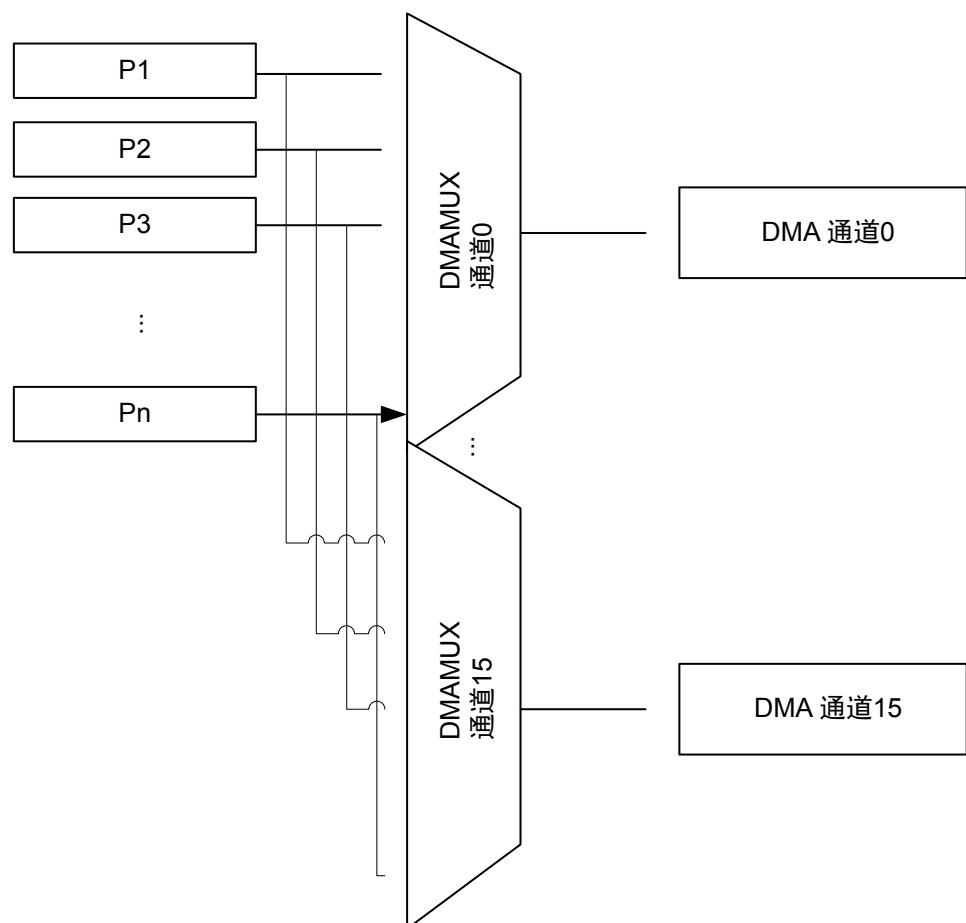
外设通过设置其 DMA 请求信号来指示 DMA 传输请求。DMA 控制器会处理 DMA 请求并生成 DMA 确认信号，而且相应的 DMA 请求信号也将变为无效，但在此之前 DMA 请求一直处于挂起状态。

本文档中并未明确描述 DMA 请求/确认协议所需的控制信号组，而是将其称作外设 DMA 请求线。

DMA 请求路由器可以看作是 DMA 控制器的扩展。它将 DMA 外设请求路由到 DMA 控制器本身。

DMAMUX 请求复用器可将外设的 DMA 请求线路由到产品 DMA 控制器。该路由功能通过可编程的多通道 DMA 请求线复用器来确保实现。每个通道（下图示例中的 DMAMUX 通道 0）选择一个唯一的 DMA 请求线转发（无条件或同步地）到关联的 DMA 控制器通道（下图示例中的 DMA 通道 0）。这就允许高度灵活地管理 DMA 请求，使并发运行的 DMA 请求数量最大化。

