M030G 快速脉冲整形输出范例

NuMicro® 32 位系列微控制器范例代码介绍

文件信息

应用简述	本范例代码使用 M030G 实现快速脉冲整形输出
BSP 版本	M029G_M030G_M031G_Series_BSP_CMSIS_V3.03.000
开发平台	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

有些应用中，需要将不规则的输入脉冲快速整形成宽度固定的脉冲。若采用 MCU，使用 I/O 口中断去检测输入脉冲，再输出宽度固定的脉冲，无法较快产生输出脉冲，且若输入脉冲较频繁，将占用大量 CPU 资源。

1.1 原理

本范例代码将 Timer0 设置为事件计数功能，并使能 time-out 触发 BPWM 和 PDMA 功能。当 Timer0 检测到脉冲输入时，会同时触发 BPWM 和 PDMA 功能，由于 BPWM 响应比较快，会先输出高电平，PDMA 接着会将其拉低，进而产生宽度固定的脉冲，其大致原理如图 1-1 所示。

采用这种方式，运行过程中，MCU 会自动检测输入脉冲，并输出宽度固定的脉冲，全部硬件完成，不需要 CPU 的介入。

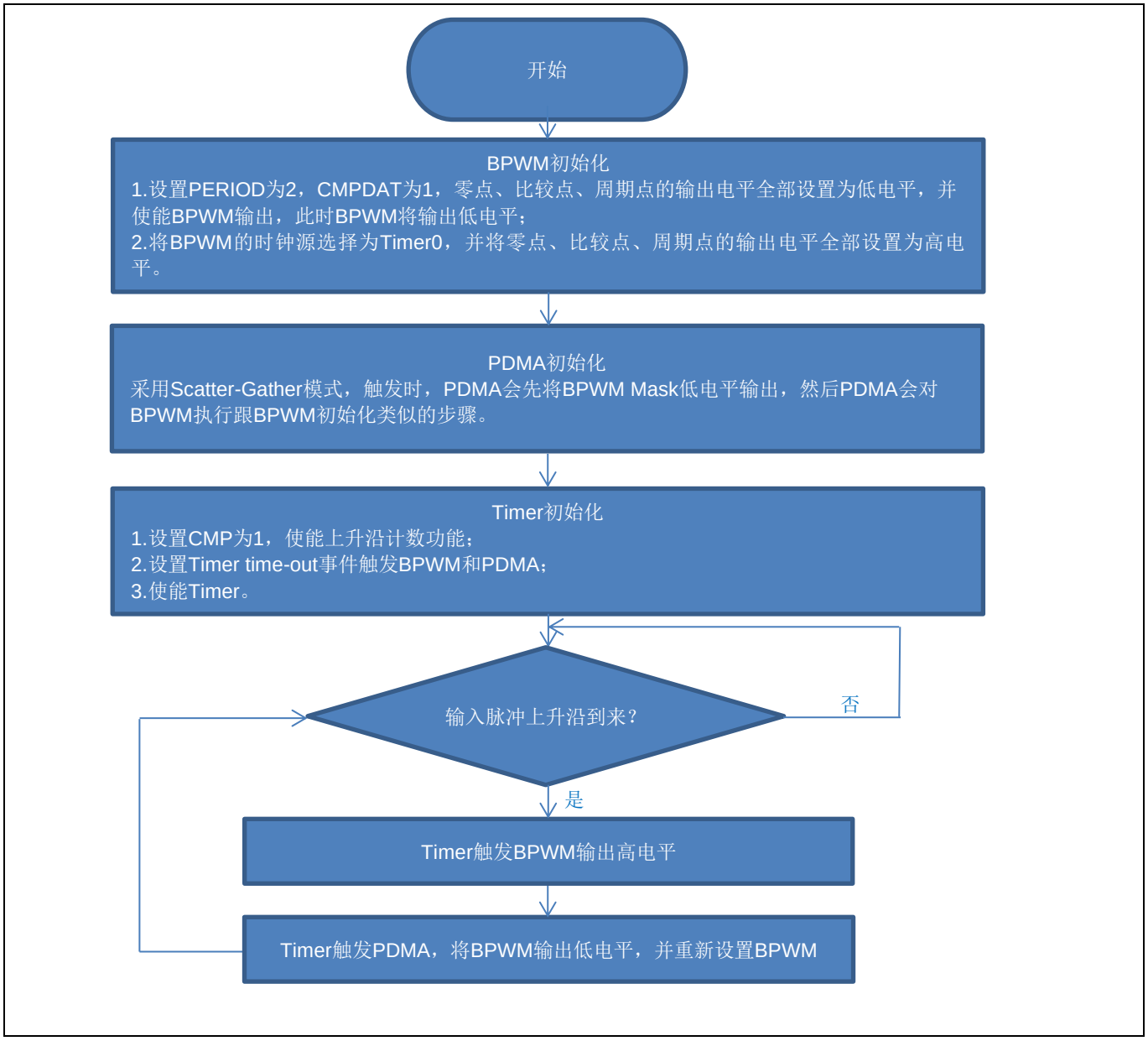

图 1-1 快速脉冲整形输出流程图

1.2 执行结果

本范例代码使用 PB4 输出脉冲，模拟需要整形的脉冲；PB5（TM0）输入整形前的脉冲、PF3（BPWM1_CH0）输出整形后的脉冲。测试时，需要将 PB4 与 PB5（TM0）接到一起。运行结果如图 1-2 所示（通道 1：输入脉冲；通道 2：输出脉冲）。

