



# CMS32M57xx 应用笔记

## 逐波限流功能

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](http://www.mcu.com.cn)

---

## 目录

<b>1. 概述</b> .....	<b>3</b>
1.1 目的 .....	3
1.2 定义、首字母缩略词和缩写词 .....	3
1.3 内容提要 .....	3
<b>2. 模块概述</b> .....	<b>4</b>
2.1 EPWM 模块 .....	4
2.2 ACMP 模块 .....	4
<b>3. 恢复模式应用</b> .....	<b>5</b>
3.1 操作实例 .....	5
3.2 样例代码 .....	6
3.3 样例波形 .....	9
<b>4. 更多信息</b> .....	<b>11</b>
<b>5. 版本修订说明</b> .....	<b>12</b>

# 1. 概述

## 1.1 目的

本文档介绍了 CMS32M57xx EPWM 模块和 ACMP 模块的部分特性，以及如何结合 EPWM 模块的恢复模式与 ACMP 模块实现逐波限流功能。

## 1.2 定义、首字母缩略词和缩写词

表 1-1: 定义、首字母缩略词和缩写词

缩写	说明
ACMP	模拟比较器
EPWM	增强型 PWM
C0P0	ACMP0 正向输入端口 0
C0O	ACMP0 输出端口

## 1.3 内容提要

本文档包含以下内容：

第 2 章 模块概述。

第 3 章 恢复模式应用。

## 2. 模块概述

### 2.1 EPWM 模块

增强性 PWM 模块支持 6 路 PWM 发生器, 可以配置成相互独立的 6 路 PWM 输出, 也可以配置成 3 对分别带有编程死区发生器的互补 PWM (EPWM0-EPWM1, EPWM2-EPWM3, EPWM4-EPWM5)。

EPWM 模块具备刹车保护功能, 存在 4 种不同刹车模式, 如下表 2-1 所示:

表 2-1: 刹车保护模式

BRKMS	刹车模式
00	停止模式 (软件恢复)
01	暂停模式 (软件恢复)
10	恢复模式 (硬件恢复)
11	延时恢复模式 (硬件恢复)

### 2.2 ACMP 模块

芯片内部包含两个模拟比较器。可按照比较器的配置适用于不同的应用场合。当正端电压大于负端电压时, 比较器 输出逻辑 1, 反之输出 0, 也可以通过输出极性选择位进行改变。当比较器输出值发生改变时, 每路比较器都可通过配置产生中断。

模拟比较器模块正端可选择 4 路端口输入与 OPA0/1, PGA0/1 的输出端口; 负端可选择端口输入与内部输入电压参考 ( $0.1 \sim 0.85$ )  $V_{REF}$ ; 输出信号可作为 EPWM 的刹车触发信号。模拟比较器结构图如下图 2-1:

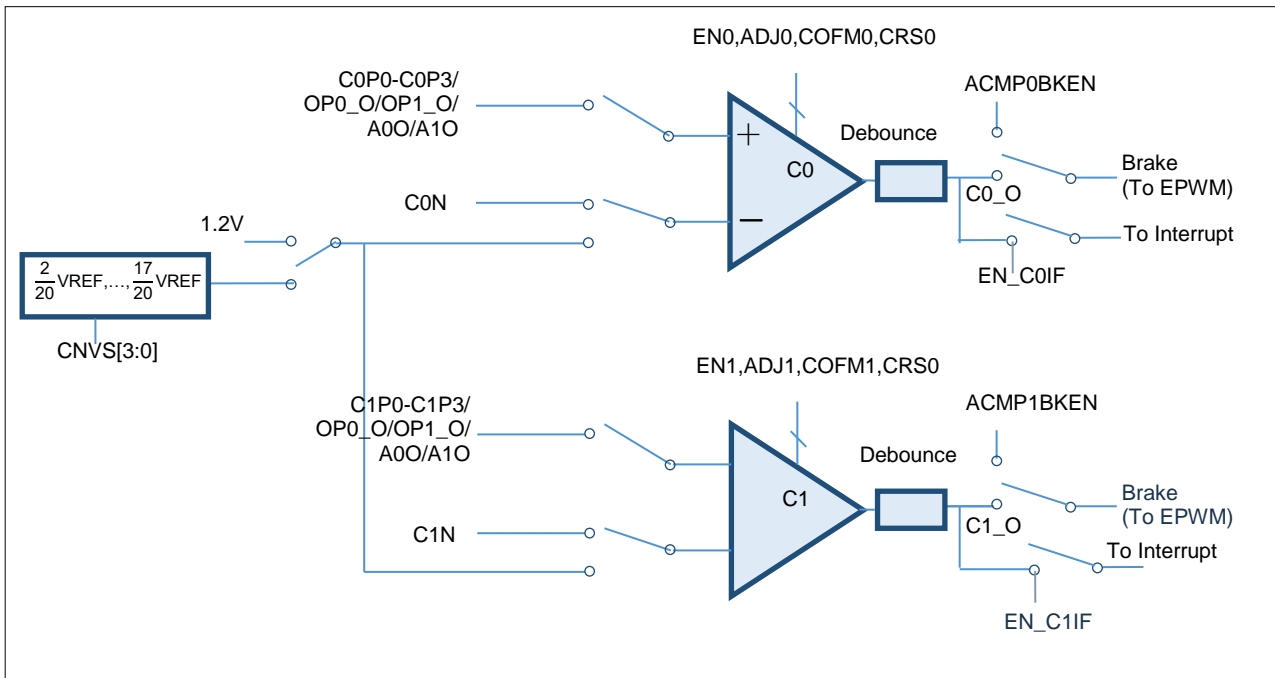


图 2-1: 比较器结构图

### 3. 恢复模式应用

ACMP0 负端选择电平为 1.2V (BG)；正向端选择 C0P0 (P04) 端口，开启 ACMP0 输出端口 C0O (P03)。设置 ACMP0 输出高电平触发 EPWM 硬件刹车保护，刹车模式设置成恢复模式，刹车模式 EPWM 输出设置成低电平。

如下图 3-1 所示，当电路正常工作时，ACMP 正端信号电压低于负端参考电压，ACMP 持续输出低电平；当电路发生过流现象，模拟比较器输出高电平，触发 EPWM 刹车，截断 EPWM 输出。当 ACMP0 输出信号恢复低电平时，EPWM 在最近一次零点恢复输出。