图 1. DMAMUX 请求复用器

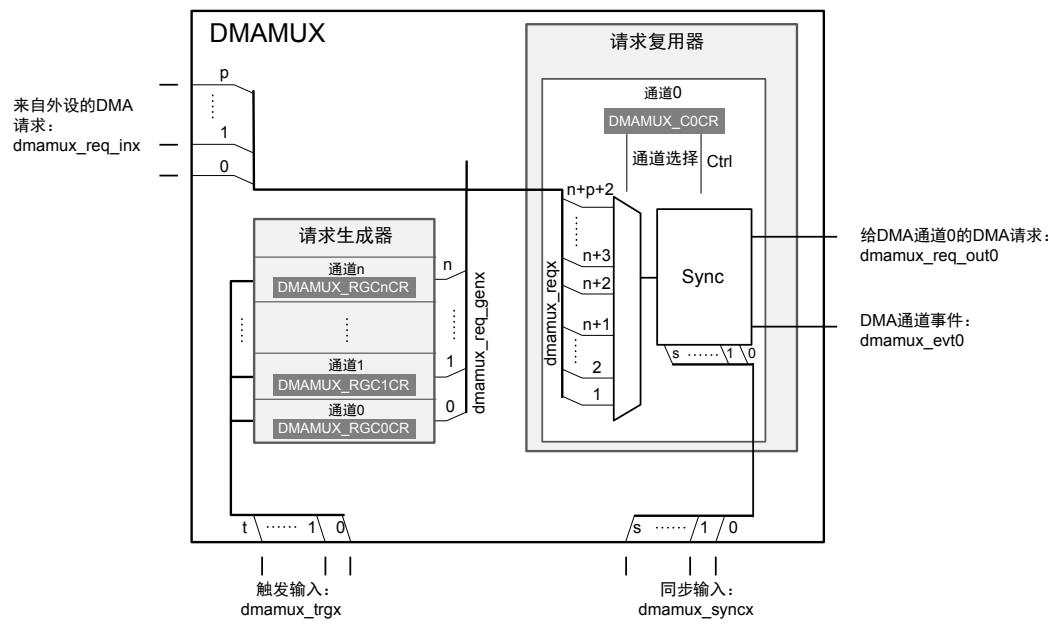


2

DMAMUX 功能

下图是一个简化的 DMAMUX 框图。根据 DMAMUX 管理的 DMA 通道数量，“请求复用器”结构被重复 N 次。

图 2. DMAMUX 简化框图



注: 简化框图仅有一个请求复用器

DMAMUX 主要由两个组件组成, 请求复用器 (或路由器块) 和请求生成器。

请求复用器包括每个通道的同步单元, 其输入/输出如下:

- 输入:

- `dmamux_reqx`: 来自外设的 DMA 请求 (`dmamux_req_inx`) 或来自请求生成器 (`dmamux_req_genx`) 的 DMA 请求
`dmamux_req_gen[0..n]` 分别受 `dmamux_req[1..n+1]` 的影响; 而 `dmamux_req_inx` 受 `dmamux_req[n+2]` 的影响。
- `dmamux_syncx`: 可选同步事件
- 输出:
 - `dmamux_req_outx`: DMA 请求 `dmamux_reqx` 从输入转发到输出
 - `dmamux_evtx`: 可选的生成事件, 可用于触发/同步其他 DMAMUX 通道

请求生成器允许根据中断信号或事件生成 DMA 请求, 输入/输出如下:

- 输入: `dmamux_trgx`, 请求生成器子块的触发事件输入
- 输出: `dmamux_req_genx`, 从请求生成器子块到 DMAMUX 请求复用器通道的 DMA 请求

请求复用器块的数量取决于 DMAMUX 管理的 DMA 通道数量。

例如:

- 如果是 8 通道 DMA, 必须有 8 个请求复用器通道。
- 如果某产品有两个 DMA 控制器 (每个 DMA 控制器各有 8 个通道), 则必须有 16 个请求复用器通道。

一旦请求生成器由 DMAMUX 实例化。它包含 N 个能够生成 DMA 请求的通道 (取决于产品)。有关详细信息, 请参阅产品参考手册中的“DMAMUX 实现”部分。

