NPD3 模式下的 LPTPWM 控制呼吸灯

NuMicro® 32 位系列微控制器范例代码介绍

文件信息

应用简述	在 M2L31 系列微控制器中，低功耗定时器（LPTMR）在 PWM 模式下称为 LPTPWM，可以在正常模式和掉电模式中周期性地产生所需的输出波形。在本示例中，LPTPWM 利用此功能，在正常掉电模式 3（NPD3）中实现呼吸灯效果。此外，M2L31 微控制器可以根据需求从 NPD3 模式下被唤醒。
BSP 版本	M2L31_Series_BSP_CMSIS_V3.00.000
开发平台	NuMaker-M2L31KI V2.2

The information described in this document is the exclusive intellectual property of Nuvoton Technology Corporation and shall not be reproduced without permission from Nuvoton.

Nuvoton is providing this document only for reference purposes of NuMicro microcontroller and microprocessor based system design. Nuvoton assumes no responsibility for errors or omissions.

All data and specifications are subject to change without notice.

For additional information or questions, please contact: Nuvoton Technology Corporation.

www.nuvoton.com

1. 概述

M2L31 系列微控制器（MCU）支持特定外设在掉电模式下运行，包括 LPTMR、LPSPI、LPUART 等。在本示例中，LPTMR 以 PWM 模式（LPTPWM）运行，通过脉冲波形生成器在 NPD3 掉电模式下周期性地产生输出信号。

在 NPD3 模式下，LPTPWM 可以在指定的比较数据（CMPDAT）或周期数据（PERIOD）点唤醒系统。无论系统是否保持在 NPD3 模式或退出该模式，LPTPWM 的周期性输出脉冲均不受影响。

本示例中，输出信号驱动 LED 以实现呼吸灯效果。

1.1 原理

LPTPWM 的时钟源可以是 LXT 或 LIRC。当 CMPDAT 和 PERIOD 值分别写入 LPTMRx_PWMCMPDAT 和 LPTMRx_PWMPERIOD 寄存器后，LPTPWM 脉冲生成器利用这两个值，在向上计数模式下生成 PWM 输出脉冲。

在周期模式下，LPTPWM 计数器从零开始计数，其当前值存储在 LPTMRx_PWMCNT 寄存器中。一旦 LPTMRx_PWMCNT 值超过 PERIOD 值，计数器将复位并重新从零开始计数。

如图 1-1 所示，当 LPTMRx_PWMCNT 值小于设定的 CMPDAT 值时，LPTPWM 脉冲生成器将输出信号设为高电平状态。当 LPTMRx_PWMCNT 值达到或超过 CMPDAT 时，输出信号切换为低电平状态。