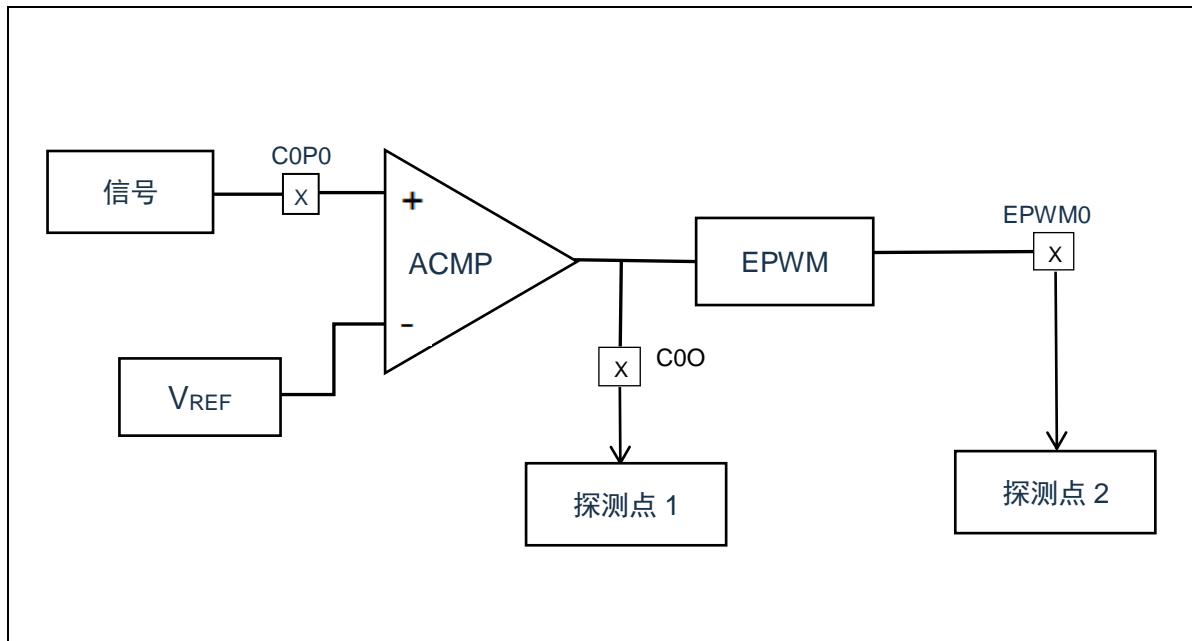


图 3-1: 刹车模式应用框图

探测点 1 观察 ACMP0 输出电平信号，探测点 2 观察 EPWM0 输出波形。

#### 3.1 操作实例

实例目标：EPWM 模块在恢复模式下实现 EPWM0 输出开/断。

操作步骤：

- 1) 设置芯片 APB 时钟 (PCLK) 为 48Mhz。
- 2) 设置 ACMP0 模块，设置正端输入为 C0P0，负端输入选择 1.2V (BG)，开启滤波输出；设置 P04 作为正端输入，P30 作为输出。
- 3) 设置 EPWM 模块，设置为互补向下计数模式，时钟为 APB 时钟 1 分频，EPWM0 周期为 4800 (100us)，通道 EPWM0 占空比为 50%；设置为自动加载模式，每个周期点和零点自动加载；EPWM 模块刹车设置为恢复模式，触发源为 ACMP0 高电平触发，刹车状态下 EPWM0 输出低电平；刹车信号撤销后，EPWM0 的最近加载点恢复。

