

Application Note

MSPM0 - 高级控制计时器有助于实现更好的控制和更好的数字输出

Yuhao Zhao

摘要

计时器模块 (TIMx) 是一个带有多个比较/捕获块的计时器计数模块。根据器件，提供两种类型的计时器：通用计时器 (TIMA) 和高级控制计时器 (TIMA)。这两个计时器都使用通用计时器架构，可在具有常见功能的计时器实例之间轻松迁移。这更大限度地减少了为基于计时器的应用编写额外软件的需求，并允许在 TIMx 实例之间轻松移植和维护。

TIMA 模块包含一个由可编程预分频器驱动的 16 位自动重新加载计数器，以及最多四个用于多次捕获/比较、具有死区插入的 PWM 输出和间隔计时的捕获/比较 (CC) 块。TIMA 具有广泛的事件生成功能，可生成不同的计数器事件（例如溢出事件、重新加载事件以及来自每个捕获/比较寄存器的事件）。它还具有处理由内部或外部电路生成的故障信号以指示系统中故障的硬件设计。

基于 TIMA，您可以配置更丰富的计时器模块行为，以设计和实现更复杂的应用，例如电机控制、LED 模块、电源模块等。高级控制计时器可帮助您实现更好的控制和更好的数字输出。

内容

1 引言	2
2 TIMA 和 TIMG 之间的比较	2
3 用例 - 3 对具有死区插入的互补 PWM	4
3.1 原理.....	4
3.2 实现.....	4
4 用例 - 具有影子加载和比较功能的时序关键型 PWM 控制	7
4.1 原理.....	7
4.2 实现.....	7
5 用例 - 故障处理程序	8
5.1 原理.....	8
5.2 实现.....	8
6 用例 - 通过软件强制输出禁用 PWM	10
6.1 原理.....	10
6.2 实现.....	10
7 用例 - 非对称 PWM	11
7.1 原理.....	11
7.2 实现.....	11
8 用例 - 使用重复计数器实现更佳的中断生成	14
8.1 原理.....	14
8.2 实现.....	14
9 总结	15
10 参考资料	15

商标

所有商标均为其各自所有者的财产。

1 引言

本应用手册介绍了如何配置 TIMA 以实现独特的应用设计，并介绍了如何使用 SysConfig 实现快速配置。如果您对 TIMA 的特定配置（如寄存器）有疑问，请参阅特定器件的 TRM。

在此应用手册中：

- “TIMx”需要理解为在 TIMA 和 TIMA 上可用的一个常见功能
- “TIMA”需要理解为仅在 TIMA 上可用的功能
- “TIMG”需要理解为仅在 TIMG 上可用的功能

2 TIMA 和 TIMG 之间的比较

每个 TIMA 实例的具体特性包括：

- 具有重复重新加载模式的 16 位递增、递减或递增/递减计数器
- 可选和可配置的时钟源
- 用于对计数器时钟频率进行分频的 8 位可编程预分频器
- 重复计数器，仅在计数器的给定周期数之后生成中断或事件
- 最多四个独立通道，用于：
 - 输出比较
 - 输入捕捉
 - PWM 输出（边沿对齐和中心对齐）
 - 单次触发模式
- 两个用于内部事件的额外捕捉/比较通道 (CC4/CC5)
- 用于加载和比较值的影子寄存器
- 具有可编程死区插入功能的互补 PWM 输出
- 非对称 PWM 输出

TIMG 模块包含由可编程预分频器驱动的 16 位和 32 位自动重新加载计数器，以及两个用于多个捕获/比较、PWM 输出和间隔计时的捕获/比较 (CC) 块。TIMG 还具有广泛的事件生成功能，包括针对各种用例的计数器溢出、重新加载和捕获/比较操作。

TIMG 的具体特性包括：

- 具有重复重新加载模式的 16 位递增、递减或递增/递减计数器
- 用于对计数器时钟频率进行分频的 8 位可编程预分频器
- 最多两个独立通道，用于：
 - 输出比较
 - 输入捕捉
 - PWM 输出（边沿对齐和中心对齐）
 - 单次触发模式
- 用于加载和比较值的影子寄存器模式
- 支持正交编码器接口 (QEI)
- 用于位置检测和速度计算的 3 路输入霍尔传感器模式
- 支持同一电源域中不同 TIMx 实例之间的同步和交叉触发
- 支持使用事件的 CPU 中断生成以及跨外设（如 ADC、DAC 等）