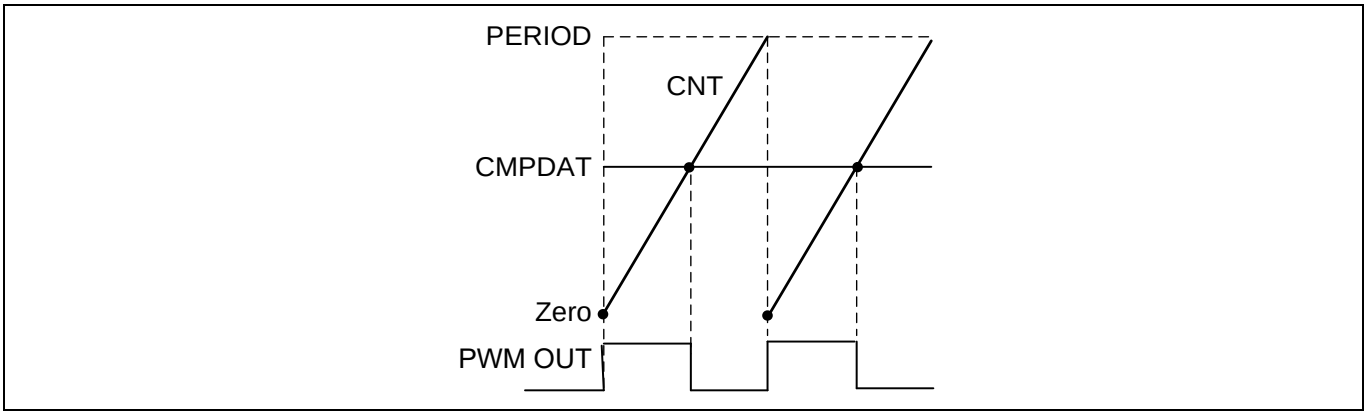

图 1-1 LPTPWM 脉冲生成器

当启用 PDCLKEN 寄存器（LPTMRx_CTL[16]）时，在 NPD3 模式下，LPTPWM 的内部计数器将继续运行，使脉冲生成器产生设定的波形。此外，如果设置并触发了 CMPDAT 或 PERIOD 唤醒事件，系统可以退出 NPD3 模式。

由于脉冲生成器的计数器始终保持活动状态，因此无论系统处于 NPD3 模式还是已被唤醒，周期性输出脉冲的频率都不会发生变化。

本示例演示了如何在散布-聚集模式（scatter-gather mode）下使用 LPPDMA 来修改 CMPDAT 设置，并在 PERIOD 事件发生时进行更新。由于 CMPDAT 会在 PERIOD 事件期间被更新，因此会影响 LPTPWM 脉冲生成器在下一个周期的输出占空比。占空比的任何变化都会改变 LED 的亮灯时间。

为了实现呼吸灯效果，本示例使用了两个 LPPDMA 散布-聚集表。其中一个表中 LED 的亮灯时间逐渐增加，而另一个表中亮灯时间逐渐减少。

1.2 执行结果

本示例代码将 M2L31 配置为 NPD3 模式。在 M2L31 NuMaker 开发板上，红色 LED 的阴极连接至 PC.14 引脚，该引脚被分配为 LPTM1 输出引脚。每次 LPTPWM 脉冲生成器完成一个周期时，LED 闪烁一次。当计数器值小于 CMPDAT 设定值时，红色 LED 保持熄灭；当计数器值达到 CMPDAT 设定值时，LED 点亮，并持续亮起直到计数器值等于或超过 PERIOD 设定值。

本示例代码提供了两个版本，并通过 OPT_WAKEUP 的定义来区分。在第一个版本中，若将 OPT_WAKEUP 声明设为零，M2L31 将保持在 NPD3 模式，并且 LPTPWM1 继续运行。在第二个版本中，若将 OPT_WAKEUP 声明设为非零值，则 M2L31 会在每个 LPTPWM1 的 CMPDAT 和 PERIOD 事件发生时唤醒，如图 1-2 所示。

在图 1-2 中，当 CMPDAT 或 PERIOD 事件发生时，M2L31 会被唤醒。一旦 M2L31 被唤醒，系统 HCLK 会重新启用。通道 1 显示来自 PA.3 的 HCLK 输出，而通道 2 显示来自 PC.14 的脉冲生成器输出。M2L31 唤醒后会简短地运行，然后重新进入 NPD3 模式。无论系统处于 NPD3 模式或已被唤醒，由 LPTPWM 产生的周期性输出脉冲保持不变。