## 3.2 样例代码

```
int main(void)
{
    SYS_DisableIOCFGProtect();           /*关闭 IOCONFIG 写保护*/
    SYS_DisableGPIO0Protect();          /*关闭 GPIO0 的相关寄存器写保护*/
    SYS_DisableGPIO1Protect();          /*关闭 GPIO1 的相关寄存器写保护*/
    SYS_DisableGPIO2Protect();          /*关闭 GPIO2 的相关寄存器写保护*/
    SYS_DisableGPIO3Protect();          /*关闭 GPIO3 的相关寄存器写保护*/
    SYS_DisableGPIO4Protect();          /*关闭 GPIO4 的相关寄存器写保护*/
    SYS_DisableGPIO5Protect();          /*关闭 GPIO4 的相关寄存器写保护*/
    SYS_ConfigHSI(SYS_CLK_HSI_48M);     /*设置内部高速时钟为 48Mhz*/
    SYS_EnableHSI();                    /*开启高速时钟*/
    SYS_ConfigAHBClock(SYS_CLK_SEL_HSI,SYS_CLK_DIV_1); /*设置 AHB 时钟为高速时钟的 1 分频*/
    SYS_ConfigAPBClock(AHB_CLK_DIV_1); /*设置 APB 时钟为 AHB 时钟的 1 分频*/
    SystemCoreClockUpdate();           /*刷新 SystemCoreClk、SystemAPBClock 变量值*/

    ACMP_ACMP0_Config();                /*配置 ACMP0 模块*/
    EPWM_Brake_ACMP0();                 /*配置 EPWM 模块*/

    while(1)
    {
        ;
    }
}

void ACMP_ACMP0_Config(void)
{
    uint32_t AdjResult;
    /*(1)开启 ACMP 模块时钟*/
    SYS_EnablePeripheralClk(SYS_CLK_ACMP_MSK);
    /*(2)获取 ACMP0 校调调节值*/
    AdjResult = ACMP_GetOffsetAdjValue(ACMP0,ACMP_ADJ_GND, NULL);
    /*(3)设置 ACMP0 正端输入*/
    ACMP_ConfigPositive(ACMP0,ACMP_POSSEL_P0);
    /*(4)设置 ACMP0 负端输入*/
    ACMP_ConfigNegative(ACMP0,ACMP_NEGSEL_VREF_BG, NULL); /*负端选择内部 1.2V 基准电压*/
    /*(5)设置电压调节*/
    ACMP_EnableCADJAdjust(ACMP0, AdjResult);
    /*(6)开启滤波输出*/
    ACMP_EnableFilter(ACMP0, ACMP_NGCLK_65_TSYS);
    /* (7)关闭反相输出*/
    ACMP_DisableReverseOutput(ACMP0);
    /*(8)关闭事件输出*/
    ACMP_ConfigEventAndIntMode(ACMP0,ACMP_EVENT_INT_RISING); /*上升沿*/
    ACMP_DisableEventOut(ACMP0);
    /*(9)设置中断*/
    ACMP_DisableInt(ACMP0); /*中断触发方式与事件触发方式共用*/
    NVIC_DisableIRQ(ACMP_IRQn);
    /*(10)设置中断优先级*/
    NVIC_SetPriority(ACMP_IRQn,2);
}
```

```

/*(11)设置迟滞*/
ACMP_DisableHYS(ACMP0);

/*(12)配置 ACMP IO 口*/
SYS_SET_IOCFG(IOP04CFG,SYS_IOCFG_P04_C0P0);          /*设置 P04 为 ACMP0 正端输入*/
SYS_SET_IOCFG(IOP30CFG,SYS_IOCFG_P30_C0O);          /*设置 P30 为 ACMP0 输出*/