因为有请求生成器块, 用户软件可以基于未实现 DMA 请求的外设发出的信号触发 DMA 传输。

2.1 请求路由与同步

2.1.1 无条件请求转发

为了执行外设到内存或内存到外设的传输, DMA 控制器通道每次都需要一个外设 DMA 请求线。每次发生请求时, DMA 通道将数据传入/传出外设。DMAMUX 请求复用器通道 X 允许选择/路由外设 DMA 请求线到 DMA 通道 X。

当复用器设置之后 (DMAREQ_ID 不等于零), 它确保 DMA 请求线的实际路由。通过通道控制寄存器 (DMAMUX_CxCR) 的 DMAREQ_ID 位中的编码 ID 选择外设 DMA 请求到复用器通道输出的连接。

对于产品中的每个外设 DMA 请求线, 都会影响唯一的 ID。值 0 (DMAREQ_ID = 0x00) 对应于没有选择 DMA 请求线。

在配置 DMAMUX 通道之后, 可以依次配置相应的 DMA 控制器通道。不能将两个不同的 DMAMUX 通道配置为选择相同的外设 DMA 请求线作为源。

2.1.2 有条件请求转发

除了无条件请求转发, 同步单元还允许软件实施有条件请求转发。只有在检测到定义的条件时, 路由才能有效完成。DMA 传输可以与内部或外部信号同步。

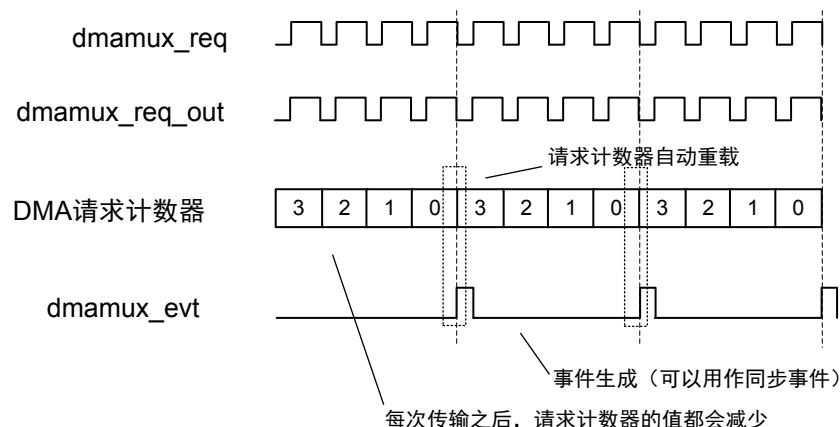
例如, 用户软件可以使用同步单元启动或调整数据传输吞吐量。

可以通过以下方式之一转发 DMA 请求:

- 每次在 GPIO 引脚 (EXTI) 上检测到沿时
- 响应来自计时器的周期性事件
- 响应来自外设的异步事件
- 响应来自另一个请求路由器 (请求链) 的事件

根据 DMA 请求条件, 同步单元允许生成其他 DMAMUX 子块 (如请求生成器或另一个 DMAMUX 请求复用器通道) 可能使用的事件。

图 3. DMA 请求线复用器通道 - 事件生成



DMAMUX 通道在同步模式下配置时, 其行为如下:

1. 请求复用器输入 (来自外设的 DMA 请求) 可以变为活动的, 但在接收到同步信号之前不会在 DMAMUX 请求复用器输出上转发。

2. 当接收到同步事件时，请求复用器连接其输入和输出，并转发等待的外设请求（如果有）。
3. 每个转发的 DMA 请求都会减少请求复用器计数器的数值（用户设定的值）。当计数器达到 0 并且最后一个转发的请求被 DMA 控制器确认后，DMA 控制器和外设之间的连接被禁用（未转发），等待一个新的同步事件。

每次计数器出现下溢时，请求复用器线可以生成一个可选事件与另一个 DMAMUX 线同步。一些低功耗场景中可以使用相同的事件将系统切换回停止模式，而不需要任何 CPU 干预。