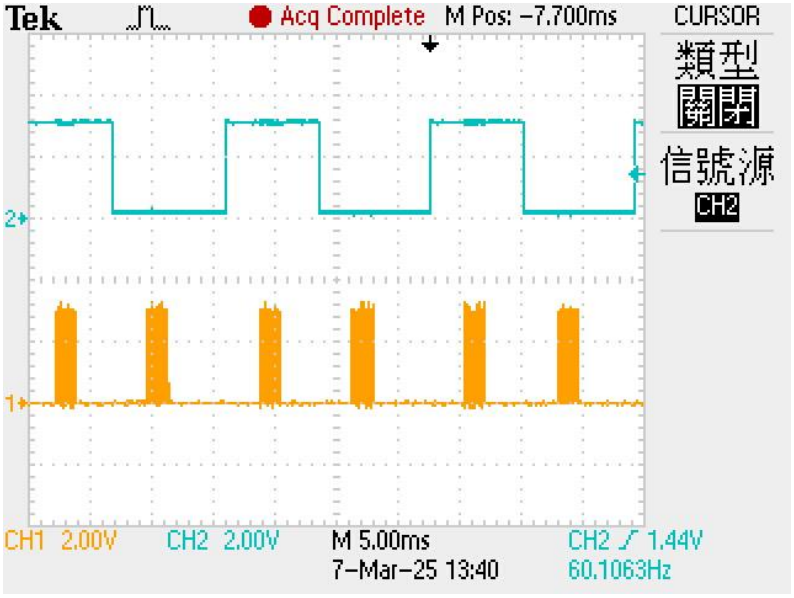

图 1-2 脉冲生成器输出和系统 HCLK 时钟信号

在图 1-3 中，调试窗口显示了 M2L31 分别被 CMPDAT 和 PERIOD 事件唤醒时的消息提示。

```
================================================
     LPTPWM pulse generator in NPD3 mode.
================================================

        Enter NPD3 mode
        Woken by CMPDAT 1 times!

        Woken by PERIOD 1 times!

        Enter NPD3 mode
        Woken by CMPDAT 2 times!

        Woken by PERIOD 2 times!
```

图 1-3 调试窗口

2. 代码介绍

本示例代码首先初始化了几个功能，包括一般系统功能、HCLK 时钟输出、LPTMR1 和 LPPDMA，分别使用 SYS_Init()、CLK_EnableCKO()、LPTPWM1_Init() 和 LPPDMA_Init() 函数进行初始化。在这些功能初始化后，启用 LPTPWM1 计数器以启动脉冲生成器。

在调用 LPPDMA_Init() 函数之前，必须正确设置两个数组缓冲区，用于 LPPDMA 的散布-聚集表。这些数组存储不同的 CMPDAT 设置：一个数组包含逐渐增加的值，另一个数组包含逐渐减少的值。

由于每次 LPPDMA 触发时 CMPDAT 的值变化为两个单位，为确保正确设置，每个数组的大小必须至少为 PERIOD 设置的一半。

```
int main(void)
{
    ...

    /* Init System, IP clock and multi-function I/O. */
    SYS_Init();
    ...
    /* Output selected clock to CKO, CKO Clock = HCLK / 2^(15+1) */
    CLK_EnableCKO(CLK_CLKSEL1_CLKOSEL_HCLK, 15, 1);

    /* Init LPTPWM1 */
    LPTPWM1_Init();

    /* Get the given value PERIOD register to be 0x214 */
    g_u32Period = LPTPWM_GET_PERIOD(LPTMR1);
    g_u32BufLength = g_u32Period/CMPDAT_VAR;

    /* Set two arrays for LPPDMA scatter-gather tables */
    for (i=0; i<g_u32BufLength; i++)
    {
        g_au16SrcArray0[i] = CMPDAT_VAR*i;
        g_au16SrcArray1[i] = g_u32Period - CMPDAT_VAR*i;
    }

    /* Init LPPDMA */
    LPPDMA_Init(g_u32BufLength);

    /* Start LPTPWM1 counter */
    LPTPWM_START_COUNTER(LPTMR1);

    ...
}
```

主循环

在 while(1){…} 主循环中，M2L31 首先进入 NPD3 模式，并在 OPT_WAKEUP 设置为非零值时等待被唤醒。当 M2L31 被唤醒后，代码会检查唤醒源是否为 CMPDAT 或 PERIOD 事件。

```
int main(void)
{
    ...

    while(1)
    {
        /* System enters NPD3 mode */
        printf("\tEnter NPD3 mode\n");
        while(!UART_IS_TX_EMPTY(UART0));
        PowerDown_Enter();

#if OPT_WAKEUP
        /* Check which is the wakeup source */
        if (g_u32IntFlag & LPTMR_PWMINTSTS0_CMPUIF_Msk)
        {
            g_u32IntFlag &= ~LPTMR_PWMINTSTS0_CMPUIF_Msk;
            printf("\tWoken by CMPDAT %d times!\n\n", ++i);
        }

        if (g_u32IntFlag & LPTMR_PWMINTSTS0_PIF_Msk)
        {
            g_u32IntFlag &= ~LPTMR_PWMINTSTS0_PIF_Msk;
            printf("\tWoken by PERIOD %d times!\n\n", ++j);
        }
        while(!UART_IS_TX_EMPTY(UART0));
#endif
    }

    ...
}
```

