低功耗模式下的脈衝計數

NuMicro® 32 位系列微控制器範例代碼介紹

文件資訊

應用簡述	本範例程式使用 TIMER0 來計數脈衝的數量，並使用 RTC 來喚醒 MCU。當 MCU 被喚醒後，它會輸出計數到的脈衝數量。
BSP 版本	251_M252_M254_M256_M258_Series_BSP_CMSIS_V3.02.006
開發平台	NuMaker_M251KG_V1.1

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

本範例代碼先設定了 RTC 和 TIMER0，然後將 MCU 切換到低功耗模式(Power-down mode)。TIMER0 會開始計算 PB5 上的脈衝訊號，直到 RTC 喚醒 MCU 後，停止計數並且將脈衝的數量輸出到電腦。

1.1 原理

圖 1-1 的流程圖可以幫助你在低功耗模式下實現脈衝計數。

1. 系統初始化

   確保 HIRC、LXT、UART0、RTC 和 TIMER0 都已經啟用，並將 PB5 設定為脈衝計數輸入腳位。

2. TIMER0 設定

   啟用 TIMER0 的事件計數模式，並且開始計數脈衝。

3. RTC 設定

   設定當前 RTC 時間，並且設定 MCU 的喚醒時間後，啟用 RTC 中斷。

4. 低功耗模式

   將 MCU 置於低功耗模式，並等待 RTC 喚醒。在這種模式下，大多數時脈都會被禁用，僅保留 LIRC 和 LXT 處於活動狀態。此外，配置為使用 LIRC 或 LXT 的周邊設備時脈（如 RTC、WDT、計時器和 UART）將保持活動狀態，以確保它們在低功耗模式下正常運行。

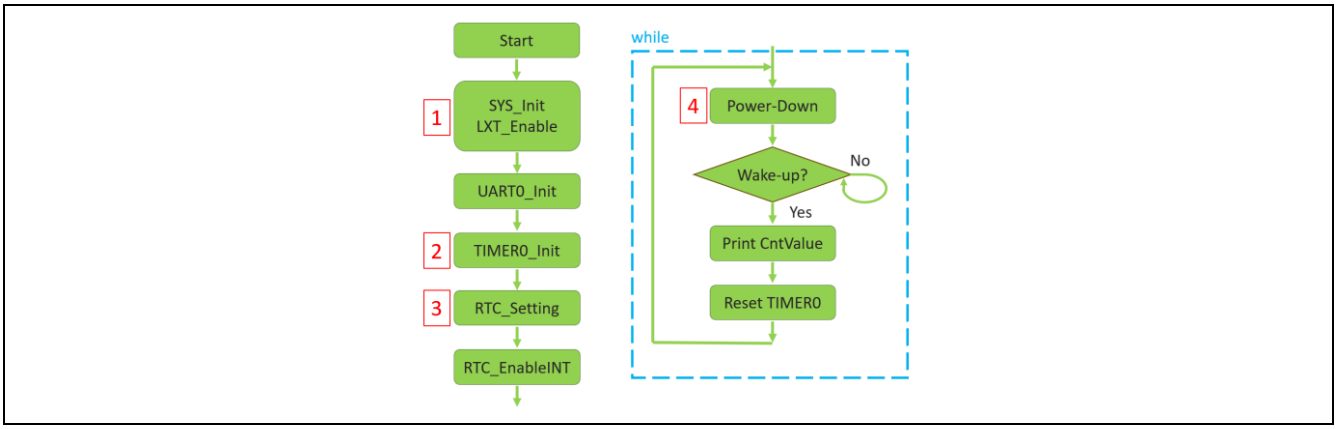

圖 1-1 程式流程圖

1.2 執行結果

最大輸入脈衝頻率應該是 LIRC 頻率的三分之一。如果啟用了 TMx 除彈跳功能（CNTDBEN (TIMERx_EXTCTL[7]) = 1），則最大輸入脈衝頻率應該是 LIRC 頻率的八分之一。然而，為了穩定性和安全性，我們建議最大輸入脈衝頻率為 LIRC 頻率的十分之一。