/*(13)开启 ACMP0*/
ACMP_Start(ACMP0);
}

void EPWM_Brake_ACMP0(void)
{
    /*(1)设置 EPWM 运行模式*/
    EPWM_ConfigRunMode( EPWM_COUNT_DOWN |             /*向下计数模式(边沿对齐)*/
                       EPWM_OCU_SYMMETRIC |         /*对称模式*/
                       EPWM_WFG_COMPLEMENTARYK |    /*互补模式*/
                       EPWM_OC_INDEPENDENT);        /*独立输出模式*/

    /*(2)设置 EPWM 时钟周期与死区*/
    SYS_EnablePeripheralClk(SYS_CLK_EPWM_MSK);      /*开启 EPWM 时钟*/
    EPWM_ConfigChannelClk( EPWM0, EPWM_CLK_DIV_1);  /*设置 EPWM0 时钟为 APB 时钟的 1 分频*/
    EPWM_ConfigChannelPeriod(EPWM0, 4800);          /*EPWM0 通道的周期 100us*/
    EPWM_ConfigChannelSymDuty(EPWM0, 2400);         /*EPWM0 通道的占空比 = 50%*/
    EPWM_DisableDeadZone(EPWM_CH_0_MSK | EPWM_CH_1_MSK); /*关闭死区*/

    /*(3)设置 EPWM 反向输出*/
    EPWM_DisableReverseOutput( EPWM_CH_0_MSK | EPWM_CH_1_MSK); /*关闭反相输出*/

    /*(4)设置 EPWM 加载方式*/
    EPWM_EnableAutoLoadMode(EPWM_CH_0_MSK);        /*设置为自动加载模式*/
    EPWM_ConfigLoadAndIntMode(EPWM0, EPWM_EACH_PERIOD_ZERO); /*加载点设置为每个周期点和零点*/

    /*(5)设置 EPWM 刹车*/
    EPWM_EnableFaultBrake(EPWM_BRK_ACMP0LE);      /*开启 ACMP0 输出电平刹车*/
    EPWM_EnableACMP0LEBrake(EPWM_ACMP0BRK_LEVEL_HIGH); /*ACMP0 输出高电平刹车*/
    EPWM_ConfigBrakeMode(EPWM_BRK_RECOVER);        /*恢复模式*/
    EPWM_ConfigBrakeRecoverLoad(EPWM_BRK_LOAD_EPWM0); /*EPWM0 的加载点恢复*/
    EPWM_ConfigFaultBrakeLevel(EPWM_CH_0_MSK,0);   /*刹车时 EPWM0 输出低电平*/
    EPWM_ConfigFaultBrakeLevel(EPWM_CH_1_MSK,0);   /*刹车时 EPWM1 输出低电平*/
    EPWM_AllBrakeEnable();

    /*(6)设置中断*/
    EPWM_EnableBrakeInt();                          /*开启刹车中断*/
    EPWM_EnableZeroInt(EPWM_CH_0_MSK);             /*开启 EPWM0 零点中断*/
    NVIC_EnableIRQ(EPWM_IRQn);
    NVIC_SetPriority(EPWM_IRQn,2);                 /*优先级 0~3, 0 最高、3 最低*/
}
    
```

```
/*(7)设置 IO 口输出*/
```

```
SYS_SET_IOCFG(IOP47CFG, SYS_IOCFG_P47_EPWM0);
```

```
SYS_SET_IOCFG(IOP46CFG, SYS_IOCFG_P46_EPWM1);
```

```
EPWM_EnableOutput(EPWM_CH_0_MSK | EPWM_CH_1_MSK);
```

```
/*(8)开启 EPWM*/
```

```
EPWM_Start(EPWM_CH_0_MSK);
```

```
}
```



