



CMS32F033 应用笔记

捕捉/比较/脉宽调制模块 (CCP0/1)

Rev. 1.00

请注意以下有关CMS知识产权政策

* 中微半导体（深圳）股份有限公司（以下简称本公司）已申请了专利，享有绝对的合法权益。与本公司MCU或其他产品有关的专利权并未被同意授权使用，任何经由不当手段侵害本公司专利权的公司、组织或个人，本公司将采取一切可能的法律行动，遏止侵权者不当的侵权行为，并追讨本公司因侵权行为所受的损失、或侵权者所得的不法利益。

* 中微半导体（深圳）股份有限公司的名称和标识都是本公司的注册商标。

* 本公司保留对规格书中产品在可靠性、功能和设计方面的改进作进一步说明的权利。然而本公司对于规格内容的使用不负责任。文中提到的应用其目的仅仅是用来做说明，本公司不保证和不表示这些应用没有更深入的修改就能适用，也不推荐它的产品使用在会由于故障或其它原因可能会对人身造成危害的地方。本公司的产品不授权适用于救生、维生器件或系统中作为关键器件。本公司拥有不事先通知而修改产品的权利，对于最新的信息，请参考官方网站 www.mcu.com.cn

目录

1. 模块概述	3
2. 模块特性	3
3. 模块功能	4
3.1 脉宽调制模式 (PWM)	4
3.1.1 时序图	4
3.1.2 寄存器说明	4
3.1.3 操作实例	4
3.1.4 样例代码	5
3.1.5 波形图	6
3.1.6 注意事项	6
3.2 捕捉模式 0	7
3.2.1 寄存器说明	7
3.2.2 操作实例	7
3.2.3 样例代码	8
3.3 捕获模式 1	8
3.3.1 时序图	9
3.3.2 寄存器说明	9
3.3.3 操作实例	9
3.3.4 样例代码	10
3.3.5 注意事项	10
4. 版本修订说明	11

1. 模块概述

芯片包含 2 组 CCP 模块（CCP0/CCP1），每组 CCP 对应 A、B 两路通道。CCP0 对应 CCP0A/CCP0B，CCP1 对应 CCP1A/CCP1B。

2. 模块特性

- ◆ 多达 2 组 CCP，最多支持 4 路 PWM 输出。
- ◆ 每组 CCP 都可设置独立的周期。
- ◆ CCPn 内部有 16 位计数器，可产生比较/溢出中断。
- ◆ CCPn 具有独立捕捉功能，可选在 A 路或 B 路管脚输入信号。
- ◆ CCP1 具有 4 通道捕捉功能，可同时捕获 CCP0A/CCP0B/CCP1A/CCP1B 输入信号。
- ◆ 捕捉模式 1 下支持捕获操作重新加载 CCP0 计数器功能。
- ◆ 内部通道 CAP3 支持模拟比较器输出捕获功能。
- ◆ 内部通道 CAP0-CAP3 支持软件捕获功能。

3. 模块功能

3.1 脉宽调制模式 (PWM)

每组 CCP 模块可控制两路 PWM (PWMnA、PWMnB)，每路 PWM 拥有独立的计数器。可通过 PWMnAOP/PWMnBOP 位设置 PWM 的输出极性。PWM 的输出通道对应为 CCPnA/CCPnB，通过芯片 IOCFG 寄存器设置为对应的复用功能。