例如，可使用同步模式自动将数据传输与定时器同步，或根据外设事件触发传输。

同步信号（SYNC_ID）、同步信号极性（SPOL）和转发请求数（NBREQ+1）在请求线复用器通道配置寄存器（DMAMUX_CxCR）中配置。

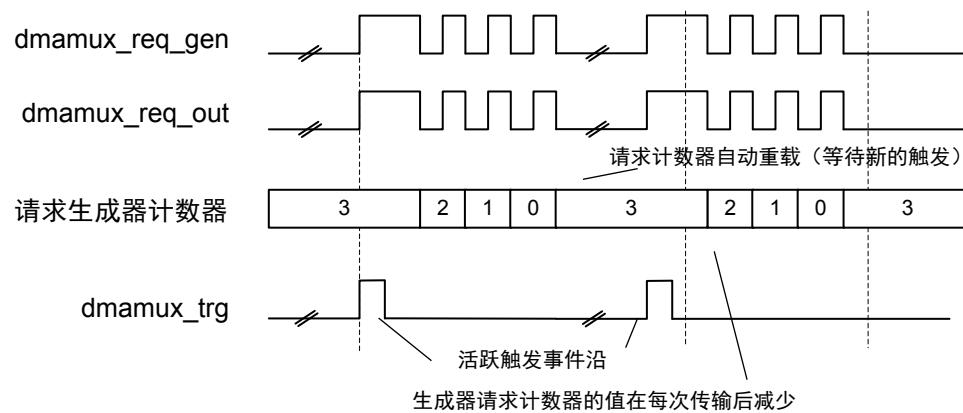
2.2 请求生成

请求生成器可以看作是外设和 DMA 控制器之间的中介。它允许没有 DMA 功能的外设（如 RTC 警报或比较器）根据事件生成一定数量（可编程）的 DMA 请求。在请求生成器配置寄存器（DMAMUX_RGxCR）中配置触发信号（SIG_ID）、触发极性（GPOL）和要生成的请求数（GNBREQ）-1。

收到触发事件后，相应生成器通道开始在其输出上生成 DMA 请求。每当 DMAMUX 生成的请求由连接的 DMA 控制器提供服务时，内置 DMA 请求计数器（每个请求生成器通道一个计数器）的值就会减少。

出现下溢时，请求生成器通道会停止生成 DMA 请求，DMA 请求计数器将在收到下一个触发事件时自动重新加载其设定值。

图 4. DMA 请求生成



如果在生成器管理之前已触发的 DMA 请求序列时接收到一个新的触发事件，则请求触发器事件上溢出标志位 **OFx** 由状态 **DMAMUX_RGSR** 寄存器中的硬件确认。

