

使用 Kinetis L 系列的异步 DMA 特性

作者: Chris Brown

内容

1 简介

器件功耗研究以及低功耗设计是当今 MCU 设计中的热门话题。在现今的 MCU 系统中,降低 MCU 功耗的常见途径是减少正在开关状态的晶体管的数量。从而意味着在任何情况下尽可能地关断外设和/或内核。这导致大多数 MCU 系统中的 MCU 基本处于静态,只能与外界进行极少的交互。本应用笔记讨论如何使用 Kinetis L 系列微控制器的异步 DMA 特性实现与外界的交互,同时降低正在开关状态的晶体管的数量,从而节省应用设计的功耗。

本应用笔记将讨论使用 Kinetis L 系列微控制器和异步 DMA 特性来实现该目标的多种途径。

值得注意的是,该应用笔记是基于 KL25Z128VLK4 微控制器进行讨论的,因而某些特性不一定适用于所有 Kinetis L 系列器件。

2 异步特性

针对本文档的用途,异步特性是指在工作时无时钟,或时钟无需与内核时钟同步(即,无需从内核时钟计时,或从内核时钟进行分频或倍频)的特性。

异步 DMA 特性允许数据在外围模块的寄存器上保存或复制,极有利于实现“在限制功耗的同时与外界交互”这一目标。然而,若要利用异步 DMA 特性,器件还须具备 DMA

1	简介.....	1
2	异步特性.....	1
3	将应用配置为异步 DMA 操作.....	2
3.1	将 DMA 配置为异步操作.....	2
3.2	配置模块为异步 DMA 操作.....	2
4	异步 DMA 操作应用.....	3
4.1	ADC 和 TPM 配置.....	4
4.2	DMA 配置.....	4

以外的其他异步特性。Kinetis L 系列提供许多异步特性，可在持续执行任务的同时实现低功耗。器件的其他异步外设包括：

- 模数转换器 (ADC)
- 内置集成电路 (I2C)
- 通用异步收发器 (UART)
- 数模转换器 (DAC)
- 定时器脉冲模块 (TPM)
- 低功耗定时器 (LPTMR)
- 串行外设接口 (SPI)

下面的章节将讨论如何使用异步 DMA 特性以及该特性如何降低功耗。此外，还将描述一个使用 L 系列异步 DMA 特性的实际应用。

3 将应用配置为异步 DMA 操作

将应用配置为异步 DMA 操作由两个任务组成：

- 配置 DMA
- 配置将用于异步 DMA 的外围模块。

下述章节将讨论 DMA 配置。

3.1 将 DMA 配置为异步操作

仅需将异步 DMA 请求使能位置 1 (DMA_DCRn[EADREQ] = 1)，便能将 DMA 配置为异步操作。所有其他位和寄存器均须配置为同步操作下的状态。在配置 DMA 为异步操作时，应遵循这些通用指南。

用户须注意，当计划在低于超低功耗等待 (VLPW) 模式下使用异步 DMA 操作时，不能使用 PIT 门控传输，因为 PIT 在低于 VLPW 模式的功耗模式下不工作。如果该应用需要使用门控传输和异步操作，则必须使用 Cycle Steal 功能，该功能会强制 DMA 通道执行一次传输，然后等待另一个触发事件才能执行另一次传输。该功能将在本应用笔记的示例应用中进行论证。

3.2 配置模块为异步 DMA 操作

每个模块将针对不同类型的应用进行不同的配置。因此，没有针对所有模块的既定设定。但用户必须遵守某些规则才能实现异步 DMA 操作。首先，以下列出了可以产生异步 DMA 请求的外设模块。

- UART0 (TX)
- UART0 (RX)
- TPM (通道中断)
- TPM (溢出)
- ADC
- CMP
- PORTA
- PORTD

注

这些模块是针对 KL25 器件而言的。对于具体的芯片，能够产生异步 DMA 请求的模块也许会有不同。此外，只有某些定时器通道才能产生异步 DMA 请求。当选择使用具有异步 DMA 请求的特定外设时，建议参考具体器件的参考手册（可从 freescale.com 上获取）。

当配置所需模块为异步 DMA 功能操作时，应遵循以下通用原则：

1. 使能要使用的时钟。该时钟必须能够驱动模块并工作在要使用的低功耗模式下。请注意，不是所有外设均需要使能时钟，例如 PORTA、PORTD 和 CMP。
2. 禁用要使用的 DMA 通道。
3. 禁用要配置的模块。
4. 按照配置中断操作的方式对模块进行配置。
5. 配置模块使用用户为该应用使能的时钟（如果可用）。
6. 将对应的 DMA 使能位置 1（例如，ADC0_SC2[DMAEN] = 1）。
7. 使能模块。
8. 使能要用于该模块的 DMA 通道。
9. 进入所需的低功耗模式。