表 2-1 展示了 TIMA 和 TIMG 实例的配置。

表 2-1. TIMx 实例配置

实例	电源域	计数器分辨率	预分频器	重复计数器	CCP 通道 (外部/内部)	外部 PWM 通道	相负载	影子负载	影子 CC	Deadband	故障处理 程序	QEI/霍尔输 入模式
TIMG0	PD0	16 位	8 位	-	2	2	-	-	-	-	-	-
TIMG1	PD0	16 位	8 位	-	2	2	-	-	-	-	-	-
TIMG2	PD0	16 位	8 位	-	2	2	-	-	-	-	-	-
TIMG3	PD0	16 位	8 位	-	2	2	-	-	-	-	-	-
TIMG4	PD0	16 位	8 位	-	2	2	-	是	是	-	-	-
TIMG5	PD0	16 位	8 位	-	2	2	-	是	是	-	-	-
TIMG6	PD1	16 位	8 位	-	2	2	-	是	是	-	-	-
TIMG7	PD1	16 位	8 位	-	2	2	-	是	是	-	-	-
TIMG8	PD0	16 位	8 位	-	2	2	-	-	-	-	-	是
TIMG9	PD0	16 位	8 位	-	2	2	-	-	-	-	-	是
TIMG10	PD1	16 位	8 位	-	2	2	-	-	-	-	-	是
TIMG11	PD1	16 位	8 位	-	2	2	-	-	-	-	-	是
TIMG12	PD1	32 位	-	-	2	2	-	-	是	-	-	-
TIMG13	PD0	32 位	-	-	2	2	-	-	是	-	-	-
TIMG14	PD1	16 位	8 位	-	4	4	-	-	-	-	-	-
TIMA0	PD1	16 位	8 位	是	4/2	8	是	是	是	是	是	-
TIMA1	PD1	16 位	8 位	是	2/2	4	是	是	是	是	是	-

3 用例 - 3 对具有死区插入的互补 PWM

3.1 原理

具有死区插入功能的互补输出是基于逆变器且采用半桥拓扑的应用的常见用例。死区对于采用半桥控制的应用非常有用，可以避免击穿情况，例如基于电机驱动器或逆变器的应用。

要生成 3 对具有死区插入的互补 PWM，3 个 CCP 通道需要 TIMA0。以下是该用例的关键特性。

- 生成具有 3 个 CCP 通道的 PWM (仅限 TIMA0)
- 启用互补 CCP 输出通道 (仅限 TIMA)
- 配置死区插入 (仅限 TIMA)

3.2 实现

TIMx 可用于使用计数器和捕捉/比较模块 (比较匹配) 生成所需的脉宽调制 (PWM) 输出波形。CCP 输出的波形生成取决于计数模式和计数器比较操作

例如，在不同的计数模式下，可以生成边沿对齐 PWM 或中心对齐 PWM。图 3-1 展示了典型的 2 通道 PWM。用户可以设置 LOAD 值来配置 PWM 周期，或设置 CC 值来配置占空比。用户还可以配置信号的极性、CCP 输出操作等。有关特定配置，请参阅 TRM。

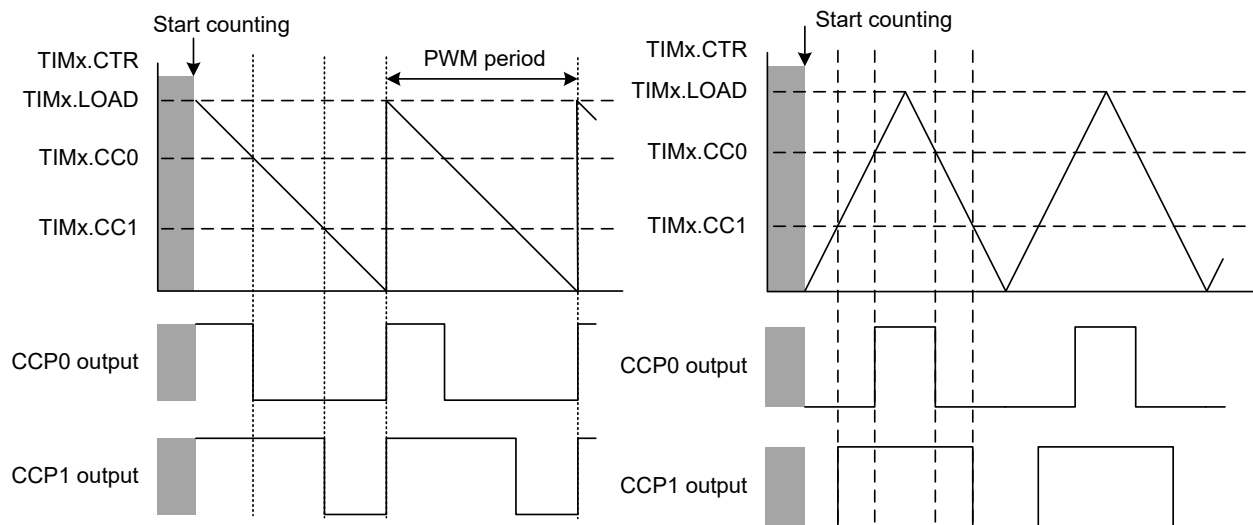


图 3-1. 边沿对齐 PWM 和中心对齐 PWM

TIMA 和 TIMG 都可以生成 PWM。但是，TIMA 在互补 CCP 输出通道 (例如 TIMA0_C1 和 TIMA0_C1N) 上提供该可选功能，用于 TIMA0 CCP 输出通道 1 上的基准 PWM 信号。此外，TIMA0 有 4 个 CCP 通道，而 TIMA1 和 TIMG 有 2 个 CCP 通道。这意味着 TIMA0 可以输出 8 个 PWM (4 对)，TIMA1 可以输出 4 个 PWM (2 对)，而 TIMG 只能输出 2 个 PWM。

- TIMA0 :
 - TIMA0_C0、TIMA0_C0N
 - TIMA0_C1、TIMA0_C1N
 - TIMA0_C2、TIMA0_C2N
 - TIMA0_C3、TIMA0_C3N
- TIMA1 :
 - TIMA1_C0、TIMA1_C0N
 - TIMA1_C1、TIMA1_C1N
- TIMGx :
 - TIMGx_C0
 - TIMGx_C1

