NPD3 模式下的 LPTPWM 控制呼吸燈

NuMicro® 32 位元系列微控制器範例程式碼介紹

文件訊息

應用簡述	在 M2L31 系列微控制器中，低功耗定時器 (LPTMR) 在 PWM 模式下稱之為 LPTPWM，可於正常模式和省電模式中週期性產生所需的輸出波形。本範例中，LPTPWM 利用此功能，在正常省電模式 3 (NPD3) 中實現呼吸燈效果。此外，M2L31 可根據需求從 NPD3 模式下被喚醒。
BSP 版本	M2L31_Series_BSP_CMSIS_V3.00.000
開發平臺	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）支持特定周邊在省電模式下運行，包括 LPTMR、LPSPI、LPUART 等。在本範例中，LPTMR 以 PWM 模式（LPTPWM）運行，通過脈衝波形產生器在 NPD3 省電模式下週期性地產生輸出信號。

在 NPD3 模式下，LPTPWM 可以在指定的比較數據（CMPDAT）或週期數據（PERIOD）點喚醒系統。無論系統是否保持在 NPD3 模式或退出該模式，LPTPWM 的週期性輸出脈衝均不受影響。

本範例中，輸出信號驅動 LED 以實現呼吸燈效果。

1.1 原理

LPTPWM 的時鐘源可以是 LXT 或 LIRC。當 CMPDAT 和 PERIOD 值分別寫入 LPTMRx_PWMCMPDAT 和 LPTMRx_PWMPERIOD 暫存器後，LPTPWM 脈衝生成器利用這兩個值，在向上計數模式下生成 PWM 輸出脈衝。

在週期模式下，LPTPWM 計數器從零開始計數，其當前值儲存在 LPTMRx_PWMCNT 暫存器中。一旦 LPTMRx_PWMCNT 值超過 PERIOD 值，計數器將復位並重新從零開始計數。

如圖 1-1 所示，當 LPTMRx_PWMCNT 值小於設定的 CMPDAT 值時，LPTPWM 脈衝產生器將輸出信號設為高電位狀態。當 LPTMRx_PWMCNT 值達到或超過 CMPDAT 時，輸出信號切換為低電位狀態。

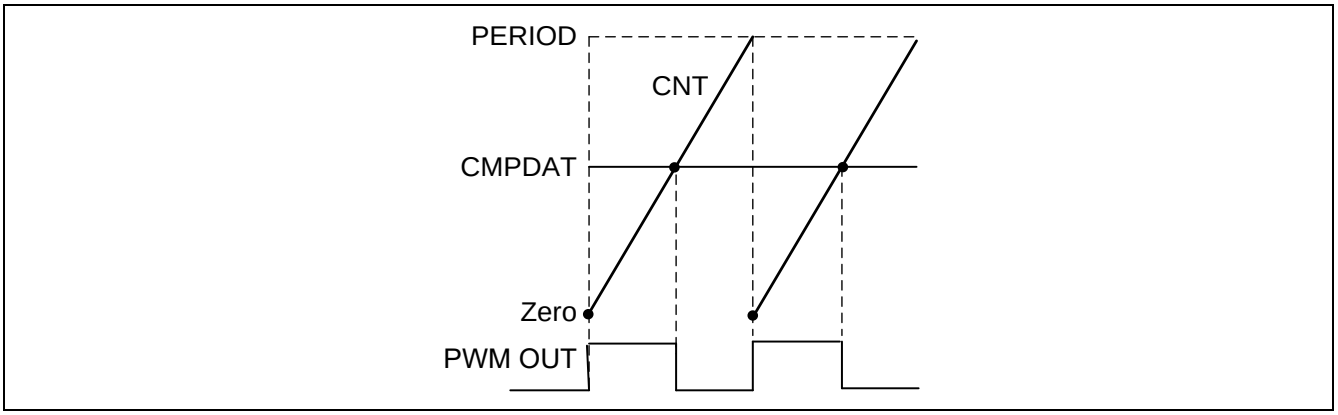

圖 1-1 LPTPWM 脈衝產生器

當啟用 PDCLKEN 暫存器（LPTMRx_CTL[16]）時，在 NPD3 模式下，LPTPWM 的內部計數器將繼續運行，使脈衝產生器輸生設定的波形。此外，如果設置並觸發了 CMPDAT 或 PERIOD 喚醒事件，系統可以退出 NPD3 模式。