3.1.1 时序图

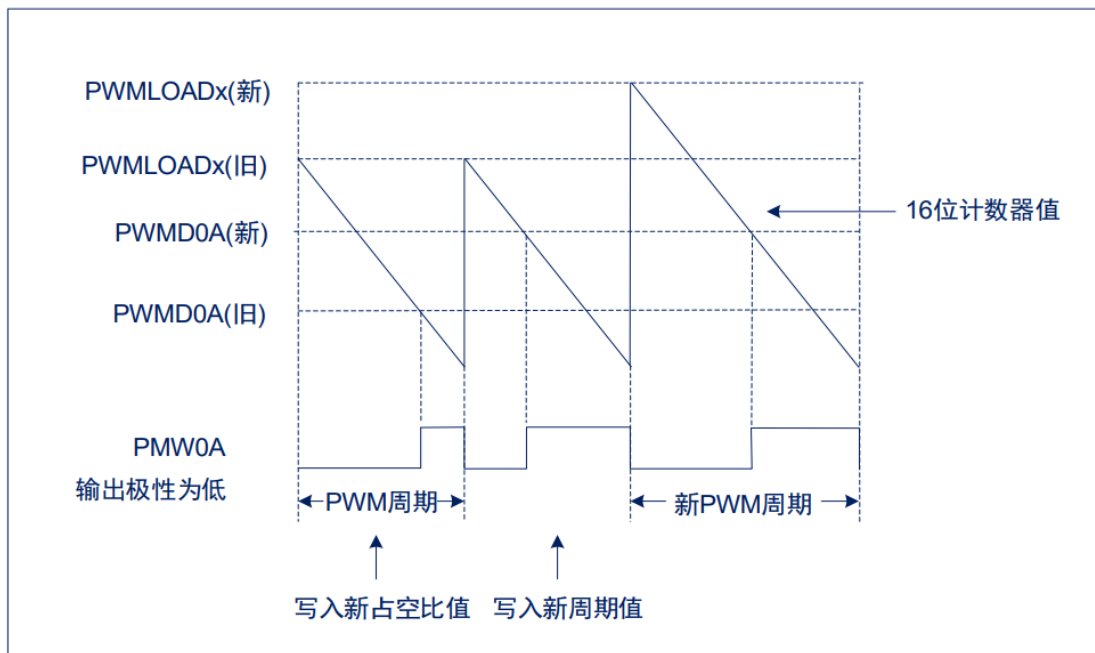


图 3-1: PWM 模式时序图

当 CCPx 运行位置 1 后，16 位计数器加载 CCPx 重加载寄存器的值并向下计数，当计数值等于 CCPDxA/B 的值时，PWMxA/PWMxB 输出电平发生改变。

3.1.2 寄存器说明

部分寄存器为被保护的寄存器，可根据手册说明对寄存器进行解锁或加锁操作。详情可参考芯片手册。

3.1.3 操作实例

实验效果：实现 1 组 PWM 的互补输出。

实现方式：使用 CCP0 模块的 CCP0A 与 CCP0B，输出引脚选择 P34、P35。

操作步骤：

- 1) 设置芯片 APB 时钟 (PCLK) 为 48Mhz，CCP 模块时钟使能位 (APBCKEN[CCPCE]) 置 1。
- 2) 设置 CCP0 模块时钟为 APB 时钟 (PCLK) 的 1 分频，CCPCON0[CCP0PS] = 0x0。
- 3) 设置 CCP0 模块的模式为 PWM 模式，CCPCON0[CCP0MS] = 0x1，CAPCON[CAPEN] = 0x0。
- 4) 设置 CCP0 模块周期值 100us 并使能重加载，CCP0LOAD[CCP0LOAD]=4800，CCP0LOAD[RELOAD] = 1。
- 5) 设置 CCP0A、CCP0B 占空比为 50%并开启 CCP0B 的反相输出，CCPD0A [CCP0ADATA]= 2400，CCPD0A[PWM0AOP] = 0，CCPD0A [CCP0BDATA] = 2400，CCPD0A[PWM0BOP] = 1。

- 6) 设置 IO 口复用功能, P34CFG = 0x5, P35CFG = 0x5。
- 7) 使能 CCP0 模块, CCPCON0[CCP0EN] = 1。
- 8) CCP0 开启运行, CCPRUN[CCPRUN0] = 1。