TIMA 还提供了死区插入选项 (互补 PWM 信号中的非重叠转换)。多种可配置死区模式可以插入到边沿对齐或中心对齐互补 PWM 中，如图 3-2 和图 3-3 所示。

用户可以使用编程的死区控制寄存器 (TIMA.DBCTL) 来配置死区模式和时序信息。采用模式 0 时，上升延迟和下降延迟应用于输出发生器信号的上升沿和下降沿，以生成 CCP 和 CCP 互补信号。采用模式 1 时，上升延迟和下降延迟均仅应用于 CCP 互补信号。图 3-2 展示了具有死区的边沿对齐 PWM (仅限模式 0)。图 3-3 展示了具有死区的中心对齐 PWM (模式 0 和模式 1)。

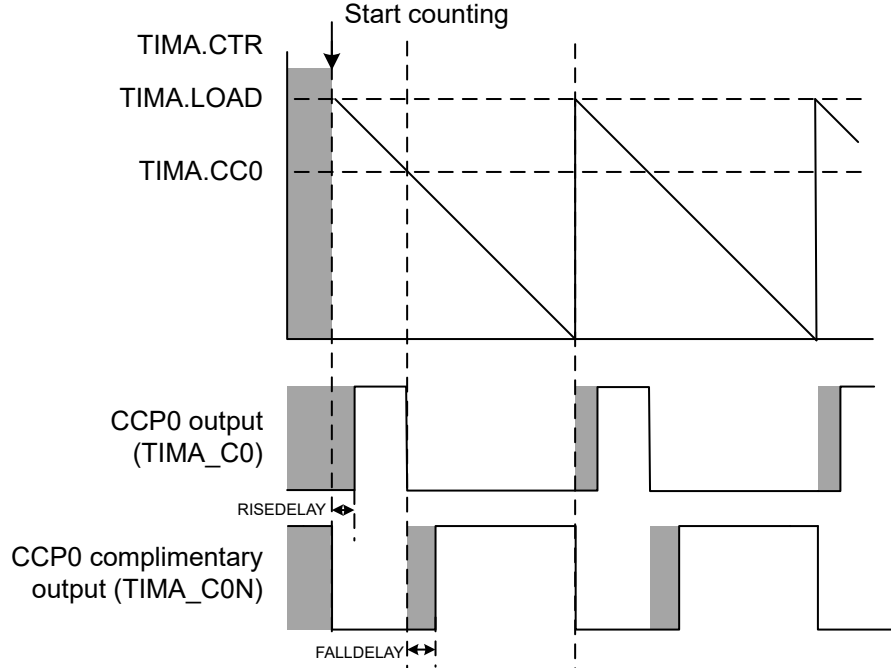


图 3-2. 具有死区的边沿对齐 PWM

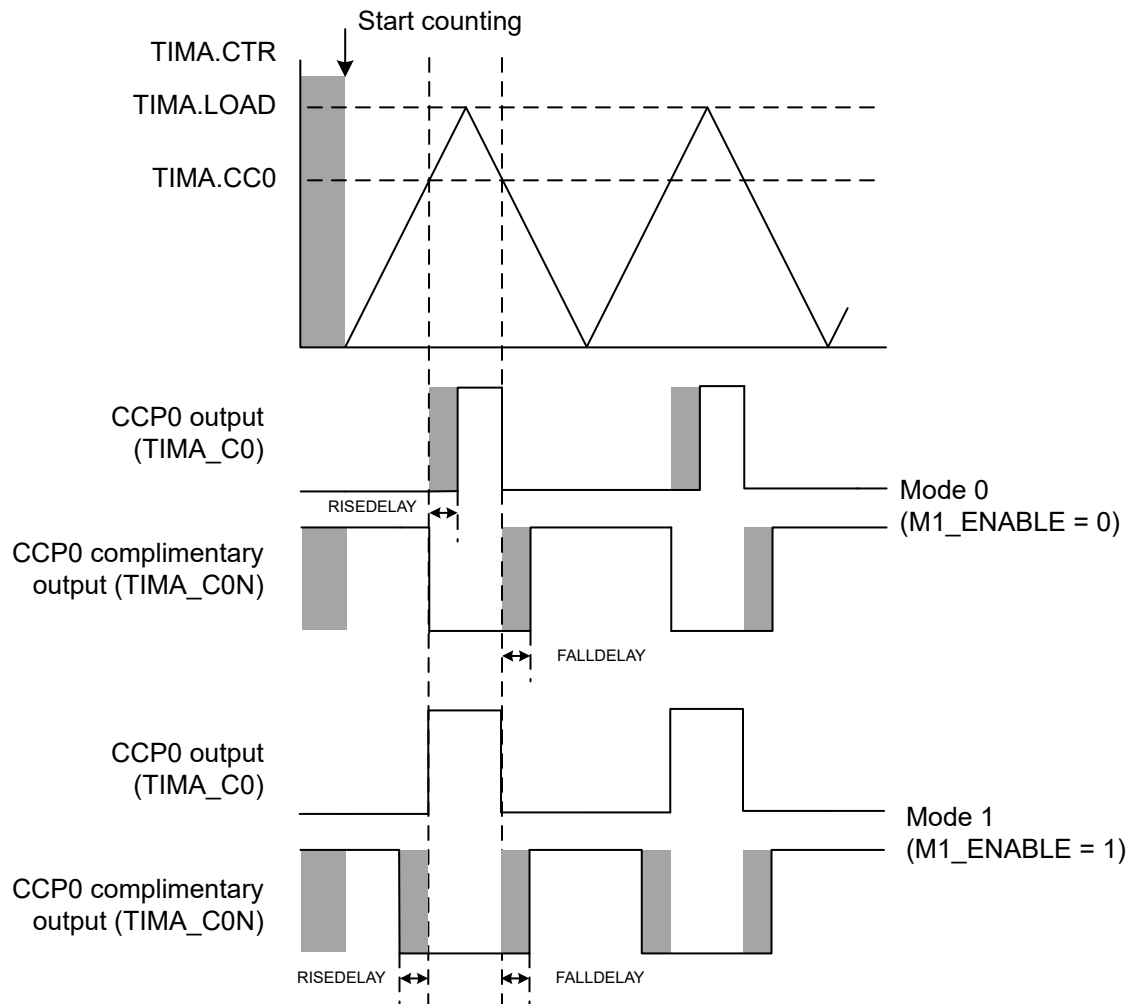


图 3-3. 具有死区的中心对齐 PWM

若要加快开发速度，请参阅以下资源。

- SDK 中的示例：[tima_timer_mode_pwm_dead_band](#)

4 用例 - 具有影子加载和比较功能的时序关键型 PWM 控制

4.1 原理

对于 MSPM0，一些计时器模块具有影子加载和影子比较寄存器功能，使用户可以灵活地保持加载和 CC 值的更新，直到某个事件发生。这在时序关键型应用中非常有用，在这些应用中，PWM 控制信号需要使用正确的时序进行更新，例如占空比更新。