將 200 Hz 方波發生器信號連接到 PB5 接腳和 GND 接腳後執行此範例程式。執行結果可以參見圖 1-2 和圖 1-3。

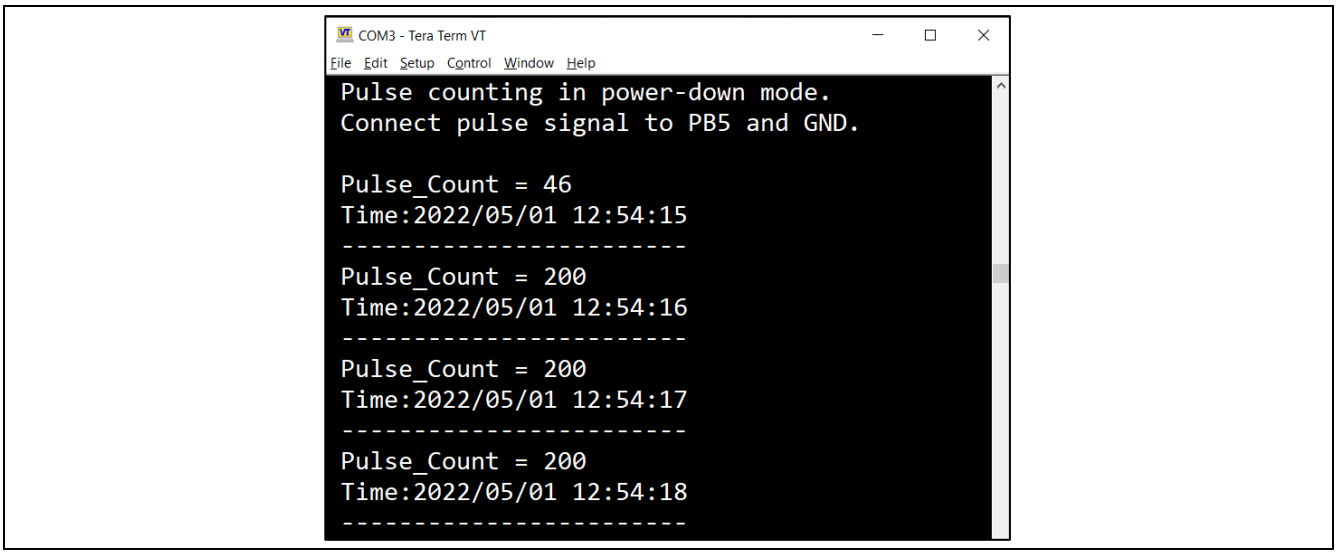

圖 1-2 執行結果

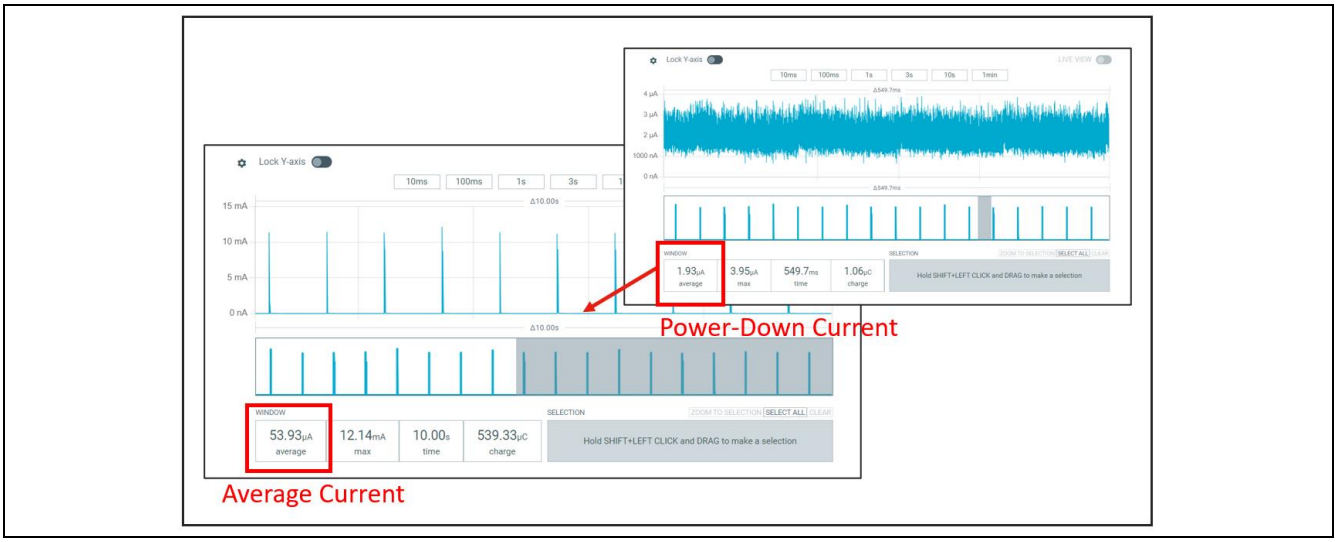

圖 1-3 消耗電流

2. 代碼介紹

初始化系統頻率、TIMER0 和 UART0 後。再設定 RTC 的現在日期和時間，然後設定 MCU 的喚醒時間。最後啟用 RTC 的中斷執行功能。

```
int32_t main(void)
{
    /* Init System, peripheral clock */
    SYS_Init();

    /* Unlock protected registers */
    SYS_UnlockReg();

    /* Init GPIO */
    Set_All_GPIO_to_Quasi_Mode();

    /* Disable POR */
    SYS_DISABLE_POR();

    /* Disable LIRC */
    CLK_DisableXtalRC(CLK_PWRCTL_LIRCEN_Msk);
    TIMER_Delay(TIMER0, 100000);

   /* Init TIMER0 */
   TIMER0_Init();

   /* Init UART0 for printf */
   UART0_Init();

   /* Lock protected registers */
   SYS_LockReg();

   printf(" Pulse counting in power-down mode.\n");
   printf(" Connect pulse signal to PB5 and GND.\n\n");

   /* Time setting */
   sInitTime.u32Year       = 2022;
   sInitTime.u32Month      = 5;
   sInitTime.u32Day        = 1;
   sInitTime.u32Hour       = 12;
   sInitTime.u32Minute     = 0;
   sInitTime.u32Second     = 0;
   sInitTime.u32DayOfWeek  = RTC_TUESDAY;
   sInitTime.u32TimeScale  = RTC_CLOCK_24;

   /* check rtc reset status */
   sInitTime_ptr = (RTC->INIT & RTC_INIT_ACTIVE_Msk) ? NULL : &sInitTime;

   if (RTC_Open(sInitTime_ptr) != 0)
   {
       printf("\n RTC initial fail!!");
       printf("\n Please check h/w setting!!");

       while (1);
   }

   /* Get the current time */
   RTC_GetDateAndTime(&sCurTime);