由於脈衝生成器的計數器始終保持活動狀態，因此無論系統處於 NPD3 模式還是已被喚醒，週期性輸出脈衝的頻率都不會發生變化。

本範例演示了如何在散佈-聚集模式（scatter-gather mode） 下使用 LPPDMA 來修改 CMPDAT 設定，當 PERIOD 事件發生時進行更新。由於 CMPDAT 在 PERIOD 事件期間會被更新，因此會影響 LPTPWM 脈衝產生器在下一個週期的輸出佔空比。結果，佔空比的任何變化都會改變 LED 的亮燈時間。

為實現呼吸燈效果，使用了兩個 LPPDMA 散佈-聚集表。在一個表中，LED 的亮燈時間逐漸增加，而在另一個表中，亮燈時間逐漸減少。

1.2 執行結果

本範例程式碼將 M2L31 配置為 NPD3 模式。在 M2L31 NuMaker 開發板上，紅色 LED 的陰極連接至 PC.14 腳位，該腳位被分配為 LPTM1 輸出腳位。每次 LPTPWM 脈衝產生器完成一個週期時，LED 閃爍一次。當計數器值小於 CMPDAT 設定值時，紅色 LED 保持熄滅；當計數器值達到 CMPDAT 設定值時，LED 點亮，並持續亮起直到計數器值等於或超過 PERIOD 設定值。

本範例程式碼提供了兩個版本，並通過 OPT_WAKEUP 的定義來區分。在第一個版本中，若將 OPT_WAKEUP 宣告設為零，M2L31 將保持在 NPD3 模式，並且 LPTPWM1 繼續運行。在第二個版本中，若將 OPT_WAKEUP 宣告設為非零值，則 M2L31 會在每個 LPTPWM1 的 CMPDAT 和 PERIOD 事件發生時喚醒，如圖 1-2 所示。

在圖 1-2 中，當 CMPDAT 或 PERIOD 事件發生時，M2L31 會被喚醒。一旦 M2L31 被喚醒，系統 HCLK 會重新啟用。通道 1 顯示來自 PA.3 的 HCLK 輸出，而通道 2 顯示來自 PC.14 的脈衝產生器輸出。M2L31 喚醒後會簡短地運行，然後重新進入 NPD3 模式。無論系統處於 NPD3 模式或已被喚醒，由 LPTPWM 產生的週期性輸出脈衝保持不變。

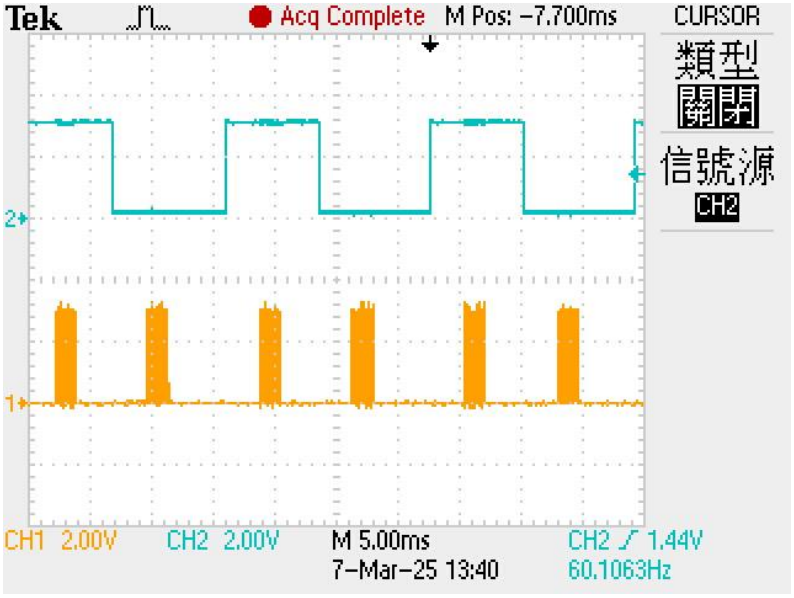

圖 1-2 脈衝產生成器輸出和系統 HCLK 時鐘信號

在圖 1-3 中，調試窗口顯示了 M2L31 分別被 CMPDAT 和 PERIOD 事件喚醒時的消息提示。

```
==========================================
    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!
```

圖 1-3 調試窗口

2. 程式碼介紹

本範例程式碼首先初始化了幾個功能，包括一般系統功能、HCLK 時鐘輸出、LPTMR1 和 LPPDMA，分別使用 SYS_Init()、CLK_EnableCKO()、LPTPWM1_Init() 和 LPPDMA_Init() 函數進行初始化。在這些功能初始化後，啟用 LPTPWM1 計數器以啟動脈衝產生器。