要控制具有影子加载/比较的 PWM，需要 TIMA 或特定 TIMG。以下是该用例的关键特性。

- 生成中心对齐/边沿对齐 PWM
- 启用影子加载
- 配置 CC 更新方法

4.2 实现

利用影子加载功能，可以保持加载值的更新，直到发生归零事件。如果 TIMx 模块具有影子加载功能，则存在一个用于加载值的内部影子寄存器 (TIMx.LOAD)。该影子寄存器可在发生零事件时更新加载值。

对于向上计数模式和向上或向下计数模式，请注意需要影子加载。影子加载可确保 TIMx 在发生归零事件之前计数到加载值，否则加载值可能会立即更新并导致不正确的时序。

当启用影子比较来更新捕捉或比较寄存器 (TIMx.CC) 时，写入相应比较寄存器的值将首先存储到影子比较寄存器中，然后在发生不同事件时传输到比较寄存器。用户还可以配置在发生不同事件时更新 CC 操作。

图 4-1 展示了影子加载和影子比较如何在递增/递减计数模式下针对 TIMx.LOAD 和 TIMx.CC 值在发生归零事件时生效的示例。

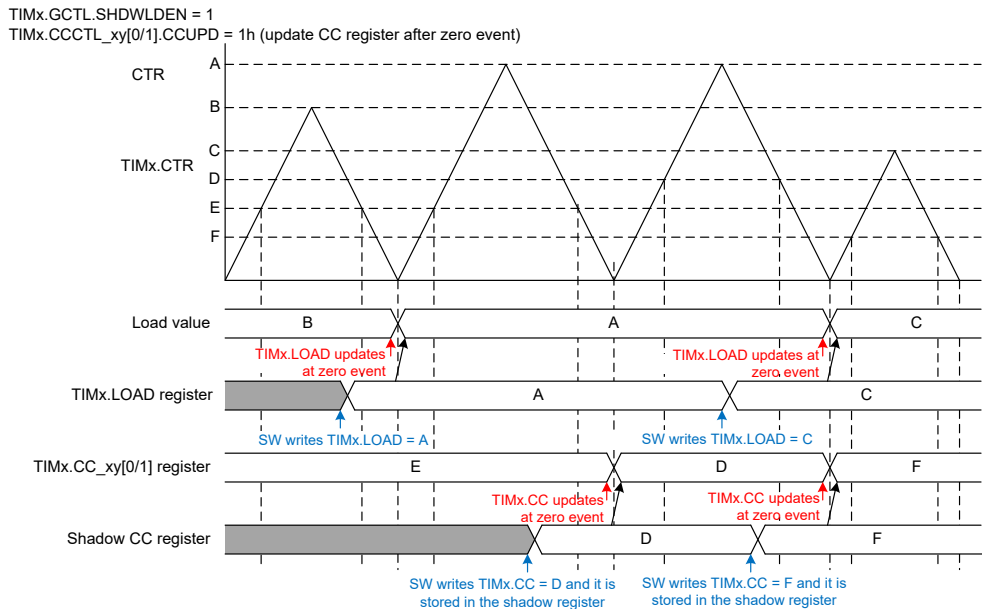


图 4-1. 在递增或递减模式下，影子加载和影子比较在发生归零事件时生效

若要加快开发速度，请参阅以下资源。

- SDK 中的示例：[timx_timer_mode_pwm_edge_sleep_shadow](#)

