M030G 快速脈衝整形輸出範例

NuMicro® 32 位系列微控制器範例代碼介紹

文件資訊

應用簡述	本範例代碼使用 M030G 實現快速脈衝整形輸出
BSP 版本	M029G_M030G_M031G_Series_BSP_CMSIS_V3.03.000
開發平台	NuMaker-M030GTD 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

有些應用中，需要將不規則的輸入脈衝快速整形成寬度固定的脈衝。若採用 MCU，使用 I/O 口中斷去檢測輸入脈衝，再輸出寬度固定的脈衝，無法較快產生輸出脈衝，且若輸入脈衝較頻繁，將佔用大量 CPU 資源。

1.1 原理

本範例代碼將 Timer0 設置為事件計數功能，並使能 time-out 觸發 BPWM 和 PDMA 功能。當 Timer0 檢測到脈衝輸入時，會同時觸發 BPWM 和 PDMA 功能，由於 BPWM 回應比較快，會先輸出高電平，PDMA 接著會將其拉低，進而產生寬度固定的脈衝，其大致原理如圖 1-1 所示。

採用這種方式，運行過程中，MCU 會自動檢測輸入脈衝，並輸出寬度固定的脈衝，全部硬體完成，不需要 CPU 的介入。

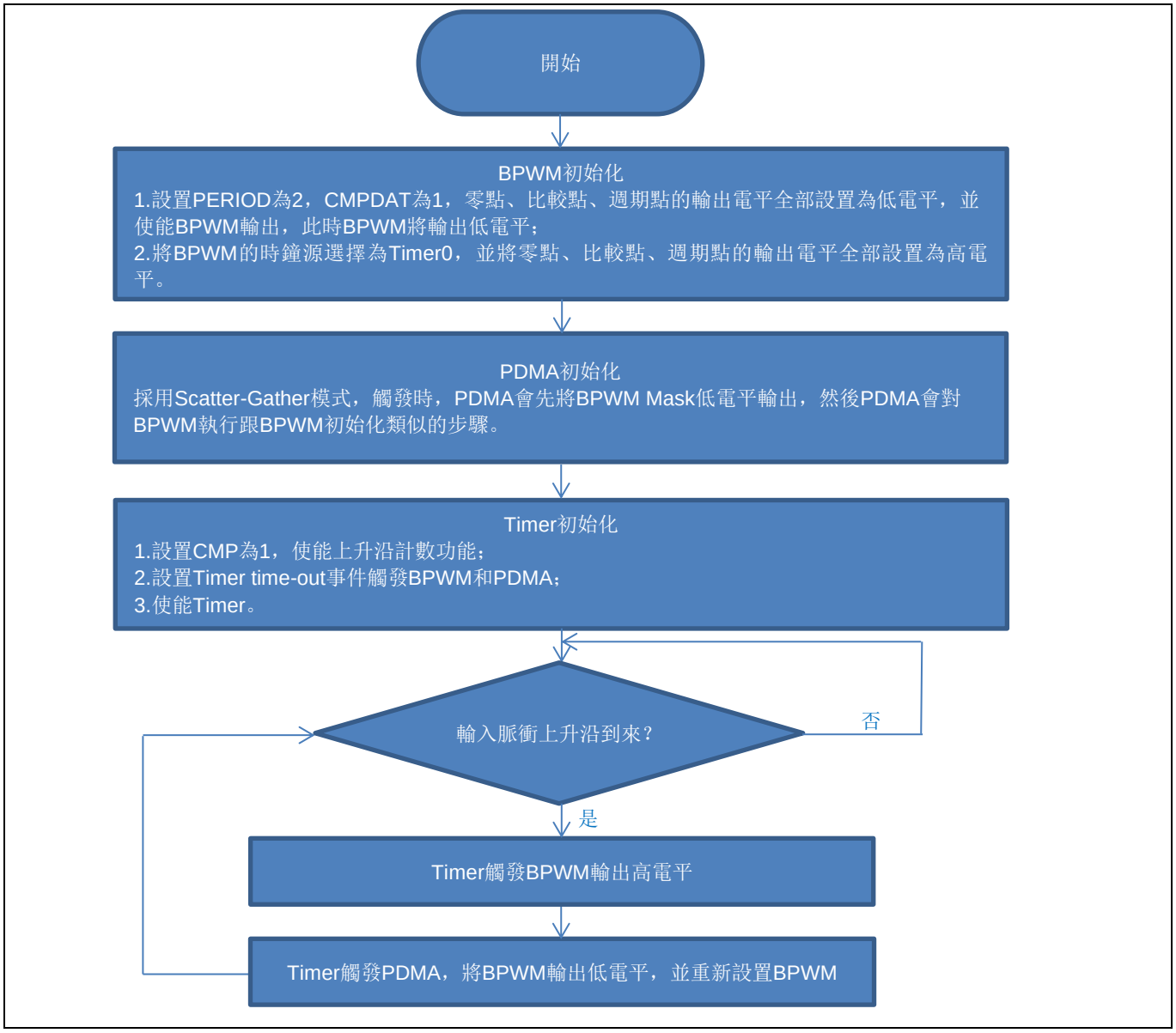

圖 1-1 快速脈衝整形輸出流程圖

1.2 執行結果

本範例代碼使用 PB4 輸出脈衝，類比需要整形的脈衝；PB5（TM0）輸入整形前的脈衝、PF3（BPWM1_CH0）輸出整形後的脈衝。測試時，需要將 PB4 與 PB5（TM0）接到一起。運行結果如圖 1-2 所示（通道 1：輸入脈衝；通道 2：輸出脈衝）。