3.1.4 样例代码

```
void CCP_PWM_Config(void)
{
    /*(1)设置 CCP 的时钟*/
    SYS_EnablePeripheralClk(SYS_CLK_CCP_MSK); /*开启 CCP 模块的时钟, Fahb=48M*/
    CCP_ConfigCLK(CCP0,CCP_CLK_DIV_1,CCP_RELOAD_CCPLOAD,4800); /* Pccp0 = 100us*/
    /*(2)设置 CCP 运行模式*/
    CCP_EnablePWMMode(CCP0); /*设置 CCP0 为 PWM 模式*/
    /*(3)设置 CCP 比较值*/
    CCP_ConfigCompare(CCP0, CCPxA, 2400); /*50%*/
    CCP_ConfigCompare(CCP0, CCPxB, 2400); /*50%*/
    /*(4)设置 CCP 反相输出*/
    CCP_DisableReverseOutput(CCP0,CCPxA); /*关闭反相输出*/
    CCP_EnableReverseOutput(CCP0,CCPxB); /*开启反向输出*/
    /*(5)设置 CCP IO 口复用 */
    SYS_SET_IOCFG(IOP34CFG, SYS_IOCFG_P34_CCP0A);
    SYS_SET_IOCFG(IOP35CFG, SYS_IOCFG_P35_CCP0B);
    /*(6)开启 CCP 模块*/
    CCP_Start(CCP0); /*CCP0 模块使能*/
    CCP_EnableRun(CCP0); /*CCP0 模块运行*/
}
```