5 用例 - 故障处理程序

5.1 原理

在实时控制中，通常需要故障处理程序来处理异常情况，例如过流或过压情况。硬件故障行为可实现比软件代码更快的保护。此外，可根据故障处理程序执行更加复杂的 PWM 控制。

仅在 TIMA 中，存在可用于控制 PWM 信号生成的内部和外部故障输入。这些输入的预期用途是作为内部或外部电路指示系统中故障的一种机制。这使硬件能够快速响应外部故障，同时可选地发出中断信号以进行软件校正并使输出信号处于安全状态。

要启用故障处理程序，需要 TIMA。以下是该用例的关键特性。

- 生成中心对齐或边沿对齐 PWM
- 启用故障处理程序 (仅限 TIMA)

5.2 实现

在配置故障处理程序之前，务必考虑系统中故障的以下基本属性，例如：

- 故障输入选择 (来自外部 IC 的故障信号、内部信号等)
- 故障情况的持续时间
- 计数器如何对故障情况的进入和退出做出反应
- 输出信号如何对故障情况的进入和退出做出反应

对于故障输入源，用户可以选择极性并启用各种故障输入源，例如比较器 (COMP) 输出、外部故障引脚、系统时钟故障或触发器。用户还可以控制故障输入的输入滤波。

- 当 COMP 用于检测过流或过压事件时，比较器输出对于故障检测非常有用。
- 有 3 个故障外部信号引脚可用于检测 MCU 外部的故障情况。
- 任何系统时钟故障均可用于将 PWM 输出触发至高阻抗状态。
- 触发器可以配置为生成可检测的故障条件。这对于执行诊断或从事件结构中的其他外设创建故障依赖性非常有用。

故障处理程序可能会对计数器和输出产生影响。对于具有故障条件的计数器行为，可以将计数器设置为立即停止，达到零时停止或继续计数。退出故障模式后，可以将计数器设置为从加载值、0 值重新开始，或从停止的位置继续计数。图 5-1 展示了不同配置下发生故障时的计数器行为。

对于具有故障条件的输出行为，CCP 输出值可以设置为高电平、设置为低电平、切换、三态 (高阻抗) 或不受故障事件的影响。

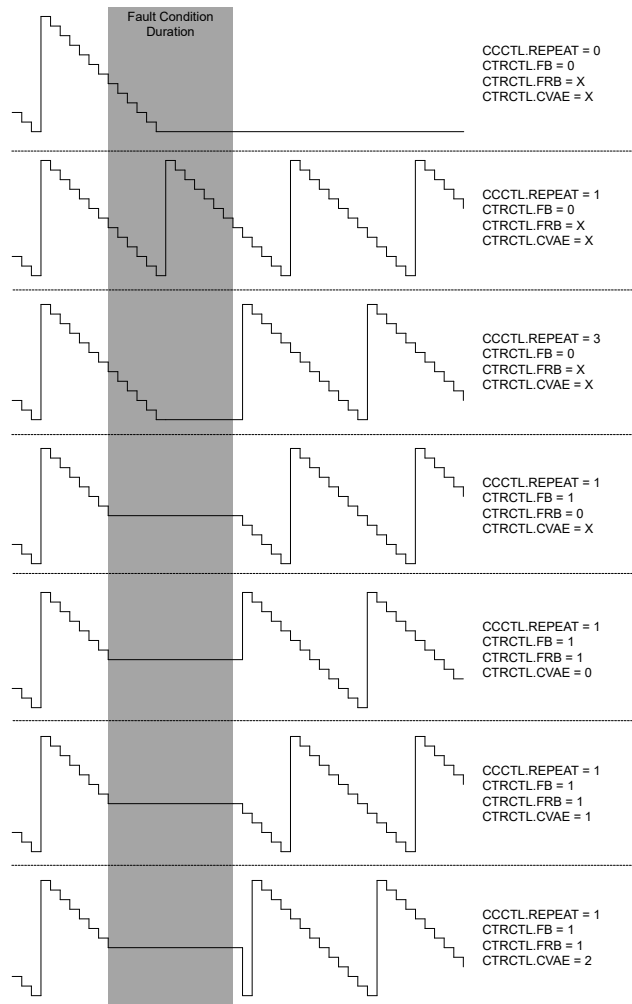


图 5-1. 故障条件下的计数器行为

若要加快开发速度，请参阅以下资源。

- SDK 中的示例：[tima_trigger_fail_mechanism](#)