在呼叫 LPPDMA_Init() 函數之前，必須正確設置兩個陣列緩衝區，用於 LPPDMA 的散佈-聚集表。這些陣列儲存不同的 CMPDAT 設定：一個陣列包含逐漸增加的值，另一個陣列包含逐漸減少的值。

由於每次 LPPDMA 觸發時 CMPDAT 的值變化為兩個單位，為確保正確設置，每個陣列的大小必須至少為 PERIOD 設置的一半。

```
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);

    ...
}
```

主循環

在 while(1){…} 主循環中，M2L31 首先進入 NPD3 模式，並在 OPT_WAKEUP 設為非零值時等待被喚醒。當 M2L31 被喚醒後，程式碼會檢查喚醒源是否為 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() 函數配置 LPTMR1 為 PWM 模式，並啟用多項功能，包括設置脈衝產生器頻率、配置 LPTMR1 輸出引腳、啟用 CMPDAT 和 PERIOD 事件中斷、允許 LPTPWM1 喚醒、管理 LPTMR1 在掉電模式中的時鐘控制，以及由 PERIOD 事件觸發 LPPDMA。在應用這些設定後，LPTPWM1 計數器會在退出函數時啟用。

```
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() 函數配置了兩個散佈-聚集表來控制 LPTPWM 脈衝產生器。每當 PERIOD 事件發生時，LPTPWM1 觸發 LPPDMA 將一個 16 位元數據值從指定的陣列緩衝區傳送到 LPTMR1_PWMCMPDAT 暫存器。在下一個脈衝產生器週期中，更新後的 CMPDAT 值生效，並相應地調整佔空比。

```
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]);
}
```

LPTPWM1 中斷處理程序

若將 OPT_WAKEUP 宣告設為非零值，當 CMPDAT 或 PERIOD 事件發生且相應的中斷被啟用時，M2L31 可從 NPD3 模式被喚醒。此時，LPTMR1 中斷服務程序 (ISR) 會首先被觸發。

在 M2L31 準備進入下一次省電-喚醒循環時，必須在 ISR 內清除所有喚醒、CMPDAT 和 PERIOD 事件旗標，以確保 M2L31 能夠再次被正確喚醒。

```
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. 軟體與硬體需求

3.1 軟體需求

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

3.2 硬體需求

- 電路元件
  - NuMaker- M2L31KI V2.2
- 線路示意圖
  - LEDG1 指示電源狀態。
  - 將 PC.14 連接到 LEDR1 的陰極。