   /* Wait the UART FIFO buffer is empty */
   while (!IsDebugFifoEmpty()) {};
```

```
    /* Enable RTC Alarm Interrupt */
    RTC_EnableInt(RTC_INTEN_ALMIEN_Msk);
    NVIC_EnableIRQ(RTC_IRQn);
    NVIC_EnableIRQ(PWRWU_IRQn);
```

進入 while 迴圈後，先讓 MCU 進入低功耗模式（power-down mode），並等待實時時鐘（RTC）喚醒 MCU。當 RTC 喚醒 MCU 並輸出脈衝數量後，重置 TIMER0 的計數並設定下一次的喚醒時間。完成一次計數後，回到 while 迴圈以開始新的循環。這樣，MCU 將持續進行計數和低功耗管理的循環。

```
    while (1)
    {
        /* Set the alarm time for wake-up */
        Set_Alarm_time(User_Config.Wakeup_AlarmTime);

        g_i32Alarm = FALSE;

        /* Select mode to enter power down */
        EnterToPowerDown(User_Config.Power_Down_Mode);

        /* IOCTLSEL will automatically be set by hardware to 1 when system power is off
           and RTC_GPIOCTL0 had been configured. */
        /* when system wake up from PD mode, you need clear this bit if you want PF.4, 5,
           6 pin I/O function is controlled by GPIO module. */
        RTC->LXTCTL &= ~(RTC_LXTCTL_IOCTLSEL_Msk);

        /* Wait RTC interrupt occurred*/
        while (!g_i32Alarm) {};

        /* Get the current time */
        RTC_GetDateAndTime(&sCurTime);

        printf(" Time:%u/%02u/%02u %02u:%02u:%02u\n", sCurTime.u32Year,
                 sCurTime.u32Month, sCurTime.u32Day, sCurTime.u32Hour, sCurTime.u32Minute,
                 sCurTime.u32Second);
        printf(" ------------------------\n");

        /* Wait the UART FIFO buffer is empty */
        while (!IsDebugFifoEmpty()) {};
    }
}
```