6 用例 – 通过软件强制输出禁用 PWM

6.1 原理

除了故障处理程序，用户还可以在软件中强制 PWM 输出。可以通过软件直接将每个输出通道信号强制设置为高电平或低电平，而与比较寄存器和计数器之间的任何比较无关。

以下是该用例的关键特性。所有计时器模块均支持该功能。

- 生成中心对齐或边沿对齐 PWM
- 设置 TIMx.CCACT_xy[0/1] 寄存器中的 SWFRCACT 位

6.2 实现

CCP 通道的输出可被强制为高电平或低电平。此外，仅在 TIMA 中，还可将互补输出通道强制设置为高电平或低电平。表 6-1 展示了软件强制输出操作配置选项。

表 6-1. 强制输出操作配置

位字段	值	说明或注释
SWFRCACT/ SWFRCACT_CMPL	0	无强制输出。输出直接来自信号生成块。
	1	强制输出高电平
	2	强制输出低电平

用户可以设置寄存器中的 SWFRCACT 位来强制或释放 PWM 输出。此外，用户可以调用 API 来产生与以下代码所示相同的效果。

```

/**
 * @brief Overrides the timer CCP output
 *
 * @param[in] gptimer    Pointer to the register overlay for the
 *                       peripheral
 * @param[in] out        Specifies the CCP output state.
 *                       @ref DL_TIMER_FORCE_OUT
 * @param[in] outComp    If timer instance supports complementary output,
 *                       it allows to override complementary out also.
 *                       If timer instance doesn't support complementary
 *                       output, this parameter is ignored.
 *                       @ref DL_TIMER_FORCE_CMPL_OUT
 * @param[in] ccIndex    Index associated to capture compare register
 *                       @ref DL_TIMER_CC_INDEX.
 */
void DL_Timer_overrideCCPOut(GPTIMER_Regs *gptimer, DL_TIMER_FORCE_OUT out,
    DL_TIMER_FORCE_CMPL_OUT outComp, DL_TIMER_CC_INDEX ccIndex);
    
```

7 用例 - 非对称 PWM

7.1 原理

相移控制广泛用于电机控制、电源等应用。仅在 TIMA 中，可以通过生成两个具有受控相移的同步中心对齐 PWM 信号来生成非对称 PWM。

要生成非对称 PWM，需要 TIMA 来提供相位加载。此外，交叉触发器用于同步计时器。以下是该用例的关键特性。

- 生成两个中心对齐 PWM
- 使用交叉触发器来同步两个 TIMx。
- 配置相位加载（仅限 TIMA）

7.2 实现

交叉触发可用于同步不同的计时器。通过将多个计时器连接在一起来使用主/次计时器配置时，交叉触发特性可以指示同一电源域或不同电源域中使用事件结构的多个计时器模块同时开始计数。

可以使用软件启用交叉触发，比较来自其他计时器实例的事件、零或加载事件或通用订阅者事件。某些应用可能需要多个计数器块，这些计数器块可在同一电源域（例如 TIMA0 和 TIMA1）或不同电源域（例如 TIMA0 和 TIMG0）上同时启动。如图 7-1 所示，TIMGx 是主计时器，TIMAx 是可在配置示例中交叉触发的次级计时器。

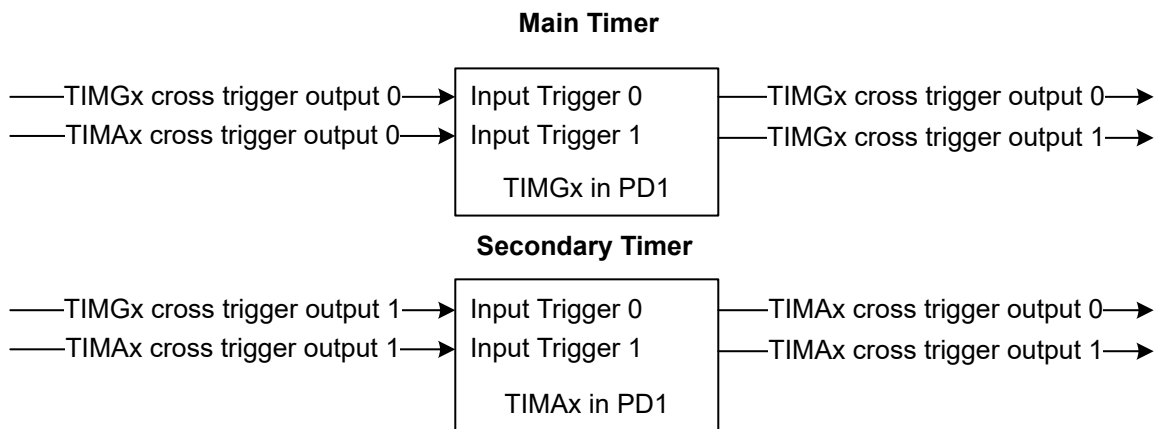


图 7-1. 电源域 1 中主计时器 (TIMGx) 和次级计时器 (TIMAx) 的交叉触发连接

通过使用交叉触发器，不同的计时器可以同时开始计数。为了产生受控相移，需要相位加载。（仅在 TIMA）相位加载寄存器使计数器能够在向上/向下计数模式下从零或负载值以外的值进行计数。相位加载用于生成非对称的中心对齐 PWM 输出信号，并在不同计时器实例之间控制相移。

当相位加载被触发时，计时器从相位加载值开始计数。当计时器启动时，相位加载值被锁存。每当计数器达到之前锁存的相位加载值时，相位加载都会同步。图 7-2 展示了当计时器向上-向下计数且相位加载值变为新值时的相位加载寄存器工作原理。