从图中可以看出，触发输入脉冲边沿到输出脉冲边沿的时间差不多为 104 ns，输出脉冲宽度约为 420 ns。

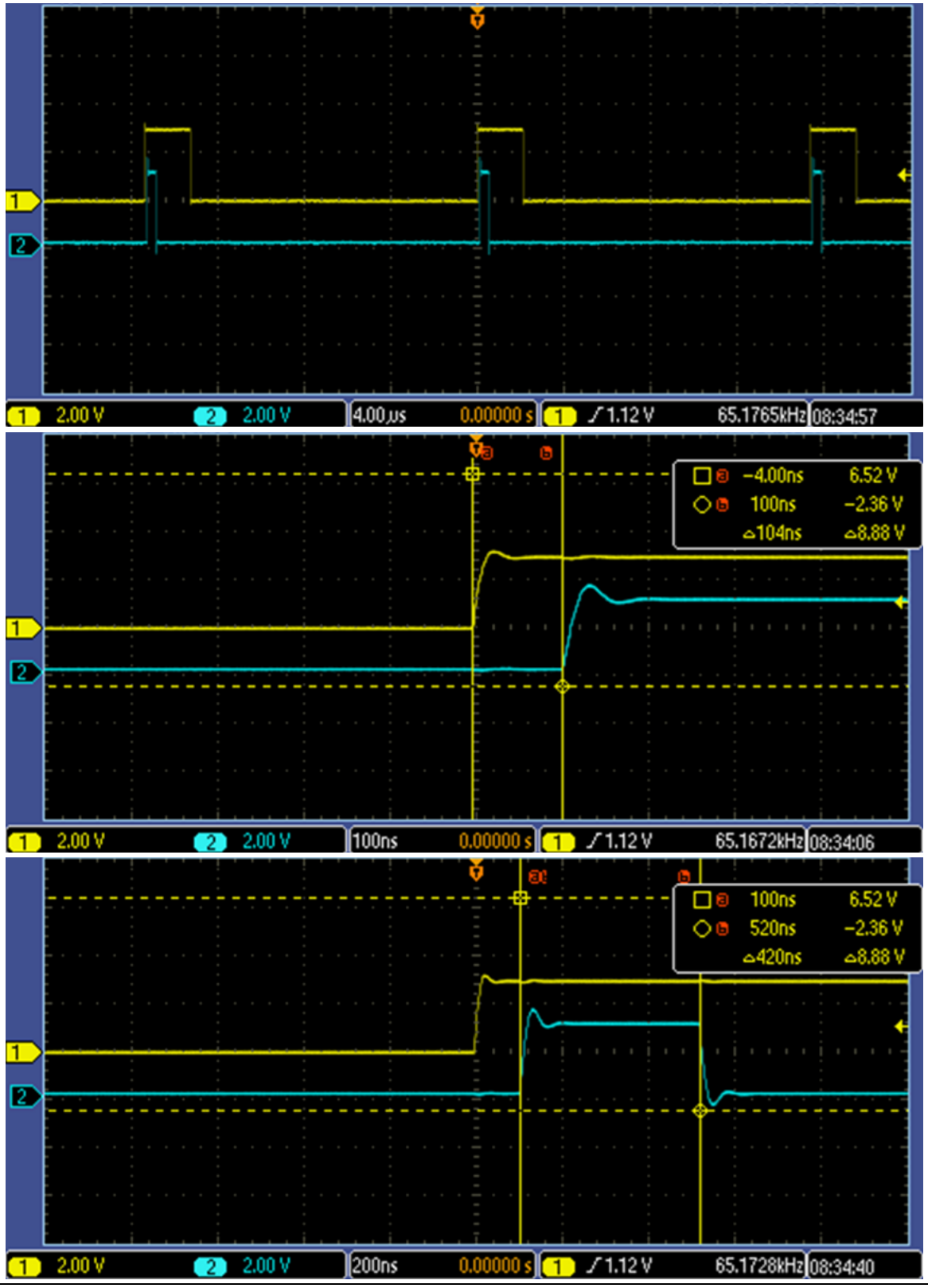
1 2.00 V
2 2.00 V
4.00 µs
0.00000 s
1 1.12 V
65.1765kHz
08:34:57
-4.00ns
6.52 V
100ns
-2.36 V
104ns
8.88 V
1 2.00 V
2 2.00 V
100ns
0.00000 s
1 1.12 V
65.1672kHz
08:34:06
100ns
6.52 V
520ns
-2.36 V
420ns
8.88 V
1 2.00 V
2 2.00 V
200ns
0.00000 s
1 1.12 V
65.1728kHz
08:34:40