2.3 请求生成与同步

为了实现自主传输和控制场景, **DMAMUX** 提供在相同配置中组合请求生成和请求同步功能的可能性。

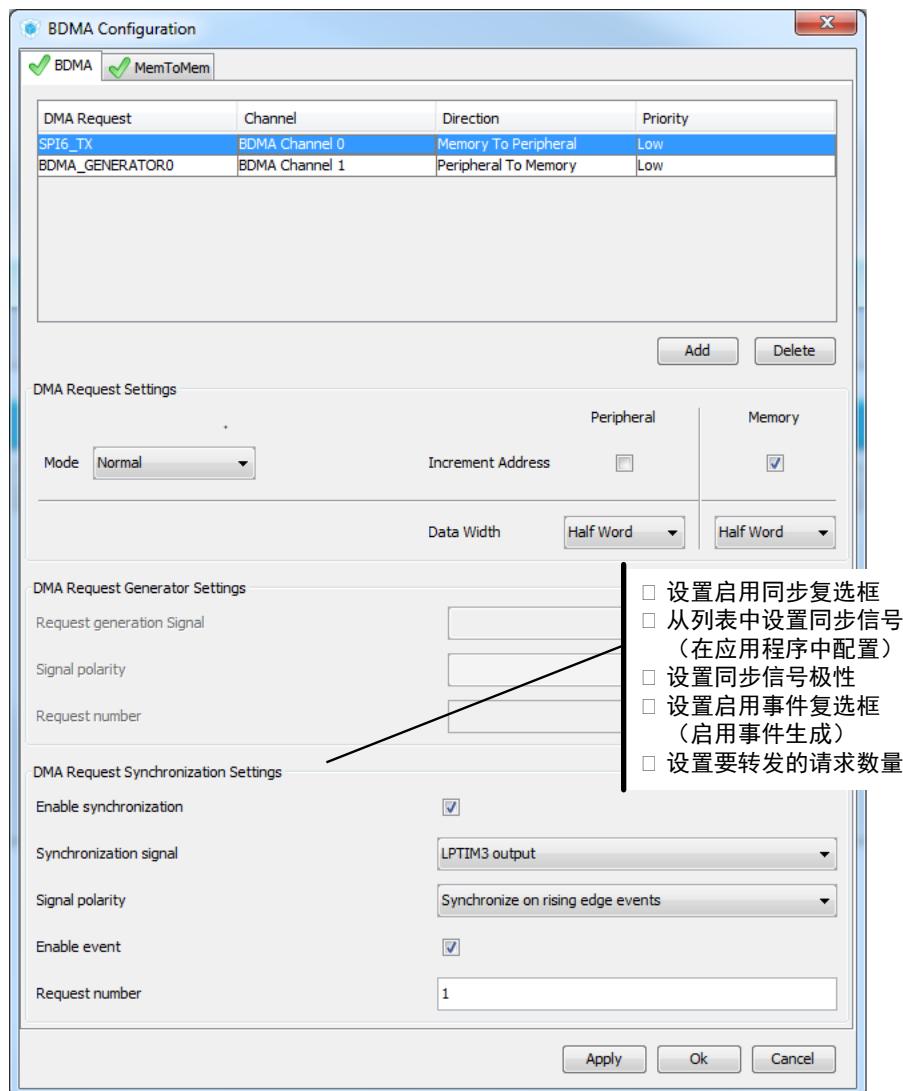
3 DMAMUX 样例

这些样例基于 4.26.1 版本的 STM32CubeMX 工具。

3.1 使用同步的示例

在配置 DMA 通道以服务外设 DMA 请求线（例如 SPI6_TX）之后，可以启用同步块，如下图所示。在这种情况下，使用 LPTIM3_out 信号上升沿控制传输周期。