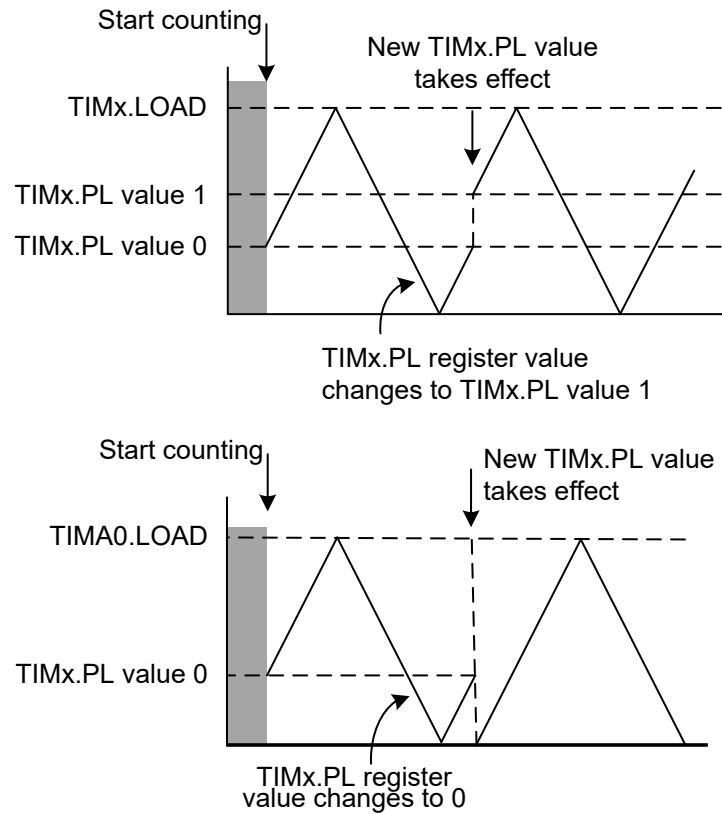


图 7-2. 向上-向下模式下的相位加载寄存器同步

作为使用 TIMA0 和 TIMA1 的 CCP 通道 0 的非对称 PWM 配置的示例，首先使用交叉触发器同步 TIMA0 和 TIMA1。使用具有相同加载值和比较值的 TIMA0 和 TIMA1 配置两个中心对齐 PWM，以生成相同的 PWM 频率和占空比。通过配置相位加载值，为 TIMA0 或 TIMA1 添加相移值。图 7-3 展示了非对称 PWM。

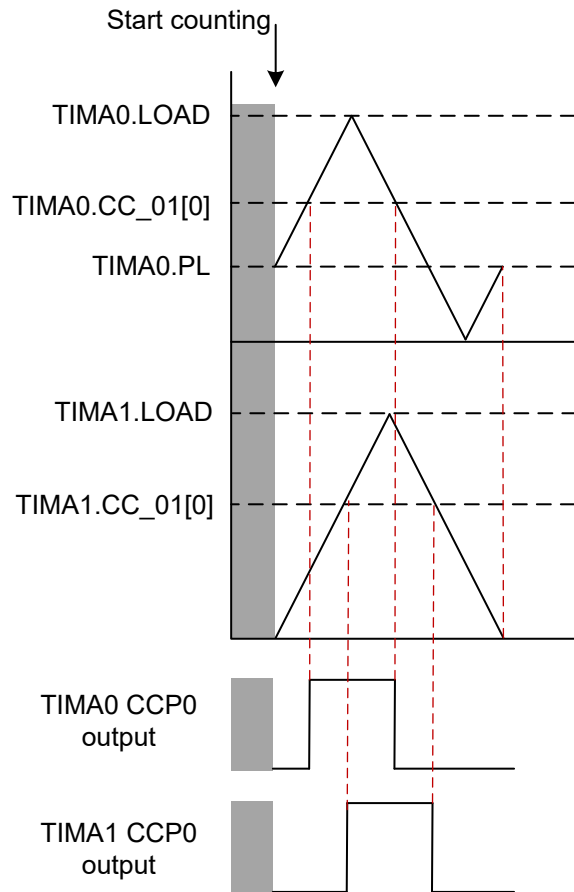


图 7-3. TIMA0 和 TIMA1 的 CCP 通道 0 具有相位加载的非对称 PWM 配置

若要加快开发速度，请参阅以下资源。

- SDK 中的示例：[timx_timer_mode_pwm_cross_trigger_stop_restore](#)

8 用例 – 使用重复计数器实现更佳的中断生成

8.1 原理

在仅 TIMA 中，重复计数器是一个 8 位计数器，它提供抑制不必要事件和生成真实事件的机制，以实现更佳的中断生成。具体来说，如果计时器正在生成针对已知周期数重复发生的事件（例如周期性 PWM 输出波形），则重复计数器可以抑制加载事件、比较事件和归零事件。这可防止在每个计时器周期内产生过多和不必要的中断。

要使用重复计数器生成更佳的中断，需要 TIMA。以下是该用例的关键特性。

- 生成中心对齐或边沿对齐 PWM
- 配置重复计数器（仅限 TIMA）

8.2 实现

当计时器计数器前进时，一旦计数器重新加载，重复计数器就会前进。用户可以设置在生成中断和事件之前重新加载的计时器计数器的数量。一旦重复计数器等于设置数量，重复计数器将被复位回零，并且中断和事件状态寄存器中会发生重复计数器清零事件 (REPC)。

此外，当 TIMA.RC 不等于零时，重复计数器提供抑制归零事件、加载事件和比较事件生成的功能。基于此，MCU 可以避免产生过多和不必要的中断。

如图 8-1 所示，计时器配置为向下计数模式，一旦计时器计数器 = 0，就会生成归零事件。重复计数器用于抑制归零和加载事件，直到发生 4 次计时器重新加载。用户可以配置重复计数器，以根据需要生成更佳的中断。