LPTPWM1 初始化

LPTPWM1_Init() 函数配置 LPTMR1 为 PWM 模式，并启用多个功能，包括设置脉冲生成器频率、配置 LPTMR1 输出引脚、启用 CMPDAT 和 PERIOD 事件中断、允许 LPTPWM1 唤醒、管理 LPTMR1 在掉电模式中的时钟控制，以及由 PERIOD 事件触发 LPPDMA。在完成这些设置后，LPTPWM1 计数器会在退出函数时启用。

```
void LPTPWM1_Init(void)
{
    /* Change LPTMR1 to PWM mode */
    LPTPWM_ENABLE_PWM_MODE(LPTMR1);

    /* Set LPTPWM1 TMx frequency to 60 Hz with duty 0% */
    LPTPWM_ConfigOutputFreqAndDuty(LPTMR1, 60, 0);
```

```
    /* Set LPTPWM1 output to LPTM1 pin */
    LPTPWM_SELECT_TOUT_PIN(LPTMR1, LPTPWM_TOUT_PIN_FROM_TX);

    /* Enable LPTPWM1 output */
    LPTPWM_ENABLE_OUTPUT(LPTMR1, 1);

#if OPT_WAKEUP
    /* Enable CMPDAT event interrupt */
    LPTPWM_ENABLE_CMP_UP_INT(LPTMR1);

    /* Enable PERIOD event interrupt */
    LPTPWM_ENABLE_PERIOD_INT(LPTMR1);

    /* Enable LPTPWM1 wake-up function */
    LPTPWM_EnableWakeup(LPTMR1);

    /* Enable LPTPWM1 interrupt */
    NVIC_EnableIRQ(LPTMR1_IRQn);
#endif

    /* Enable LPTPWM1 clock in NPD3 mode */
    LPTMR_EnablePDCLK(LPTMR1);

    /* Enable LPTPWM1 PERIOD event to trigger LPPDMA */
    LPTPWM_EnableTriggerLPPDMA(LPTMR1, LPTPWM_TRIGGER_AT_PERIOD_POINT);
}
```

LPPDMA 初始化

LPPDMA_Init() 函数配置了两个散布-聚集表来控制 LPTPWM 脉冲生成器。每当 PERIOD 事件发生时，LPTPWM1 触发 LPPDMA 将一个 16 位数据值从指定的数组缓冲区传送到 LPTMR1_PWMCMPDAT 寄存器。在下一个脉冲生成器周期中，更新后的 CMPDAT 值生效，并相应地调整占空比。

```
void LPPDMA_Init(uint32_t g_u32BufLength)
{
    LPPDMA_Open(LPPDMA0,1 << LPPDMA_CH);
    DMA_DESC[0].u32Ctl = LPPDMA_OP_SCATTER |
                         LPPDMA_REQ_SINGLE |
                         LPPDMA_SAR_INC |           /* source address -> incremented */
                         LPPDMA_DAR_FIX |           /* destination address -> fixed  */
                         LPPDMA_WIDTH_16 |          /* transfer width -> 16-bit      */
                         LPPDMA_TBINTDIS_DISABLE |  /* Table Interrupt -> disabled   */
                         ((g_u32BufLength - 1) << LPPDMA_DSCT_CTL_TXCNT_Pos);
    DMA_DESC[0].u32Src = (uint32_t)g_au16SrcArray0;
    DMA_DESC[0].u32Dest = (uint32_t)&(LPTMR1->PWMCMPDAT);
    DMA_DESC[0].u32Offset = (uint32_t)&DMA_DESC[1] - (PDMA0->SCATBA);

    DMA_DESC[1].u32Ctl = LPPDMA_OP_SCATTER |
                         LPPDMA_REQ_SINGLE |
                         LPPDMA_SAR_INC |           /* source address -> incremented */
                         LPPDMA_DAR_FIX |           /* destination address -> fixed  */
```

```
                            LPPDMA_WIDTH_16 |           /* transfer width -> 16-bit      */
                            LPPDMA_TBINTDIS_DISABLE |   /* Table Interrupt -> disabled   */
                            ((g_u32BufLength - 1) << LPPDMA_DSCT_CTL_TXCNT_Pos);
    DMA_DESC[1].u32Src = (uint32_t)g_au16SrcArray1;
    DMA_DESC[1].u32Dest = (uint32_t)&(LPTMR1->PWMCMPDAT);
    DMA_DESC[1].u32Offset = (uint32_t)&DMA_DESC[0] - (PDMA0->SCATBA);

    LPPDMA_SetTransferMode(LPPDMA0, LPPDMA_CH, LPPDMA_LPTMR1, 1, (uint32_t)&DMA_DESC[0]);
}
```