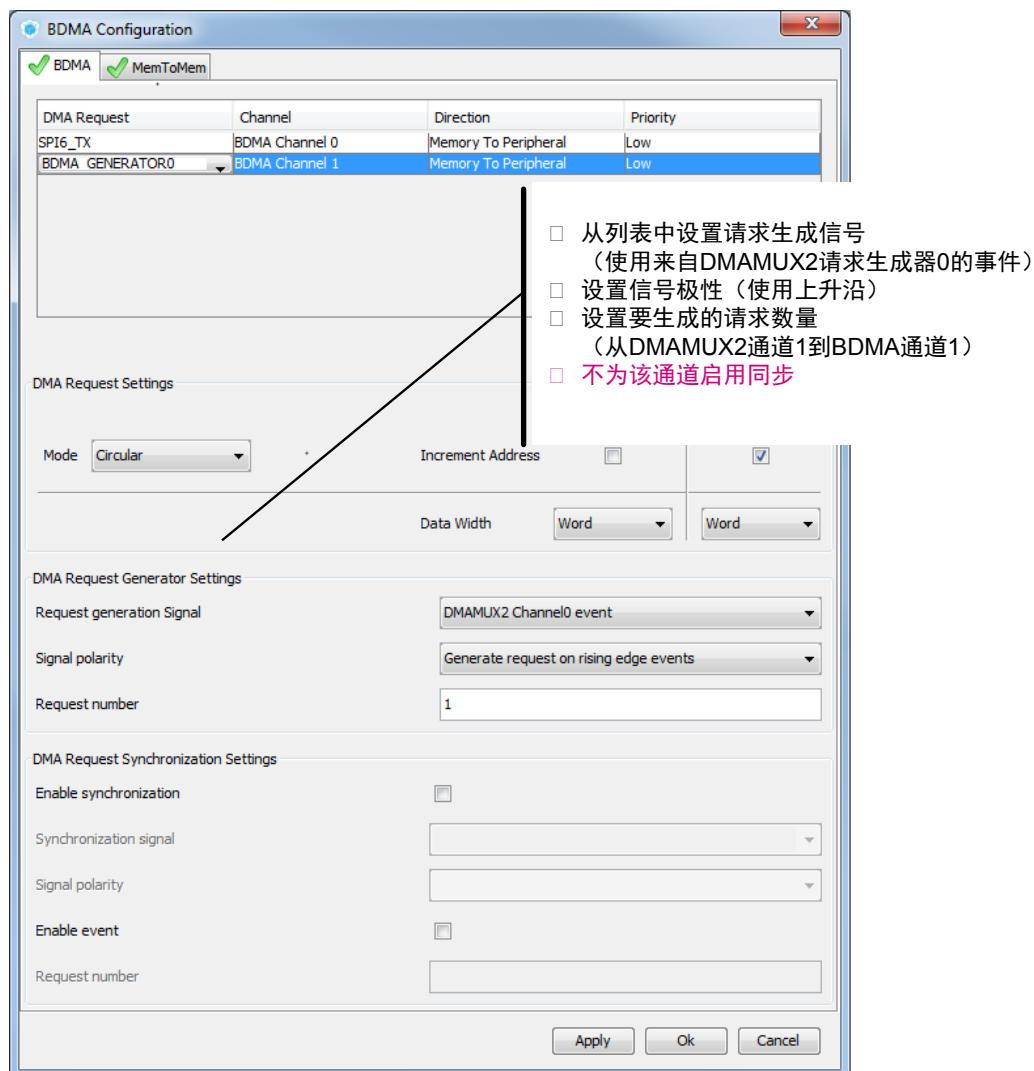
图 5. 使用同步的示例（基于 STM32CubeH7）



3.2 使用 DMAMUX 请求生成器子块的示例

为了实现部分自动化，可以在 DMA 传输到 SPI6 之后生成新的 DMA 传输。由于 DMAMUX 通道 0 事件生成，可以触发请求生成器。可以如下图所示配置该场景。

图 6. 使用请求生成器的示例（基于 STM32CubeH7）



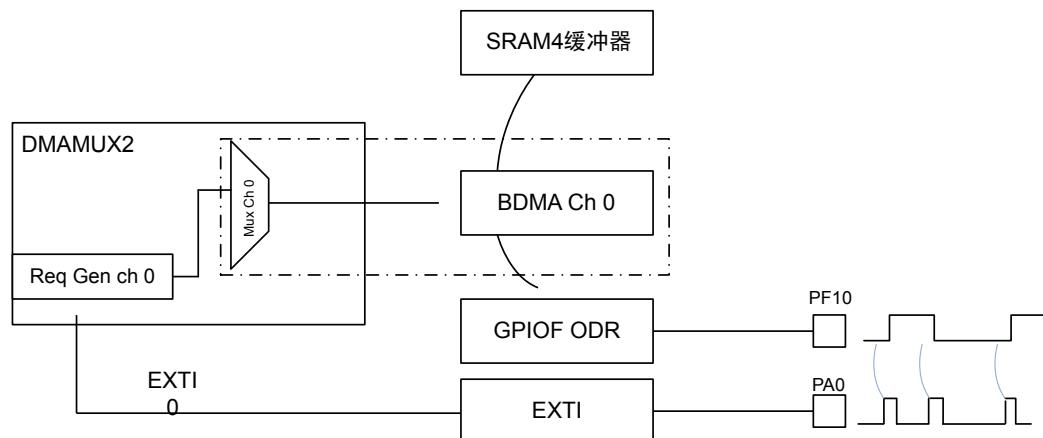
3.3 STM32CubeH7 样例

以下示例取自 STM32CubeH7:

- **DMAMUX_RequestGen**

该示例使用 EXTI0 线触发 DMAMUX 请求生成器，并将 DMA 数据从 SRAM 缓冲区传输到 GPIO 输出数据寄存器，在每次 EXTI0 上升沿发生时更改输出引脚状态。