4 异步 DMA 操作应用

用于展示的实例是一个根据用户输入实现 LED 亮度从 0-100%变化的应用。

必须记录用户输入，并计算平均值和平均变化量，然后输出到终端。该应用复杂程度一般，需要使用异步 DMA 来实现可能的最低功耗。下图显示了应用的框图。

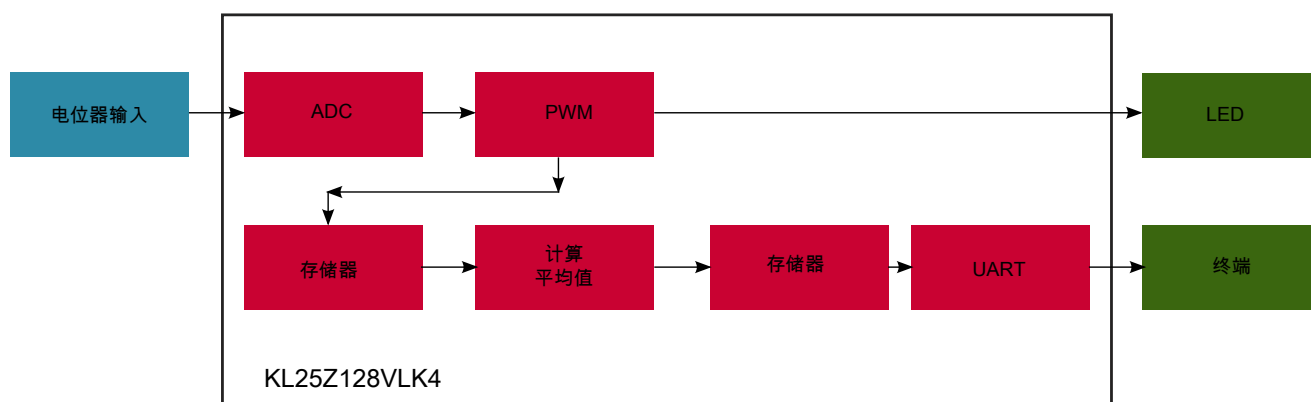


图 1. 异步 DMA 应用框图

应用要求（从微控制器角度而言）如下：

- 测量电位器上的电压。
- 输出 50 Hz PWM。
- 基于 ADC 结果改变 PWM 信号的脉冲宽度。
- 在存储器阵列中存放 ADC 读值。
- 使用异步 DMA 来执行外设/存储器之间的所有数据传输。
- 计算测量结果的平均值和平均值导数。
- 在终端输出上显示测量结果的平均值和平均值导数。
- 使用超低功耗运行（VLPR）模式。

该应用的代码在结构方面分为三个不同的部分：

- 初始化
- 低功耗模式
- 计算

下面的章节将详细介绍每个部分。

注

该应用通过 ProcessorExpert 10.0 和 IAR 来生成驱动程序代码。

4.1 ADC 和 TPM 配置

框图中首先要实现的组件是 ADC 和 PWM，因为这些组件对于系统的操作至关重要。如要求列表所述，PWM 输出必须设置为在低功耗模式下产生 50 Hz 信号。L 系列的 TPM 模块非常适用于该任务，因为其无需 CPU 干预便能产生 PWM 信号，并且可用于低功耗模式。下一个任务是选择模块的时钟源。有 4 种 TPM 时钟可供选择：

- MCGIRCLK
- OSCERCLK
- MCGFLLCLK
- MCGPLLCLK

其中，可在低功耗模式下使用且功耗最低的时钟是 MCGIRCLK。因此将使用该时钟。

配置 ADC 时应始终考虑转换分辨率、转换时间，以及应用是否调用差分输入。简化起见，将假设该应用不在高噪声环境下工作，因而无需差分输入。

从转换分辨率入手，示例应用需要使 LED 亮度进行从 0 至 100% 的变化。也就是说提供给 LED 的电流必须从 0 至 100% 变化。PWM 占空比（由 TPM 输出）由写入 16 位通道值寄存器中的值控制。向通道值寄存器写入 0x0000 将使 PWM 占空比为 0%，向通道值寄存器写入 0xFFFF 将使 PWM 占空比为 100%。因此，ADC 将配置为 16 位转换。