3.1.5 波形图

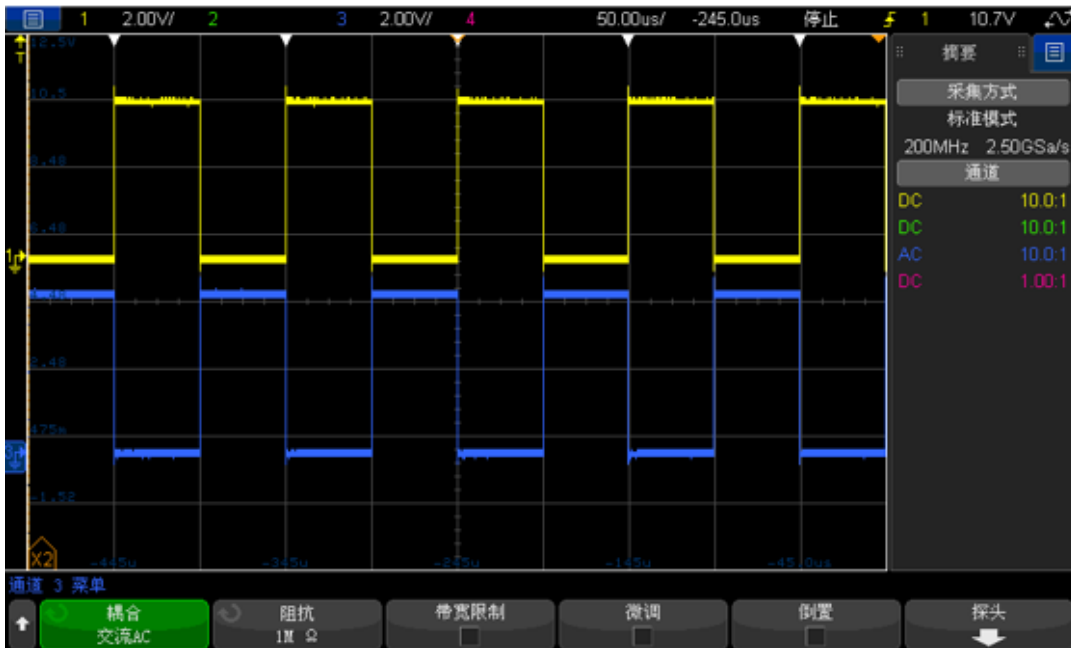


图 3-2: PWM 波形图

3.1.6 注意事项

- 1) CCPn 模块的计数器加载值与 CCPnA/CCPnB 的数据寄存器值的加载点相同，即在 CCPn 计数向下溢出后加载。
- 2) 设置模块时 建议先 使能 CCPn 模块 (CCPnEN=1)，再运行 CCPn 模块(CCPRUNn=1)，否则会导致第一个 PWM 周期内无占空比输出。
- 3) 若没有使能重加载位，则第二个 PWM 周期计数值将加载默认值即 0xFFFF。
- 4) 在 PWM 模式下，CCPn 可产生两种中断：
 - a) 当计数器递减到 0 时，产生溢出中断。
 - b) 当计数器的值与 CCPDnA 或 CCPDnB 的值相等时，产生比较中断，即会产生 2 个中断信号，因此在使用时，建议 CCPnA 与 CCPnB 的中断时间间隔长一些，给中断响应与处理预留足够的时间，防止中断信号出现遗漏。
 - c) CCPnA 与 CCPnB 产生的比较中断无法使用标志位进行区分，在一个完整周期内，CCPDnX 越大，就优先产生对应的比较中断。
- 5) PWM 的输出需要设置对应的 IOCFG 寄存器，并复用为 CCPn 功能。因此可通过修改 IOCFG 寄存器值进行 PWM 的输出开启与关闭，但 PWM 的波形开启与关闭时间是未知的，此时 CCP 模块仍处于工作状态。为了防止 IO 口处于不确定状态，需要设置 IO 口对应的 GPIO 模式，例如设置为输出或者上拉输入等模式。
- 6) 关于 CCPRUNn 位与 CCPnEN 位的使用。
 - a) 若在运行 (CCPRUNn =1, CCPnEN = 1) 过程中 CCPnEN=0，则 CCPn 停止运行，且 PWM 的输出 IO 口会处于悬空（不确定）状态。重新使能 CCPnEN 位，CCPn 重新开启计数并恢复 PWM 输出。
 - b) 若在运行 (CCPRUNn =1, CCPnEN = 1) 过程中 CCPRUNn =0，则 CCPn 停止运行，且 PWM 的输出 IO 口会处于悬空（不确定）状态。重新使能 CCPRUNn 位，CCPn 重新开启计数并恢复 PWM 输出。

3.2 捕捉模式 0

每组 CCP 可设置从 A 路或 B 路作为外部捕捉信号管脚。当触发捕捉条件时，计数器停止计数，CCPx A 或 CCPxB 返回当前计数器的值，若需要进行下一次捕捉，需将 CCPRUNx 清零，再置位。

3.2.1 寄存器说明

部分寄存器为被保护的寄存器，可根据手册说明对寄存器进行解锁或加锁操作。详情可参考芯片手册。

3.2.2 操作实例

实验效果：捕获周期为 10KHz，占空比为 50%的方波。

实现方式：使用 CCP0 模块的 CCP0A，输入引脚选择 P34。

操作步骤：

- 1) 设置芯片 APB 时钟 (PCLK) 为 48Mhz，CCP 模块时钟使能位 (APBCKEN[CCPCE]) 置 1。
- 2) 设置 CCP0 模块时钟为 APB 时钟(PCLK)的 1 分频，CCPCON0[CCP0PS] = 0x0。
- 3) 设置 CCP0 模块的模式为捕获模式 0，CCPCON0[CCP0MS] = 0x0，CAPCON[CAPEN] = 0x0。
- 4) 设置 CCP0 模块周期值 100us 并使能重加载，CCP0LOAD[CCP0LOAD]=4800，CCP0LOAD[RELOAD] = 1。
- 5) 设置 CCP0 模块的捕获通道为 CCP0A，并设置捕获模式为上升沿开始计数，下降沿捕捉并产生中断。CCPCON0[CCP0CM0CS] = 0，CCPCON0[CCP0CM0ES] = 0x2。
- 6) 设置 CCP0 模块捕获中断。
- 7) 设置捕获 IO 引脚，设置 IOCFG 寄存器，P34CFG = 0x5。
- 8) 使能 CCP0 模块，CCPCON0[CCP0EN] = 1。
- 9) CCP0 开启运行，CCPRUN[CCPRUN0] = 1。