從圖中可以看出，觸發輸入脈衝邊沿到輸出脈衝邊沿的時間差不多為 104 ns，輸出脈衝寬度約為 420 ns。

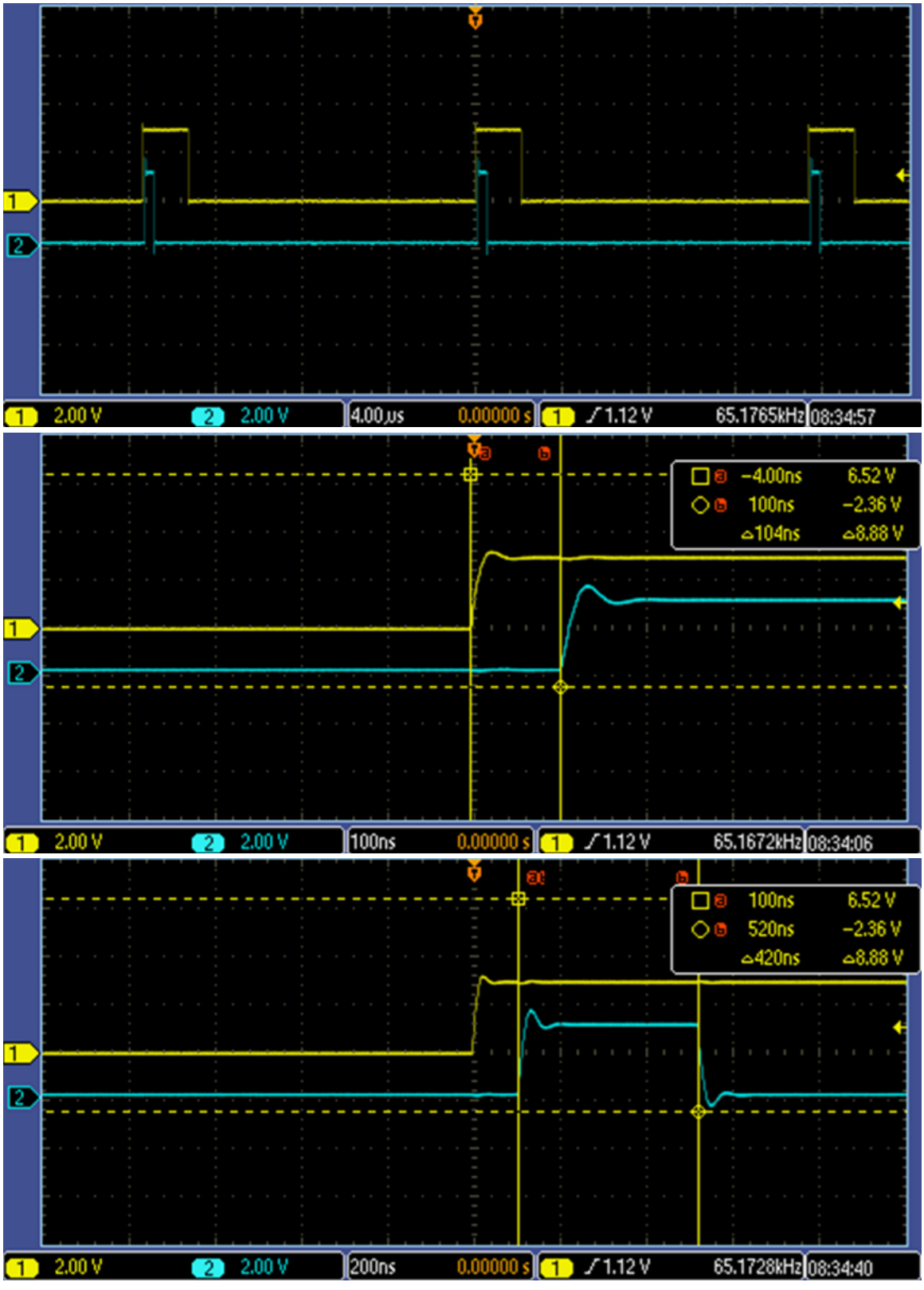
1 2.00 V
2 2.00 V
4.00 µs
0.00000 s
1 1.12 V
65.1765kHz
08:34:57
-4.00ns 6.52 V
100ns -2.36 V
Δ104ns Δ8.88 V
1 2.00 V
2 2.00 V
100ns
0.00000 s
1 1.12 V
65.1672kHz
08:34:06
100ns 6.52 V
520ns -2.36 V
Δ420ns Δ8.88 V
1 2.00 V
2 2.00 V
200ns
0.00000 s
1 1.12 V
65.1728kHz
08:34:40

圖 1-2 運行效果

2. 代碼介紹

Timer0 初始化時，會使能計時器的事件計數功能，並開啟 time-out 觸發 BPWM 和 PDMA 功能。

```
void TIMER0_Init()
{
    /* Set Timer0 to operate in periodic mode */
    TIMER_Open(TIMER0, TIMER_PERIODIC_MODE, 1);

    /* Set Timer0 prescale value */
    TIMER_SET_PRESCALE_VALUE(TIMER0, 0);

    /* Set Timer0 compared value */
    TIMER_SET_CMP_VALUE(TIMER0, 1);

    /* Counter increase on rising edge */
    TIMER_EnableEventCounter(TIMER0, TIMER_COUNTER_RISING_EDGE);

    /* Set trigger source is from Timer0 time-out interrupt signal */
    TIMER_SetTriggerSource(TIMER0, TIMER_TRGSRC_TIMEOUT_EVENT);

    /* Enalbe Timer0 trigger BPWM and PDMA */
    TIMER_SetTriggerTarget(TIMER0, TIMER_TRG_TO_BPWM | TIMER_TRG_TO_PDMA);
}
```