当选择转换时间时，用户必须牢记该应用的目标是低功耗。通常有两种操作可实现低功耗：

- 较慢地运行器件，从而消耗较少的功。
- 尽可能快地运行器件，然后将其置为低功耗状态，从而消耗较少的功。

该应用对用户两次采样间必须等待的时间没有严格要求。因此，希望器件利用低功耗模式，并且大部分时间处于该模式下。从而，将选择可能的最快转换时间。假定 ADC 将使用其自身的内部时钟（因为该时钟工作在异步模式下），则可执行的最快转换时间为 10.42 μ s。使用最快转换时间还将减小由输入变化产生的误差。

4.2 DMA 配置

在该应用中使用 DMA 时，首先要考虑的变量是需要使用的 DMA 通道数。KL25Z128VLK4 器件中有 6 个方框（见图 1）：

- ADC
- PWM (TPM)
- 两个存储器方框（一个为阵列；另一个保存计算结果）
- 计算（计算平均值和平均变化量）
- UART 输出

此外，有些组件互相连接，表示它们之间存在互操作。因此，明确哪些组件/连接不需要 CPU 或 DMA 的干预是很重要的。

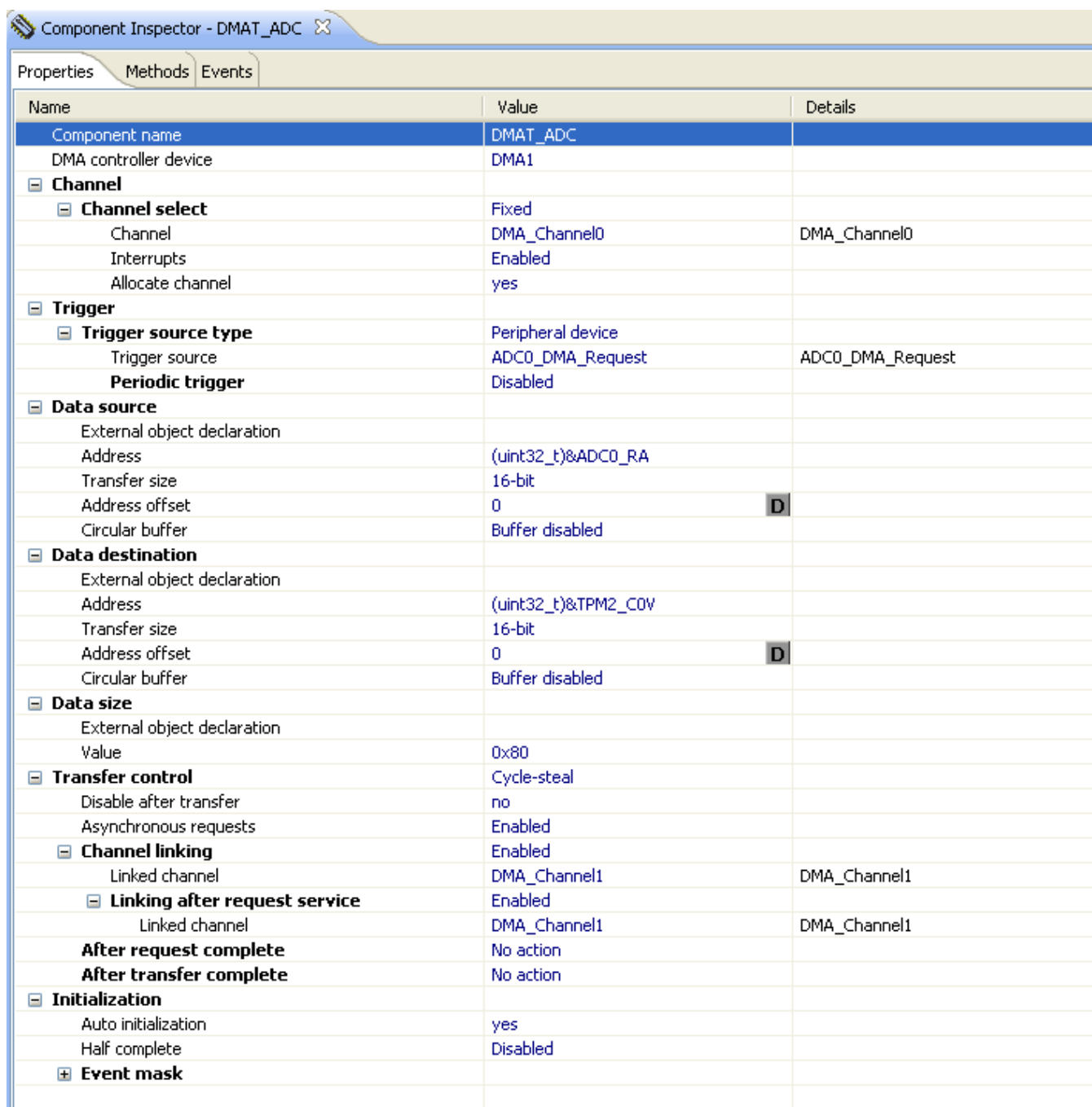
- ADC 无需 CPU 或 DMA 就可以执行异步转换，因此其与电位器的连接无需 DMA 干预。