### 3.3 样例波形

备注：蓝线代表 EPWM0 输出波形 红线代表 ACMP0 输出波形。

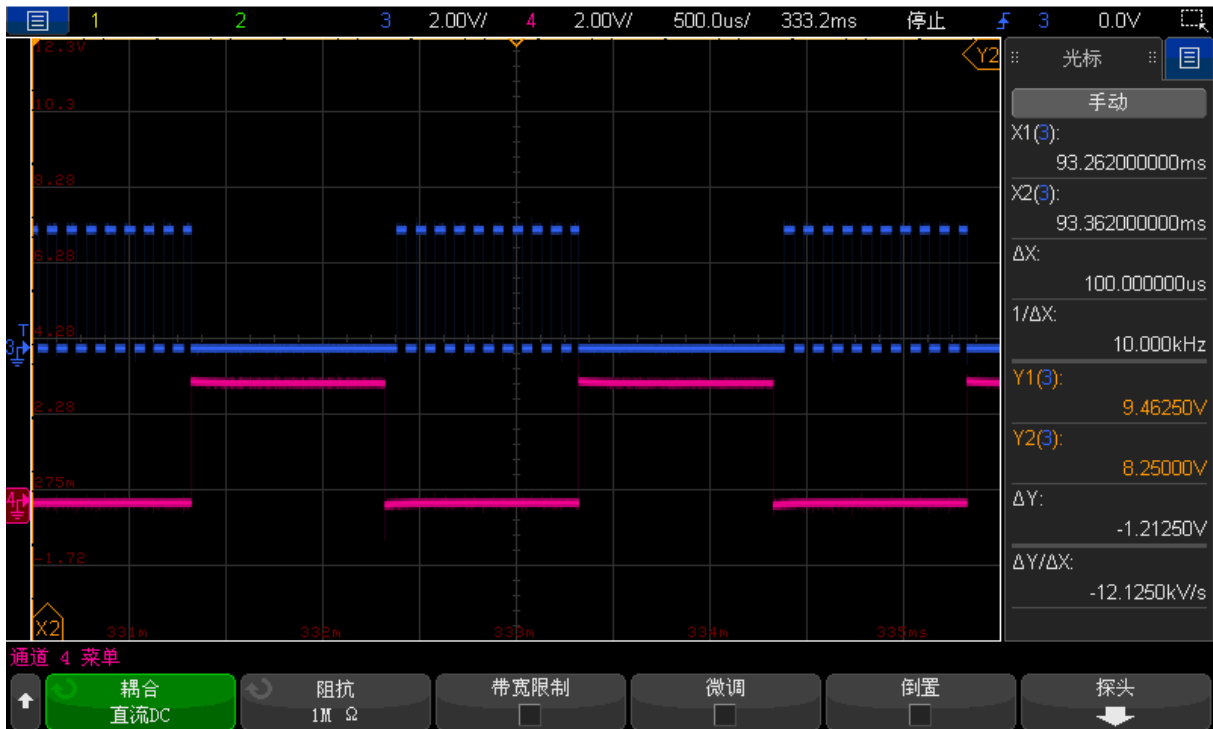


图 3-2：恢复模式波形

当发生过流状况时，ACMP0 输出高电平，EPWM 触发刹车保护。

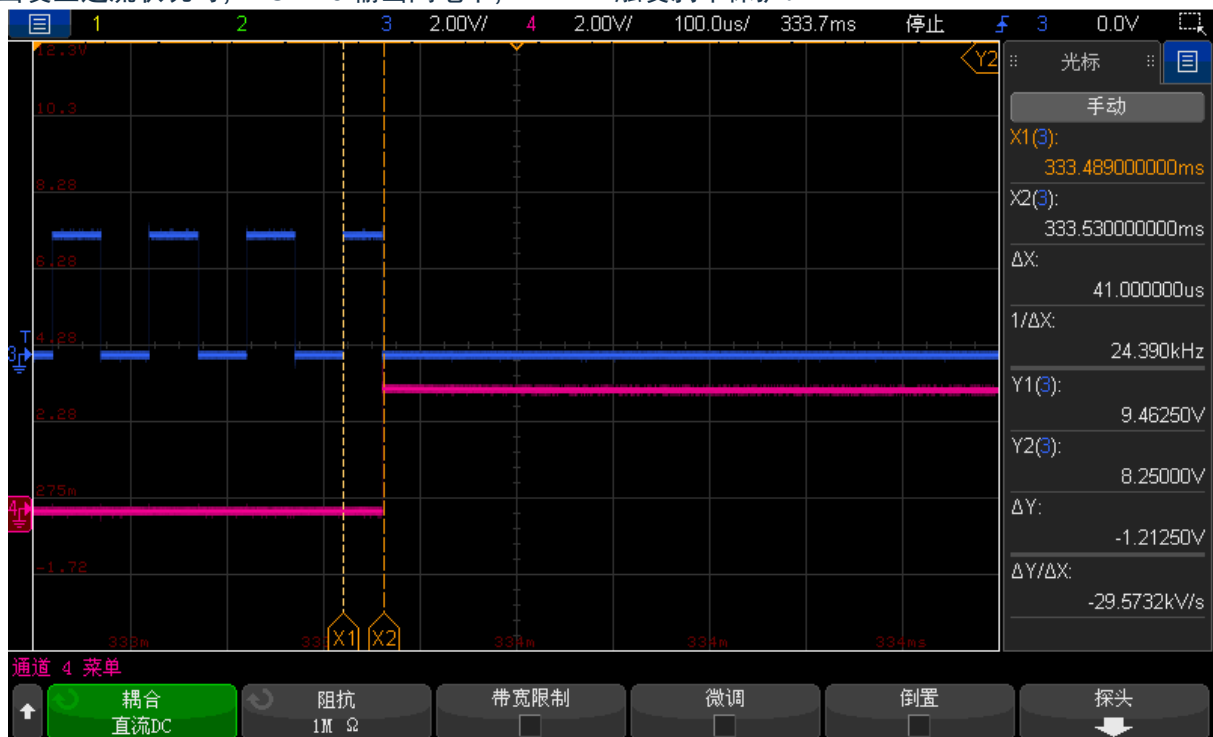


图 3-3：EPWM 刹车保护触发

EPWM0 周期为 100us, 占空比 50%, 高电平时间应为 50us; 由上图 3-3 所示, 当刹车信号触发时, EPWM0 输出会立刻截断, 输出低电平。