BPWM1 初始化時，會先將 BPWM1 的零點、週期點、向上比較點、向下比較點都設置為低電平，時鐘源設置成“BPWM_CLK”，BPWM1 運行起來後，BPWM1 將輸出低電平；接著會將 BPWM1 的時鐘源切換成“Timer0”，然後設置 BPWM1 的零點、週期點、向上比較點、向下比較點都為高電平。

```
void BPWM_Init(void)
{
    /* Set BPWM1 period */
    BPWM_SET_CNR(BPWM1, 0, 2);

    /* Set BPWM1 duty */
    BPWM_SET_CMR(BPWM1, 0, 1);

    /* Enable output of BPWM1 channel 0 */
    BPWM_EnableOutput(BPWM1, BIT0);

    /* Start BPWM1 counter */
    BPWM_Start(BPWM1, BIT0);

    /* Set BPWM1 zero point, period point, compare up point and compare down point output
    low */
    BPWM_SET_OUTPUT_LEVEL(BPWM1, BIT0, BPWM_OUTPUT_LOW, BPWM_OUTPUT_LOW, BPWM_OUTPUT_LOW,
    BPWM_OUTPUT_LOW);

    /* Set BPWM1 clock source to BPWM_CLK */
    BPWM_SetClockSource(BPWM1, 0, BPWM_CLKSRC_BPWM_CLK);
```

```
    /* Set BPWM1 clock source to TIMER0 */
    BPWM_SetClockSource(BPWM1, 0, BPWM_CLKSRC_TIMER0);

    /* Set BPWM1 zero point, period point, compare up point and compare down point output
    high */
    BPWM_SET_OUTPUT_LEVEL(BPWM1, BIT0, BPWM_OUTPUT_HIGH, BPWM_OUTPUT_HIGH,
    BPWM_OUTPUT_HIGH, BPWM_OUTPUT_HIGH);
}
```