一旦達到 RTC 設定的時間，會執行 RTC 中斷服務程序，讀取由 TIMER0 計算的脈衝數，並透過 UART0 將其輸出到電腦。

```
void RTC_IRQHandler(void)
{
    uint32_t Pulse_Count;

    /* alarm interrupt occurred */
```

```
    if ((RTC->INTEN & RTC_INTEN_ALMIEN_Msk) && RTC_GET_ALARM_INT_FLAG())
    {
         g_i32Alarm = TRUE;

        RTC_CLEAR_ALARM_INT_FLAG();

        /* Select HIRC as the clock source of Timer0 */
        Pulse_Count = TIMER_GetCounter(TIMER0);
         TIMER_ResetCounter(TIMER0);
        printf(" Pulse_Count = %d\n", Pulse_Count);
    }
}
```

3. 軟體與硬體需求

3.1 軟體需求

- BSP 版本
  - M251_M252_M254_M256_M258_Series_BSP_CMSIS_V3.02.006
- IDE 版本
  - Keil uVersion 5.36

3.2 硬體需求

- 電路元件
  - NuMaker_M251KG_V1.1
- 線路示意圖
  - 將訊號發生器的脈衝信號連接至 NuMaker-M251KG 的 PB5 腳位，並將 USB 連接至電腦如圖 3-1。