TPM PWM 输出也是如此。它无需干预便能异步发生，因此其与 LED 的连接无需 DMA 通道。

- UART 也可以异步发送，因此 UART 与终端的连接不需要 DMA 通道。
- 但若没有 CPU 或 DMA 干预，数据无法到达 UART。计算必须由 CPU 执行。由于 CPU 在执行该动作时处于运行状态，可通过其从/向存储器取出/存入数据。因此，这些连接将无需 DMA。

现在，还剩下 ADC 到 TPM、TPM 到存储器，以及存储器到 UART 的连接。接下来将讨论这些连接。

- ADC 到 TPM 的 DMA 配置相当直接。应用要求将每个 ADC 结果移入相应通道的 TPM 通道值寄存器。因此，下图显示了 ProcessorExpert (PEX) 配置。



Name	Value	Details
Component name	DMAT_ADC	
DMA controller device	DMA1	
Channel		
Channel select	Fixed	
Channel	DMA_Channel0	DMA_Channel0
Interrupts	Enabled	
Allocate channel	yes	
Trigger		
Trigger source type	Peripheral device	
Trigger source	ADC0_DMA_Request	ADC0_DMA_Request
Periodic trigger	Disabled	
Data source		
External object declaration		
Address	(uint32_t)&ADC0_RA	
Transfer size	16-bit	
Address offset	0	D
Circular buffer	Buffer disabled	
Data destination		
External object declaration		
Address	(uint32_t)&TPM2_COV	
Transfer size	16-bit	
Address offset	0	D
Circular buffer	Buffer disabled	
Data size		
External object declaration		
Value	0x80	
Transfer control	Cycle-steal	
Disable after transfer	no	
Asynchronous requests	Enabled	
Channel linking	Enabled	
Linked channel	DMA_Channel1	DMA_Channel1
Linking after request service	Enabled	
Linked channel	DMA_Channel1	DMA_Channel1
After request complete	No action	
After transfer complete	No action	
Initialization		
Auto initialization	yes	
Half complete	Disabled	
Event mask		

图 2. PEx ADC 到 TPM 的 DMA 配置

请注意，选择的触发机制为 ADC0 DMA 请求。通过该配置，ADC 将在一次转换完成后触发 DMA 请求。

源地址和目的地址分别指定为 ADC0 结果寄存器和 TPM2 通道 0 值寄存器。这意味着 ADC 结果将传输到 TPM 通道值寄存器。此外，请注意源地址和目的地址缓冲区和地址偏移量均被禁用。这样设置 DMA 可使每次传输均使用相同的源地址和目的地址位置。

在传输控制部分，置位 Cycle Steal。这样可防止 DMA 持续进行从源地址位置向目的地址位置的传输，强制每个 DMA 请求仅执行一次传输。

同时还使能了通道链接。选择该选项是为了在该模块每次传输后产生另一个 DMA 请求。该传输会继而触发从 TPM 通道值寄存器到存储器的 DMA 传输。这确保了每次的 ADC 转换结果均会保存到存储器中。

- 从 TPM 通道值寄存器到存储器的 ProcessorExpert DMA 通道配置如下图所示。

Name	Value	Details
Component name	DMAT_Mem	
DMA controller device	DMA1	
Channel		
Channel select	Fixed	
Channel	DMA_Channel1	DMA_Channel1
Interrupts	Enabled	
Allocate channel	yes	
Trigger		
Trigger source type	Explicit software	
Data source		
External object declaration		
Address	(uint32_t)&TPM2_COV	
Transfer size	16-bit	
Address offset	0	D
Circular buffer	Buffer disabled	
Data destination		
External object declaration		
Address	(uint32_t)&MyStaticArray	
Transfer size	16-bit	
Address offset	2	D
Circular buffer	128 Bytes	
Data size		
External object declaration		
Value	SAMPLE_SIZE	
Transfer control	Cycle-steal	
Disable after transfer	no	
Asynchronous requests	Enabled	
Channel linking	Disabled	
After request complete	No action	
After transfer complete	No action	
Initialization		
Auto initialization	yes	
Half complete	Disabled	
Event mask		