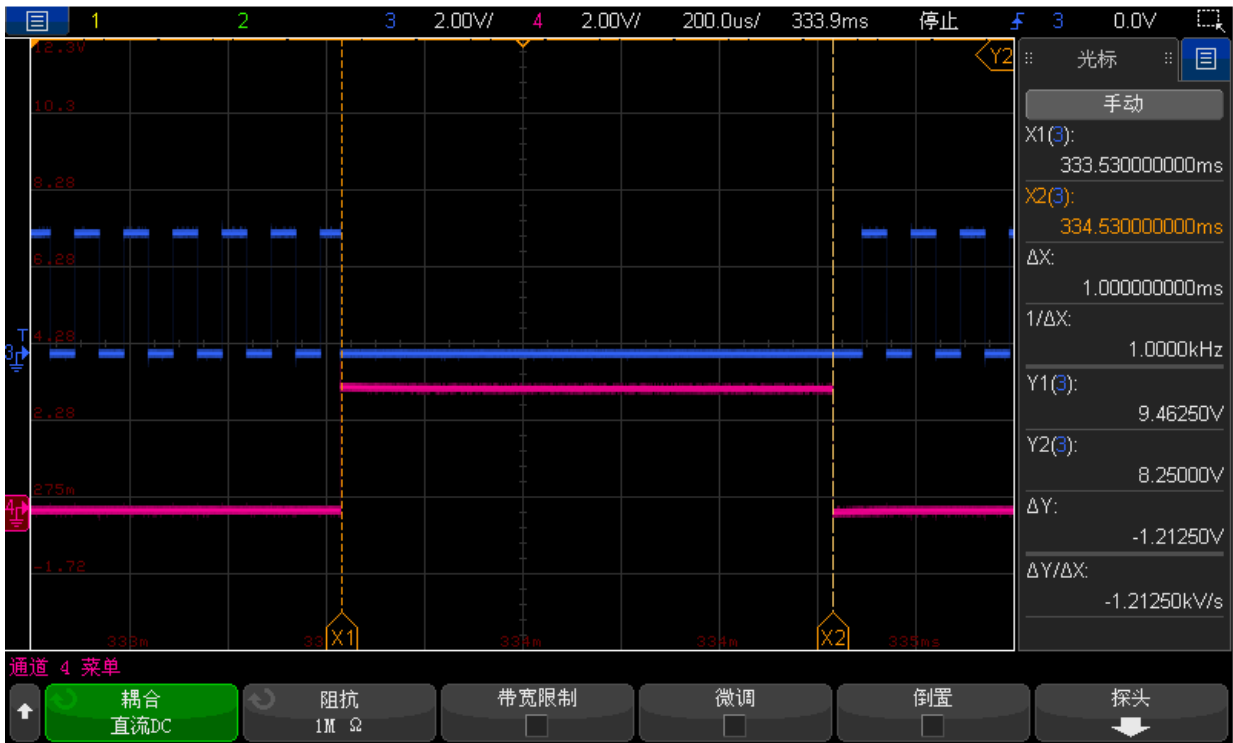


图 3-4：刹车电平持续时间

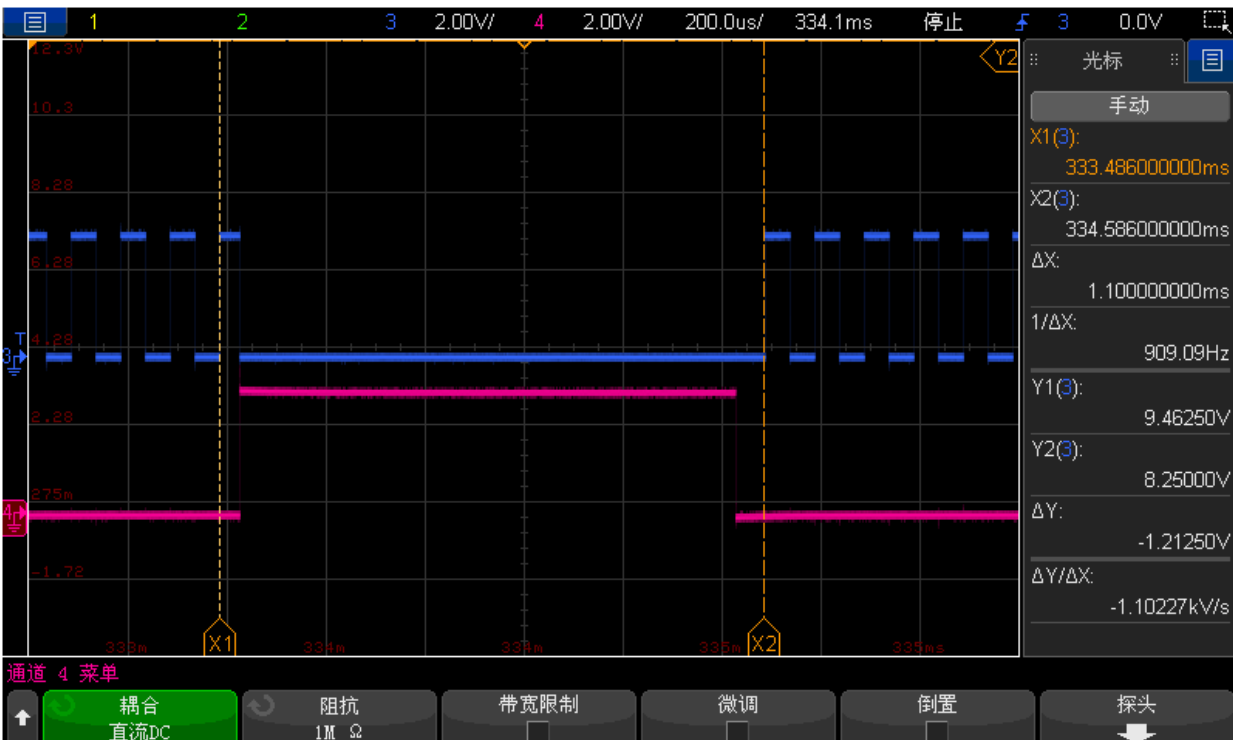


图 3-5：EPWM 输出信号恢复时间

如图 3-4 所示，在刹车信号存在期间，EPWM 保持低电平输出；当刹车信号撤销时，EPWM 在最近加载点恢复输出。

## 4. 更多信息

更多信息，请登录中微半导体网站查看 <http://www.mcu.com.cn>。

## 5. 版本修订说明

版本号	时间	修改内容
V1.00	2021 年 12 月	初始版本