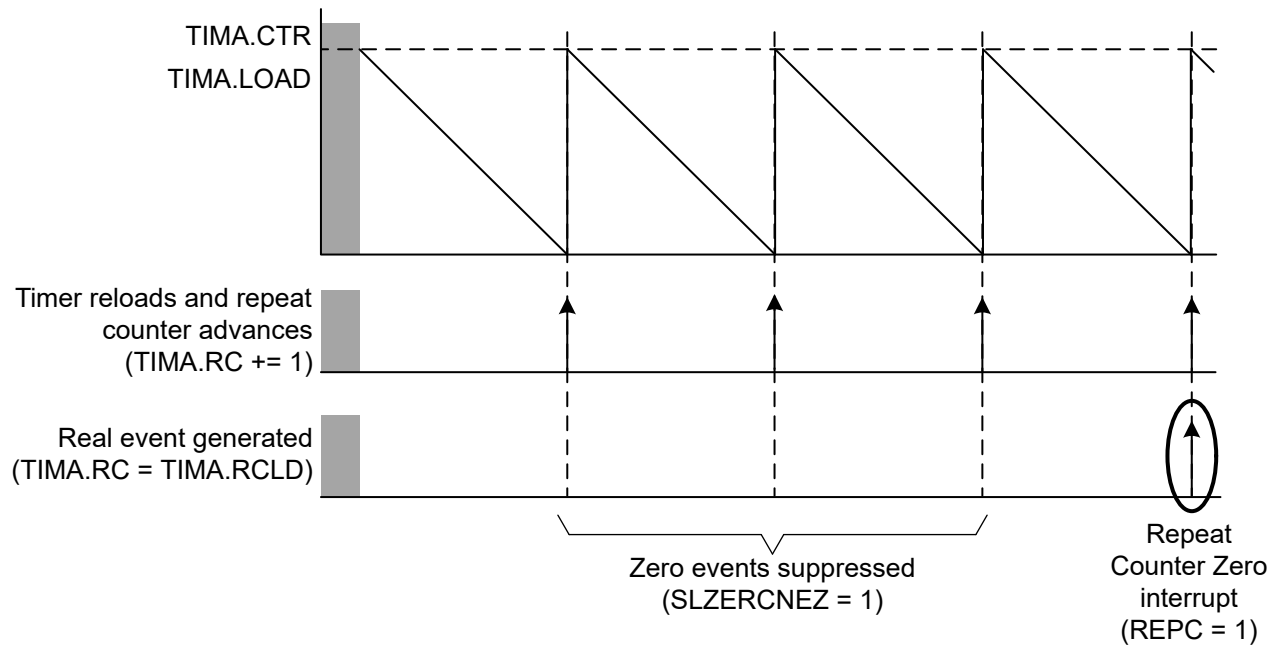


图 8-1. 重复计数器抑制的事件

若要加快开发速度，请参阅以下资源。

- SDK 中的示例：[tima_timer_mode_periodic_repeat_count](#)

9 总结

本应用手册主要介绍了高级控制计时器的一些典型应用场景。基于高级控制计时器，用户可实现更丰富的配置，以适应更多样化的使用场景。用户可以结合单个计时器的许多独特功能或同时使用多个计时器来实现更好的控制和更好的数字输出。

有关更多详细信息，请参阅特定器件的技术参考手册。

10 参考资料

- 德州仪器 (TI)，[下载 MSPM0 SDK](#)。
- 德州仪器 (TI)，[系统配置工具](#) 产品页面。
- 德州仪器 (TI)，[MSPM0 C 系列 24MHz 微控制器](#) 技术参考手册。
- 德州仪器 (TI)，[适用于 24MHz Arm® Cortex®-M0+ MCU 的 MSPMC1104 LaunchPad™ 开发套件](#)，LaunchPad 开发套件。
- 德州仪器 (TI)，[MSPM0 计时器 Academy](#)。

重要声明和免责声明

TI“按原样”提供技术和可靠性数据（包括数据表）、设计资源（包括参考设计）、应用或其他设计建议、网络工具、安全信息和其他资源，不保证没有瑕疵且不做任何明示或暗示的担保，包括但不限于对适销性、某特定用途方面的适用性或不侵犯任何第三方知识产权的暗示担保。

这些资源可供使用 TI 产品进行设计的熟练开发人员使用。您将自行承担以下全部责任：(1) 针对您的应用选择合适的 TI 产品，(2) 设计、验证并测试您的应用，(3) 确保您的应用满足相应标准以及任何其他功能安全、信息安全、监管或其他要求。

这些资源如有变更，恕不另行通知。TI 授权您仅可将这些资源用于研发本资源所述的 TI 产品的应用。严禁对这些资源进行其他复制或展示。您无权使用任何其他 TI 知识产权或任何第三方知识产权。您应全额赔偿因在这些资源的使用中对 TI 及其代表造成的任何索赔、损害、成本、损失和债务，TI 对此概不负责。

TI 提供的产品受 [TI 的销售条款](#) 或 [ti.com](#) 上其他适用条款/TI 产品随附的其他适用条款的约束。TI 提供这些资源并不会扩展或以其他方式更改 TI 针对 TI 产品发布的适用的担保或担保免责声明。

TI 反对并拒绝您可能提出的任何其他或不同的条款。

邮寄地址：Texas Instruments, Post Office Box 655303, Dallas, Texas 75265

Copyright © 2024，德州仪器 (TI) 公司