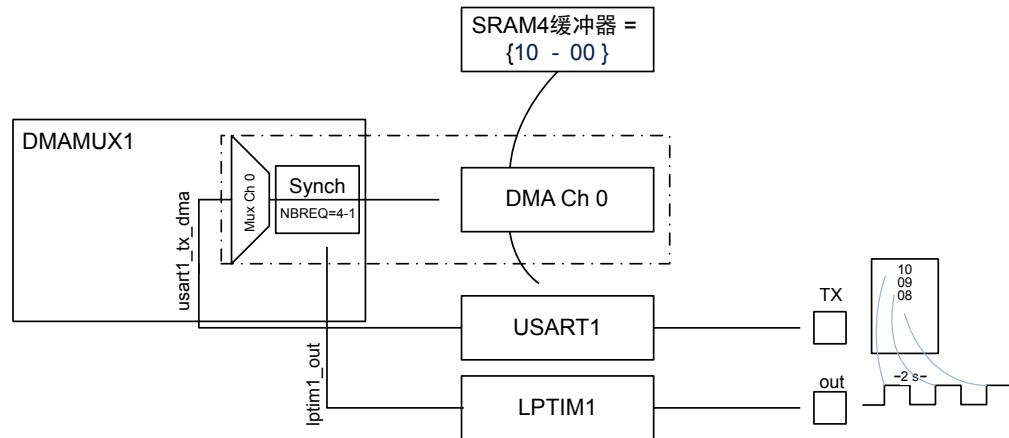
图 7. DMAMUX_RequestGen



- **DMAMUX_SYNC**

该示例使用 DMA 同步模式下的 USART1 以 2 秒的间隔发送从 10 到 00 的倒计时。可以配置 DMAMUX 同步块，以 LPTIM1 输出信号同步 DMA 传输。同步信号（LPTIM1 输出信号）的每个上升沿授权使用 DMA 将四个 USART1 请求传输到 USART1 外设。这四个请求表示两个字符'\n\r'加上从'10'倒数到'00'的两个字符。配置 LPTIM1，以 2 秒间隔生成 PWM。

图 8. DMAMUX_SYNC



4

结论

DMAMUX 控制器的设计目的是简化嵌入式应用的资源分配，因为它为向 DMA 通道动态分配外设提供了灵活性。此外，它还提供同步机制，允许通过传输控制和同步为 CPU 进一步减荷，提高了 DMA 功能。还可以组合运用同步和请求生成实现功率优化的数据传输（在不涉及 CPU 的自主模式下）。

版本历史

表 2. 文档版本历史

日期	版本	变更
2018 年 10 月 16 日	1	初始版本。
2018 年 11 月 20 日	2	更新了表 1. 适用产品。

目录

1	DMAMUX 描述	2
2	DMAMUX 功能	4
2.1	请求路由与同步	6
2.1.1	无条件请求转发	6
2.1.2	有条件请求转发	6
2.2	请求生成	8
2.3	请求生成与同步	9
3	DMAMUX 样例	11
3.1	使用同步的示例	11
3.2	使用 DMAMUX 请求生成器子块的示例	12
3.3	STM32CubeH7 样例	14
4	结论	17
	版本历史	18

图一览

图 1.	DMAMUX 请求复用器	3
图 2.	DMAMUX 简化框图	5
图 3.	DMA 请求线复用器通道 - 事件生成	7
图 4.	DMA 请求生成	9
图 5.	使用同步的示例（基于 STM32CubeH7）	12
图 6.	使用请求生成器的示例（基于 STM32CubeH7）	14
图 7.	DMAMUX_RequestGen.	15
图 8.	DMAMUX_SYNC	16

重要通知 - 请仔细阅读

意法半导体公司及其子公司（“ST”）保留随时对 ST 产品和/或本文档进行变更、更正、增强、修改和改进的权利，恕不另行通知。买方在订货之前应获取关于 ST 产品的最新信息。ST 产品的销售依照订单确认时的相关 ST 销售条款。

买方自行负责对 ST 产品的选择和使用，ST 概不承担与应用协助或买方产品设计相关的任何责任。

ST 不对任何知识产权进行任何明示或默示的授权或许可。

转售的 ST 产品如有不同于此处提供的信息的规定，将导致 ST 针对该产品授予的任何保证失效。

ST 和 ST 徽标是 ST 的商标。若需 ST 商标的更多信息，请参考 www.st.com/trademarks。所有其他产品或服务名称均为其各自所有者的财产。

本文档中的信息取代本文档所有早期版本中提供的信息。

© 2019 STMicroelectronics - 保留所有权利