3.2.3 样例代码

```
void CCP_Capture_Mode0_Config(void)
{
    /*(1)设置 CCP 的时钟*/
    SYS_EnablePeripheralClk(SYS_CLK_CCP_MSK); /*开启 CCP 模块时钟,Fccp = 48Mhz*/
    CCP_ConfigCLK(CCP0,CCP_CLK_DIV_1,CCP_RELOAD_CCPLOAD,48000); /*捕获周期 1000us*/
    /*(2)设置 CCP 运行模式*/
    CCP_EnableCAPMode0(CCP0); /*设置 CCP0 的工作模式为捕获模式 0*/
    /*设置 CCP0 的捕获模式 0 的捕获方式为上升沿到下降沿(高电平时间)*/
    CCP_ConfigCAPMode0(CCP0,CCP_CAP_MODE0_RISING_TO_FALLING);
    /*(3)设置 CCP 捕获通道*/
    CCP_SelCAPMode0Channel(CCP0,CCPxA); /*设置捕获通道为 CCPA0*/
    SYS_SET_IOCFG(IOP34CFG, SYS_IOCFG_P34_CCP0A); /*设置 P34 为 CCPA0 捕获通道*/
    /*(4)设置 CCP 捕获中断*/
    CCP_EnableOverflowInt(CCP0); /*开启 CCP0 模块计数溢出中断*/
    CCP_EnableCAPMode0Int(CCP0); /*开启 CCP0 模块捕获模式 0 中断*/
    NVIC_EnableIRQ(CCP_IRQn);
    /*(5)设置 CCP 中断优先级*/
    NVIC_SetPriority(CCP_IRQn,3); /*优先级 0~3, 0 最高、3 最低*/
    /*(6)开启 CCP 模块*/
    CCP_Start(CCP0); /*CCP0 模块使能*/
    CCP_EnableRun(CCP0);
}
```

3.3 捕获模式 1

捕获模式 1 只适用于 CCP1 模块。CCP1 包括 4 路内部通道有 CAP0, CAP1, CAP2, CAP3。其中一路通道可选择外部通道中 ECAP00-03 或 ECAP10-13 中任意一路作为捕获通道。也可分别选择 CCP0A/CCP0B/CCP1A/CCP1B 作为捕获通道。