PDMA 設置為 Scatter-Gather 模式，使用 Timer0 進行觸發，觸發時，首先會使能 BPWM1 的 Mask 功能，使 BPWM1 輸出低電平，接著會做一些類似 BPWM1 初始化的動作，詳情參考如下代碼。

```
void PDMA_Init(void)
{
    uint8_t i = 0, j = 1;

    /* Open PDMA */
    PDMA_Open(PDMA, 1 << BPWM_PDMA_CHANNEL);

    /* Set PDMA transfer mode */
    PDMA_SetTransferMode(PDMA, BPWM_PDMA_CHANNEL, PDMA_TMR0, TRUE,
    (uint32_t)&DMA_DESC[0]);

    /* Mask BPWM1 to output low */
    i=0;
    j=1;
    DMA_DESC[i].ctl = ((1 - 1) << PDMA_DSCT_CTL_TXCNT_Pos) | PDMA_WIDTH_32 | PDMA_SAR_FIX
    | PDMA_DAR_FIX | PDMA_REQ_BURST | PDMA_OP_SCATTER;
    DMA_DESC[i].src = (uint32_t)&u32Mask_Enable;
    DMA_DESC[i].dest = (uint32_t)&BPWM1->MSKEN;
    DMA_DESC[i].offset = (uint32_t)&DMA_DESC[j] - (PDMA->SCATBA);

    /* Set BPWM1 zero point and period point output low */
    i++;
    j++;
    DMA_DESC[i].ctl = ((1 - 1) << PDMA_DSCT_CTL_TXCNT_Pos) | PDMA_WIDTH_32 | PDMA_SAR_FIX
    | PDMA_DAR_FIX | PDMA_REQ_BURST | PDMA_OP_SCATTER;
    DMA_DESC[i].src = (uint32_t)&u32WGCTL0_Low;
    DMA_DESC[i].dest = (uint32_t)&BPWM1->WGCTL0;
    DMA_DESC[i].offset = (uint32_t)&DMA_DESC[j] - (PDMA->SCATBA);

    /* Set BPWM1 compare up point and compare down point output low */
    i++;
    j++;
    DMA_DESC[i].ctl = ((1 - 1) << PDMA_DSCT_CTL_TXCNT_Pos) | PDMA_WIDTH_32 | PDMA_SAR_FIX
    | PDMA_DAR_FIX | PDMA_REQ_BURST | PDMA_OP_SCATTER;
    DMA_DESC[i].src = (uint32_t)&u32WGCTL1_Low;
    DMA_DESC[i].dest = (uint32_t)&BPWM1->WGCTL1;
    DMA_DESC[i].offset = (uint32_t)&DMA_DESC[j] - (PDMA->SCATBA);

    /* Set BPWM1 clock source to BPWM_CLK */
    i++;
    j++;
```

```
    DMA_DESC[i].ctl = ((1 - 1) << PDMA_DSCT_CTL_TXCNT_Pos) | PDMA_WIDTH_32 | PDMA_SAR_FIX
    | PDMA_DAR_FIX | PDMA_REQ_BURST | PDMA_OP_SCATTER;
    DMA_DESC[i].src = (uint32_t)&u32CLKSRC_CLK;