图 3. PEx TPM 到存储器的 DMA 配置

请牢记，该 DMA 传输链接从 ADC 到 TPM 的传输。因此，无需硬件触发，用户可以选择使用软件触发模式触发 DMA。请注意，选择 TPM2 通道 0 值寄存器作为源（缓冲区禁用），而阵列存储器作为目的地（缓冲区使能）。这样可允许在程序中声明静态数组，并且 DMA 将传输该寄存器中 TPM2 通道 0 的值到指定的存储器阵列中。

注

若使用缓冲区选项和用户定义数组，很重要的一点是，在用户软件中定义数组时要注意的数组的字节对齐问题。否则，DMA 可能会写到已定义数组外的位置，从而导致意外结果。

还需注意该 DMA 传输已选择 Cycle Steal 选项。这将有效地将这些传输与 ADC 转换传输同步，从而确保每个 ADC 结果均保存用于计算。

- 从存储器到 UART 的 ProcessorExpert DMA 通道配置如下图所示。

Name	Value	Details
Component name	DMAT_UART_OUT	
DMA controller device	DMA1	
Channel		
Channel select	Fixed	
Channel	DMA_Channel2	DMA_Channel2
Interrupts	Disabled	
Allocate channel	yes	
Trigger		
Trigger source type	Peripheral device	
Trigger source	UART0_Transmit_DMA_Request	UART0_Transmit_DMA_Request
Periodic trigger	Disabled	
Data source		
External object declaration	#include "Events.h"	
Address	(uint32_t)&MyOutputArray	
Transfer size	8-bit	
Address offset	1	D
Circular buffer	Buffer disabled	
Data destination		
External object declaration		
Address	(uint32_t)&UART0_D	
Transfer size	8-bit	
Address offset	0	D
Circular buffer	Buffer disabled	
Data size		
External object declaration		
Value	OUTPUT_ARRAY_SIZE	
Transfer control	Cycle-steal	
Disable after transfer	yes	
Asynchronous requests	Enabled	
Channel linking	Disabled	
After request complete	No action	
After transfer complete	No action	
Initialization		
Auto initialization	yes	
Half complete	Disabled	
Event mask		

图 4. PEX 存储器到 UART 的 DMA 配置

存储器到 UART 的 DMA 传输是该项目中最简单的 DMA 配置。配置为将数据从用户定义的数组 (MyOutputArray) 移动到 UART D 寄存器。该操作无需缓冲区。一旦 DMA 通道配置完成并使能, 发送数据寄存器空标志清零 (UARTx_S1[TDRE]) 后, UART 立即开始请求数据 (在已配置并使能的条件下)。只要 UARTx_D 寄存器为空, DMA 就继续传输, 直到字节计数寄存器递减为零。此时, 根据 ProcessorExpert 配置中的定义, 禁止硬件请求。

请注意, 这里再次选择 Cycle Steal 选项。如果未选择该选项, 则一旦写入 UARTx_D 寄存器, DMA 就会将输出内存 (数组) 中的所有数据立即传输到 D 寄存器中。这会导致 UART 仅输出输出数组中的第一个字符。这是因为用户在写入另一个字符前必须等待发送寄存器变为空。Cycle Steal 选项可实现该要求。

How to Reach Us:

Home Page:
freescale.com

Web Support:
freescale.com/support

本文档中的信息仅供系统和软件实施方使用 Freescale 产品。本文并未明示或者暗示授予利用本文档信息进行设计或者加工集成电路的版权许可。Freescale 保留对此处任何产品进行更改的权利，恕不另行通知。

Freescale 对其产品在任何特定用途方面的适用性不做任何担保、表示或保证，也不承担因为应用程序或者使用产品或电路所产生的任何责任，明确拒绝承担包括但不限于后果性的或附带性的损害在内的所有责任。Freescale 的数据表和/或规格中所提供的“典型”参数在不同应用中可能并且确实不同，实际性能会随时间而有所变化。所有运行参数，包括“经典值”在内，必须经由客户的技术专家对每个客户的应用程序进行验证。Freescale 未转让与其专利权及其他权利相关的许可。Freescale 销售产品时遵循以下网址中包含的标准销售条款和条件：freescale.com/SalesTermsandConditions。

Freescale, the Freescale logo, and Kinetis, are trademarks of Freescale Semiconductor, Inc., Reg. U.S. Pat. & Tm. Off. All other product or service names are the property of their respective owners.

© 2012 Freescale Semiconductor, Inc.

© 2012 飞思卡尔半导体有限公司