图 1-2 运行效果

2. 代码介绍

Timer0 初始化时，会使能定时器的事件计数功能，并开启 time-out 触发 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 初始化时，会先将 BPWM1 的零点、周期点、向上比较点、向下比较点都设置为低电平，时钟源设置成“BPWM_CLK”，BPWM1 运行起来后，BPWM1 将输出低电平；接着会将 BPWM1 的时钟源切换成“Timer0”，然后设置 BPWM1 的零点、周期点、向上比较点、向下比较点都为高电平。

```
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 设置为 Scatter-Gather 模式，使用 Timer0 进行触发，触发时，首先会使能 BPWM1 的 Mask 功能，使 BPWM1 输出低电平，接着会做一些类似 BPWM1 初始化的动作，详情参考如下代码。

```
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. 软件与硬件需求

3.1 软件需求

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

3.2 硬件需求

- 电路组件
  - NuMaker-M030GTD V1.1
- 线路示意图

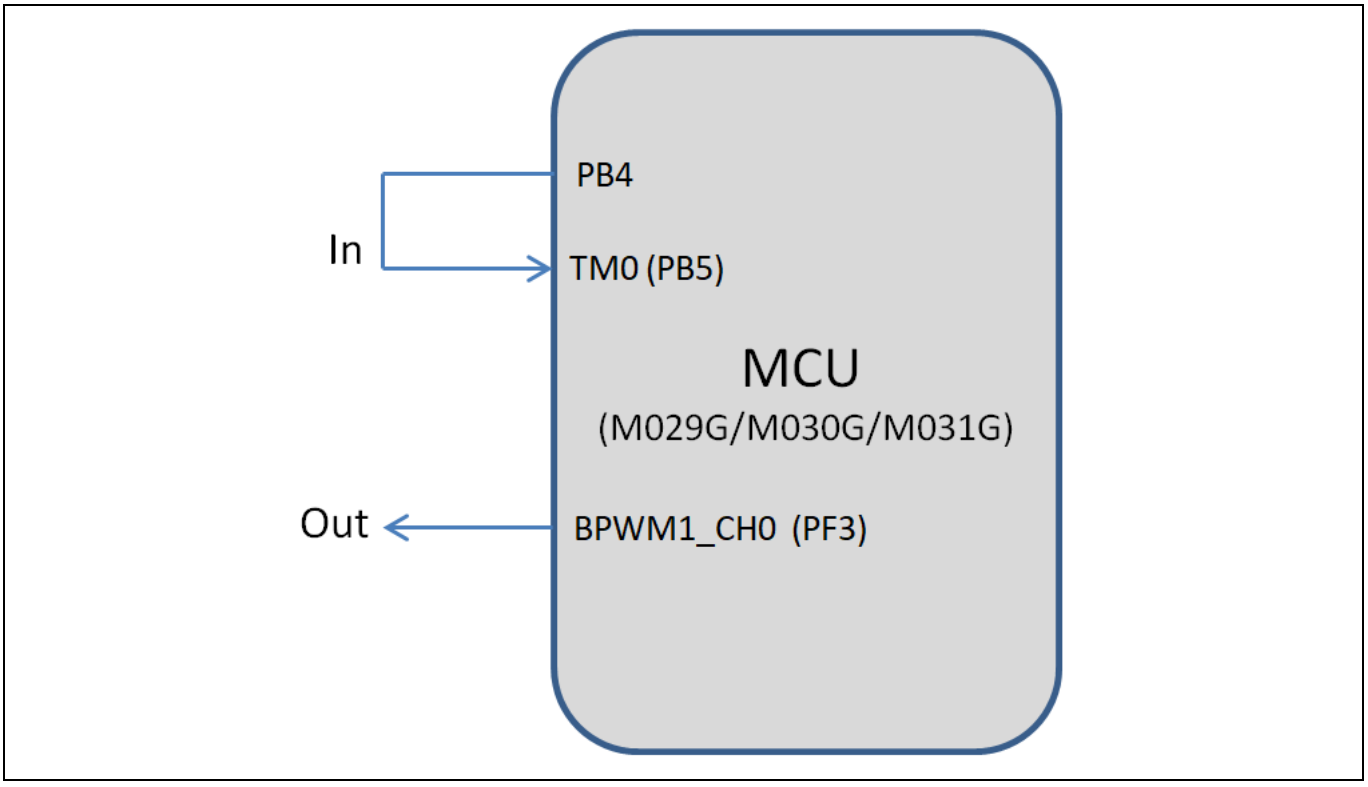

图 3-1 线路示意图

4. 目录信息

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

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 *Rapid_Pulse_Shaping_Output.uvproj*。
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
2024.04.08	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.