3.3.1 时序图

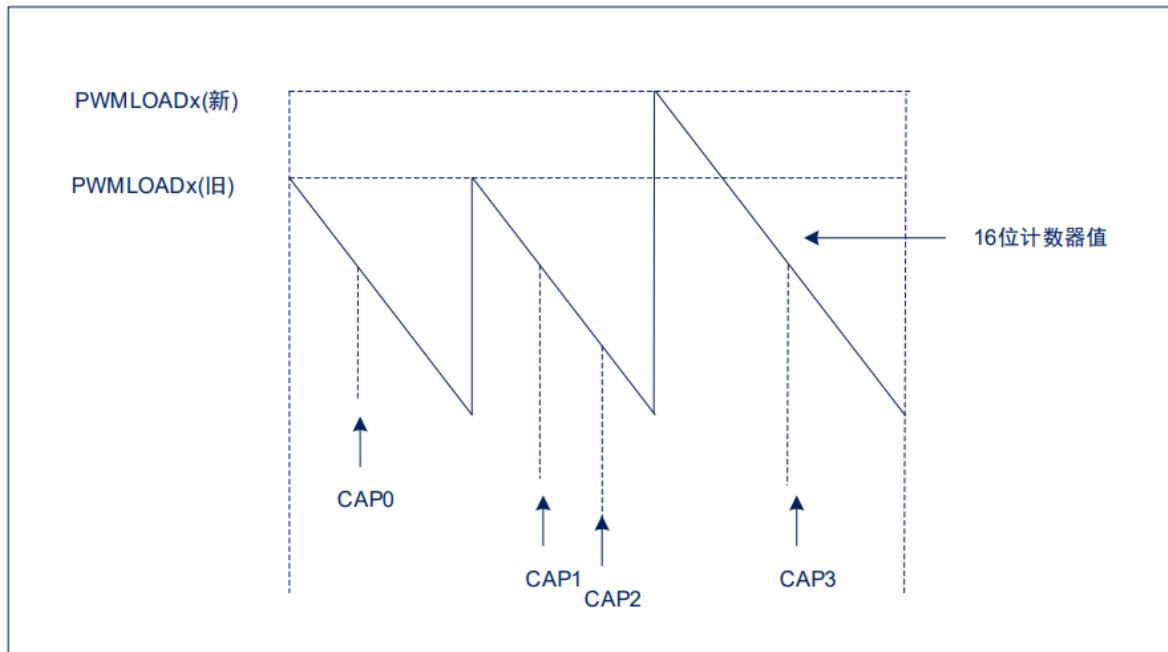


图 3-3: CAP0-CAP3 通道捕获操作

3.3.2 寄存器说明

部分寄存器为被保护的寄存器，可根据手册说明对寄存器进行解锁或加锁操作。详情可参考芯片手册。

3.3.3 操作实例

实验效果：捕获周期为 10KHz，占空比为 50%的方波。

实现方式：使用 CCP1 模块的 CAP0，输入引脚选择 P15(ECAP02)。

操作步骤：

- 1) 设置芯片 APB 时钟 (PCLK) 为 48Mhz，CCP 模块时钟使能位 (APBCKEN[CCPCE]) 置 1。
- 2) 设置 CCP0 模块时钟为 APB 时钟(PCLK)的 64 分频，CCPCON0[CCP0PS] = 0x3。
- 3) 设置 CCP1 模块的模式为捕获模式 1，CCPCON0[CCP0MS] = 0x0，CAPCON[CAPEN] = 0x1。
- 4) 设置捕获通道为 CAP0,开启上升沿捕获，输入引脚选择 ECAP02 (P15)，CAPCON[CAP0ES]=0x1，CAPCHS[ECAPS] = 0，CAPCHS[CAP0CHS] = 0x2。
- 5) 设置 CCP1 计数溢出中断、CAP0 捕获中断。
- 6) 设置捕获 IO 引脚，设置 IOCFG 寄存器，P15CFG = 0x0。
- 7) 使能 CCP1 模块，CCPCON1[CCP1EN] = 1。
- 8) CCP1 开启运行，CCPRUN[CCPRUN1] = 1。

3.3.4 样例代码

```
void CCP_Capture_Mode1_Config(void)
{
    /*(1)设置 CCP 的时钟*/
    SYS_EnablePeripheralClk(SYS_CLK_CCP_MSK); /*开启 CCP 模块时钟, Fapb = 48Mhz*/
    CCP_ConfigCLK(CCP1,CCP_CLK_DIV_64,CCP_RELOAD_CCPLOAD,4800);/*捕获计数周期 6400us*/
    /*(2)设置 CCP 运行模式*/
    CCP_EnableCAPMode1(); /*设置 CCP 模块工作模式为捕获模式 1*/
    CCP_ConfigCAPMode1(CAP0,CCP_CAP_MODE1_RISING); /*设置捕获通道为 CAP0, 上升沿捕获*/
    /*(3)设置 CCP 捕获通道*/
    CCP_SelCAPMode1Channel(CAP0,ECAP02); /*CAP0 通道选择连接外部 ECAP02 接口*/
    SYS_SET_IOCFG(IOP15CFG, SYS_IOCFG_P15_ECAP02); /*设置 P15 为 ECAP02*/
    /*(4)设置 CCP 捕获中断*/
    CCP_EnableOverflowInt(CCP1); /*开启 CCP1 溢出中断*/
    CCP_EnableCAPMode1Int(CAP0); /*开启 CAP0 通道捕获中断*/
    NVIC_EnableIRQ(CCP_IRQn);
    /*(5)设置 CCP 中断优先级*/
    NVIC_SetPriority(CCP_IRQn,3); /*优先级 0~3, 0 最高、3 最低*/
    /*(6)开启 CCP 模块*/
    CCP_Start(CCP1); /*CCP1 模块使能*/
    CCP_EnableRun(CCP1);
}
```

3.3.5 注意事项

- 1) 选择 ECAP0x 与 ECAP1x 作为捕获信号输入时, 需要将 IOCFG 设置为 GPIO (0x0) 模式, 并且可以使用 GPIO 模块的相关功能, 例如上拉输入、输入滤波功能等。
- 2) 选择 CAPnX 作为捕获信号输入时, 需要将 IOCFG 设置为 CCP (0x5) 模式。

4. 版本修订说明

版本号	时间	修改内容
V1.00	2021 年 9 月	初始版本