    DMA_DESC[i].dest = (uint32_t)&BPWM1->CLKSRC;
    DMA_DESC[i].offset = (uint32_t)&DMA_DESC[j] - (PDMA->SCATBA);

    /* Set BPWM1 clock source to TIMER0 */
    i++;
    j++;
    DMA_DESC[i].ctl = ((1 - 1) << PDMA_DSCT_CTL_TXCNT_Pos) | PDMA_WIDTH_32 | PDMA_SAR_FIX
    | PDMA_DAR_FIX | PDMA_REQ_BURST | PDMA_OP_SCATTER;
    DMA_DESC[i].src = (uint32_t)&u32CLKSRC_Timer0;
    DMA_DESC[i].dest = (uint32_t)&BPWM1->CLKSRC;
    DMA_DESC[i].offset = (uint32_t)&DMA_DESC[j] - (PDMA->SCATBA);

    /* Set BPWM1 zero point and period point output high */
    i++;
    j++;
    DMA_DESC[i].ctl = ((1 - 1) << PDMA_DSCT_CTL_TXCNT_Pos) | PDMA_WIDTH_32 | PDMA_SAR_FIX
    | PDMA_DAR_FIX | PDMA_REQ_BURST | PDMA_OP_SCATTER;
    DMA_DESC[i].src = (uint32_t)&u32WGCTL0_High;
    DMA_DESC[i].dest = (uint32_t)&BPWM1->WGCTL0;
    DMA_DESC[i].offset = (uint32_t)&DMA_DESC[j] - (PDMA->SCATBA);

    /* Set BPWM1 compare up point and compare down point output high */
    i++;
    j++;
    DMA_DESC[i].ctl = ((1 - 1) << PDMA_DSCT_CTL_TXCNT_Pos) | PDMA_WIDTH_32 | PDMA_SAR_FIX
    | PDMA_DAR_FIX | PDMA_REQ_BURST | PDMA_OP_SCATTER;
    DMA_DESC[i].src = (uint32_t)&u32WGCTL1_High;
    DMA_DESC[i].dest = (uint32_t)&BPWM1->WGCTL1;
    DMA_DESC[i].offset = (uint32_t)&DMA_DESC[j] - (PDMA->SCATBA);

    /* Disable mask */
    i++;
    j = 0;
    DMA_DESC[i].ctl = ((1 - 1) << PDMA_DSCT_CTL_TXCNT_Pos) | PDMA_WIDTH_32 | PDMA_SAR_FIX
    | PDMA_DAR_FIX | PDMA_REQ_SINGLE | PDMA_OP_SCATTER;
    DMA_DESC[i].src = (uint32_t)&u32Mask_Disable;
    DMA_DESC[i].dest = (uint32_t)&BPWM1->MSKEN;
    DMA_DESC[i].offset = (uint32_t)&DMA_DESC[j] - (PDMA->SCATBA);
}
```

3. 軟體與硬體需求

3.1 軟體需求

- BSP 版本
  - M029G_M030G_M031G_Series_BSP_CMSIS_V3.03.000
- IDE 版本
  - Keil uVersion 4.74