LPTMR1 中断处理程序

若将 OPT_WAKEUP 声明设为非零值，当 CMPDAT 或 PERIOD 事件发生且相应的中断被启用时，M2L31 可从 NPD3 模式被唤醒。此时，LPTMR1 中断服务程序（ISR）会首先被触发。

在 M2L31 准备进入下一次掉电-唤醒循环时，必须在 ISR 内清除所有唤醒、CMPDAT 和 PERIOD 事件标志，以确保 M2L31 能够再次被正确唤醒。

```
void LPTMR1_IRQHandler(void)
{
    /* Clear LPTPWM1 wakeup flag */
    LPTPWM_ClearWakeupFlag(LPTMR1);

    /* Check if system woken up by CMPDAT event */
    if (LPTPWM_GET_CMP_UP_INT_FLAG(LPTMR1))
    {
        /* Clear CMPDAT interrupt flag */
        LPTPWM_CLEAR_CMP_UP_INT_FLAG(LPTMR1);

        g_u32IntFlag |= LPTMR_PWMINTSTS0_CMPUIF_Msk;
    }

    /* Check if system woken up by PERIOD event */
    if (LPTPWM_GET_PERIOD_INT_FLAG(LPTMR1))
    {
        /* Clear PERIOD interrupt flag */
        LPTPWM_CLEAR_PERIOD_INT_FLAG(LPTMR1);

        g_u32IntFlag |= LPTMR_PWMINTSTS0_PIF_Msk;
    }
}
```

3. 软件与硬件需求

3.1 软件需求

- BSP 版本
  - M2L31_Series_BSP_CMSIS_V3.01.000
- IDE 版本
  - Keil uVersion 5.38.0.0

3.2 硬件需求

- 电路组件
  - NuMaker- M2L31KI V2.2
- 线路示意图
  - LEDG1 指示电源状态。
  - 将 PC.14 连接到 LEDR1 的阴极。