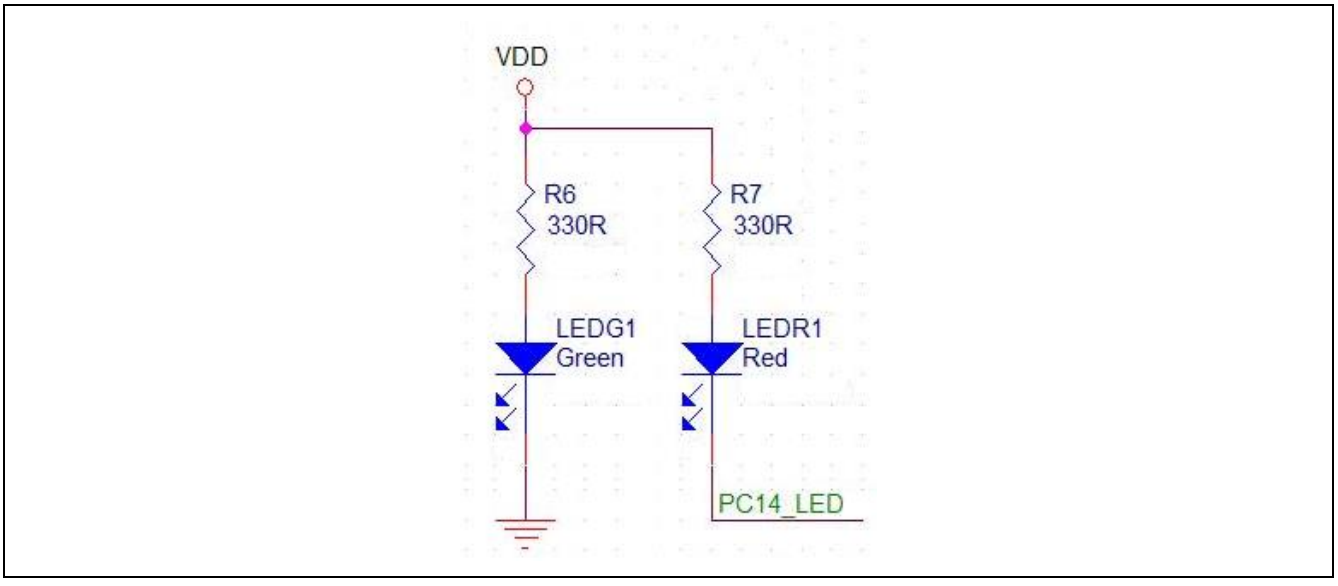

圖 3-1 腳位連接圖

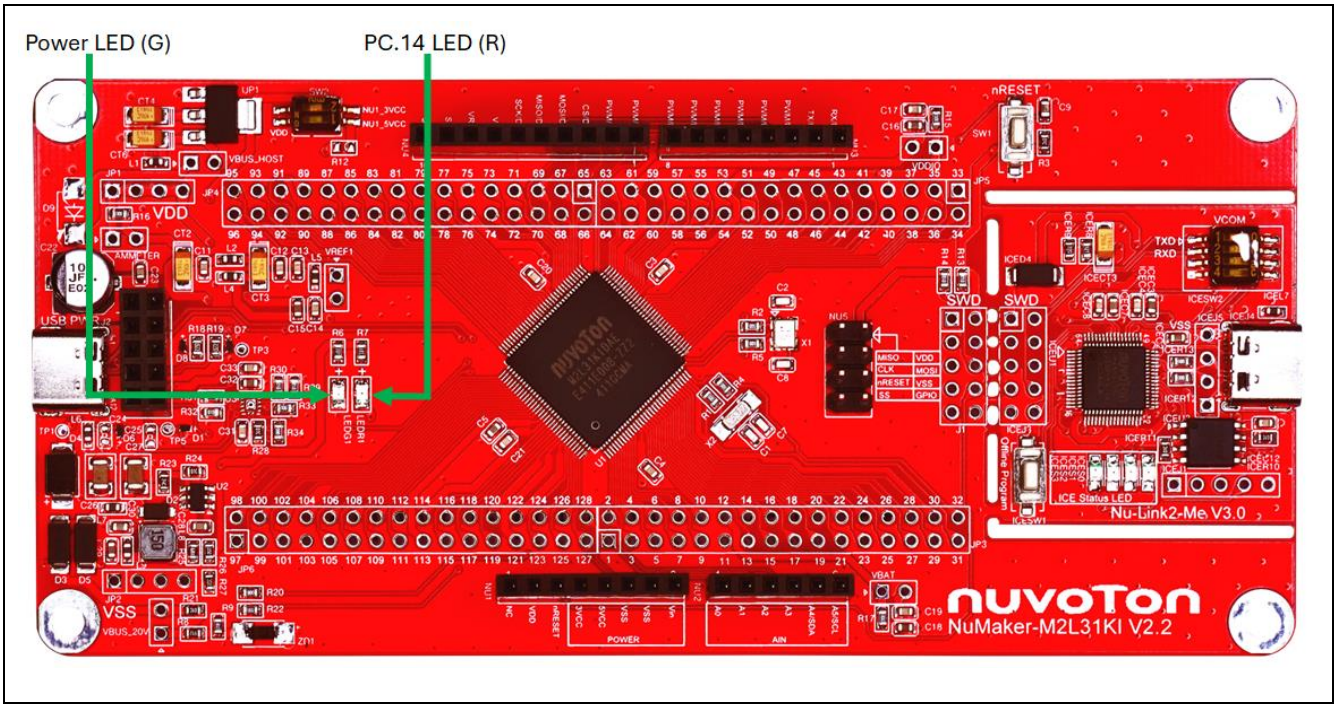

圖 3-2 NuMaker M2L31KI V2.2

4. 目錄資訊

目錄結構如圖 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	
Project	Source file of example code

圖 4-1 目錄資訊

5. 範例程式執行

1. 根據目錄資訊章節進入 ExampleCode 路徑中的 KEIL 資料夾，雙擊 *EC_M2L31_LPTPWM_BreathingLight_NPD3.uvprojx*。
2. 進入編譯模式介面
   - 編譯
   - 下載程式碼至記憶體
   - 進入 / 離開模擬模式
3. 進入模擬模式介面
   - 執行程式碼

6. 修訂紀錄

Date	Revision	Description
2025.03.10	1.00	初始發佈。

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.