3.2 硬體需求

- 電路元件
  - NuMaker-M030GTD V1.1
- 線路示意圖

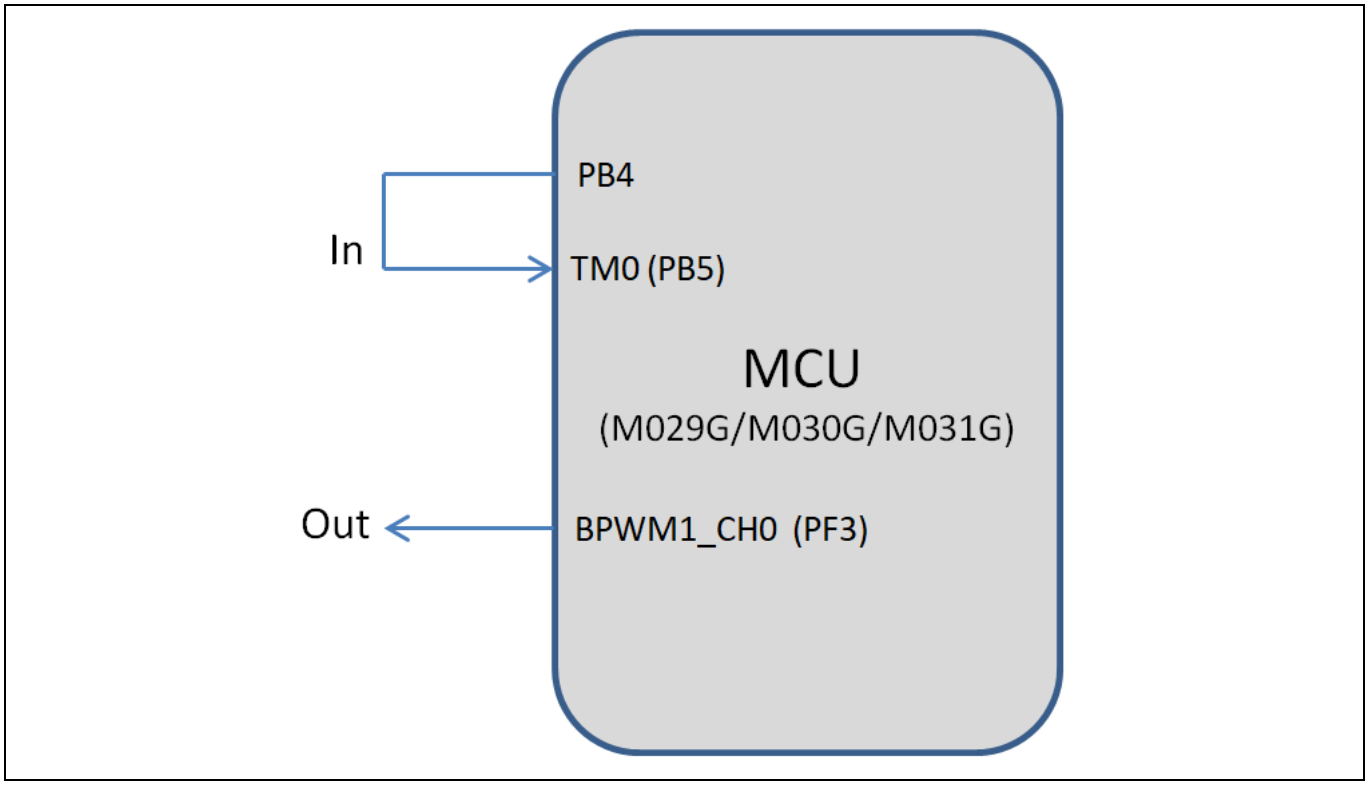

圖 3-1 線路示意圖

4. 目錄資訊

EC_M030G_Rapid_Pulse_Shaping_Output_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 *Rapid_Pulse_Shaping_Output.uvproj*。
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

6. 修訂紀錄

Date	Revision	Description
2024.04.08	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.