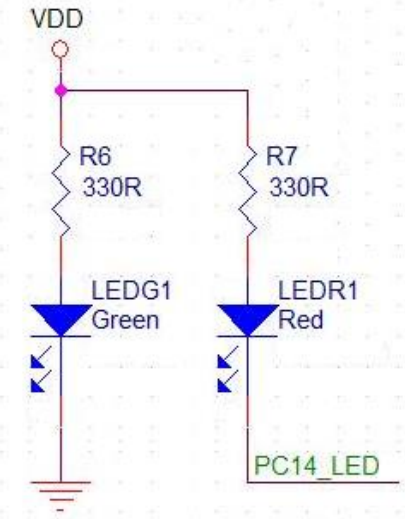

图 3-1 引脚连接图

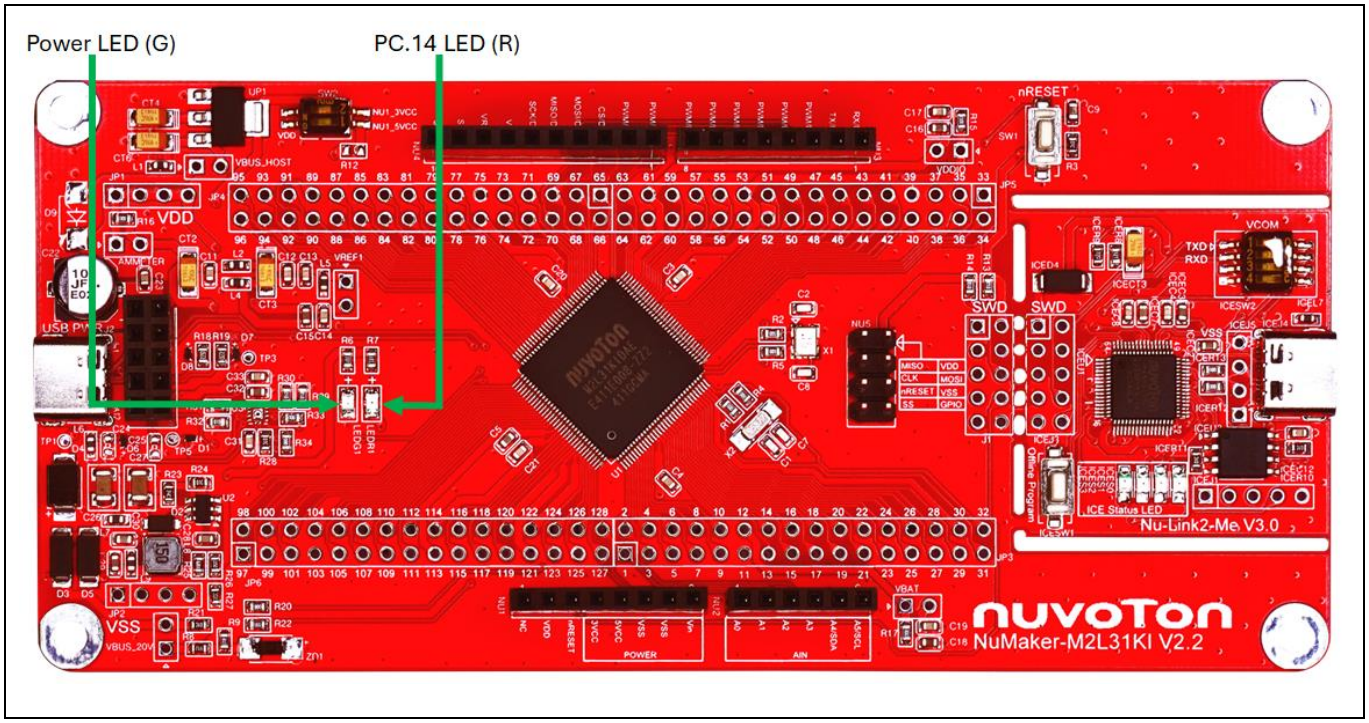

图 3-2 NuMaker M2L31KI V2.2

4. 目录信息

目录结构如图 4-1 所示。

EC_ M2L31_LPTPWM_BreathingLight_NPD3_V1.00	
Library	Sample code header and source files
CMSIS	Cortex® Microcontroller Software Interface Standard (CMSIS) by Arm® Corp.
Device	CMSIS compliant device header file
StdDriver	All peripheral driver header and source files
SampleCode	
ExampleCode	
Proect	Source file of example code

图 4-1 目录信息

5. 范例程序执行

1. 根据目录信息章节进入 ExampleCode 路径中的 KEIL 文件夹，双击 *EC_M2L31_LPTPWM_BreathingLight_NPD3.uvprojx*。
2. 进入编译模式界面
   - 编译
   - 下载代码至内存
   - 进入 / 离开仿真模式
3. 进入仿真模式界面
   - 执行代码

6. 修订纪录

Date	Revision	Description
2025.03.10	1.00	初始发布。

Important Notice

Nuvoton Products are neither intended nor warranted for usage in systems or equipment, any malfunction or failure of which may cause loss of human life, bodily injury or severe property damage. Such applications are deemed, "Insecure Usage".

Insecure usage includes, but is not limited to: equipment for surgical implementation, atomic energy control instruments, airplane or spaceship instruments, the control or operation of dynamic, brake or safety systems designed for vehicular use, traffic signal instruments, all types of safety devices, and other applications intended to support or sustain life.

All Insecure Usage shall be made at customer's risk, and in the event that third parties lay claims to Nuvoton as a result of customer's Insecure Usage, customer shall indemnify the damages and liabilities thus incurred by Nuvoton.

Please note that all data and specifications are subject to change without notice.
All the trademarks of products and companies mentioned in this datasheet belong to their respective owners.