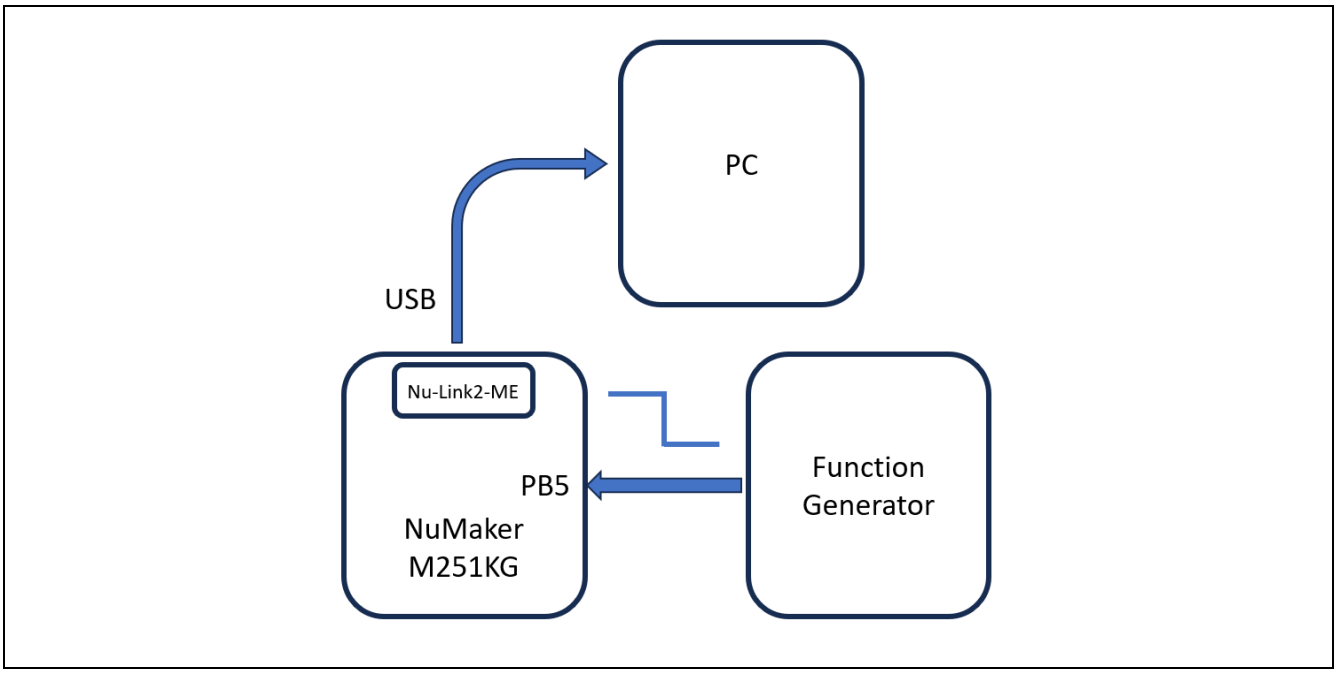

圖 3-1 線路示意圖

4. 目錄資訊

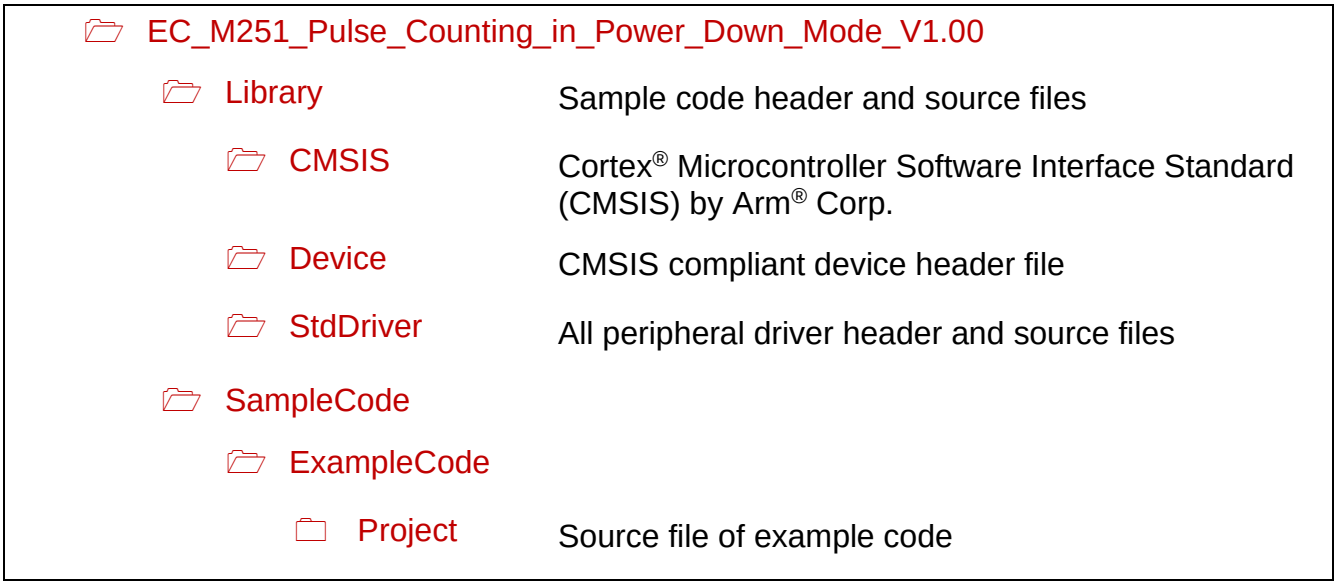

圖 4-1 目錄資訊

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 *Pulse_Counting_in_Power_Down_Mode.uvprojx*。
2. 進入編譯模式介面
   - 編譯
   - 下載代碼至記憶體
   - 進入 / 離開除錯模式
3. 進入除錯模式介面
   - 執行代碼
4. 超級終端機畫面如圖 5-1

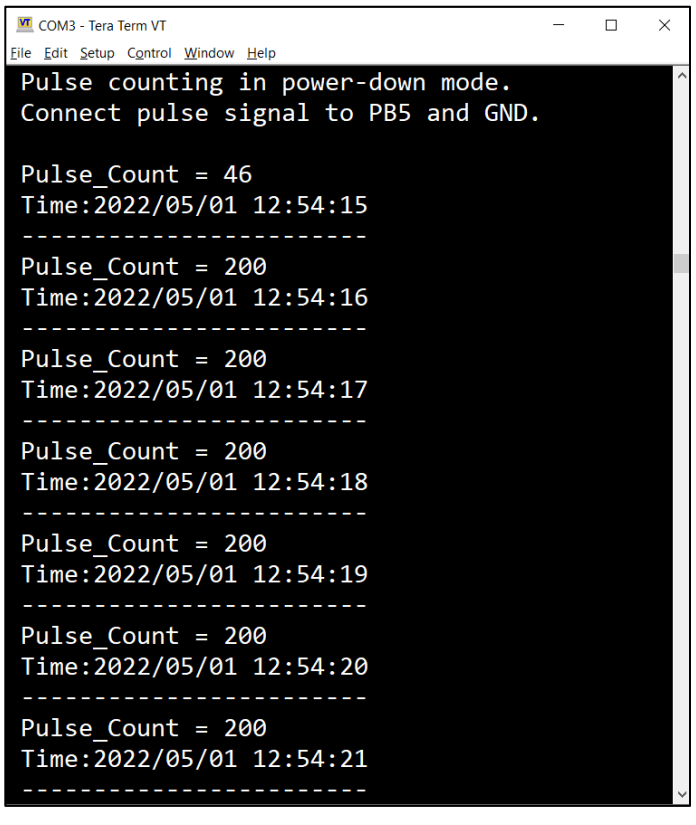

圖 5-1 顯示計數數量

6. 修訂紀錄

Date	Revision	Description
2024.06.13	1.00	初始發布。

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.