

应用笔记

Application Note

文档编号: **AN1107**

APM32F411 系列 COMP 应用笔记

版本: **V1.0**

1 引言

本应用笔记提供如何在 APM32F411 系列上配置和使用比较器。

APM32F411 内嵌两个可独立使用的比较器 (COMP1 和 COMP2)，其中 COMP1 支持超低功耗运行，COMP2 支持轨对轨输入 (输入电压范围为最小电压到最大电压)、快慢模式，COMP1 和 COMP2 可以组合成一个窗口比较器使用。比较器的输出信号可以重定向至定时器，作为定时器的刹车信号、输入捕获信号或是计数器的外部时钟源使用。

本应用笔记将从以下三种应用场景展示 APM32F411 比较器的用法以及它如何与其他模块配合：

- PWM 控制
- 频率测量
- 窗口比较器

目录

1	引言	1
2	COMP 简介	3
2.1	COMP 输入	3
2.2	COMP 输出	4
2.3	COMP 引脚配置	5
3	COMP 应用程序示例	6
3.1	PWM 控制	6
3.2	频率测量	8
3.3	窗口比较器	10
4	版本历史	12

2 COMP 简介

COMP（比较器）是一种用于比较输入信号并根据比较结果产生相应输出的模块，通常用于检测输入信号的大小或状态。

COMP 具有两个输入端——同相输入端（INP）和反相输入端（INM）。COMP 使能后将开始比较同相输入信号和反相输入信号大小，当同相输入信号大于反相输入信号时，COMP 输出高电平；当同相输入信号小于反相输入信号时，COMP 输出低电平；当同相输入信号等于反相输入信号时，COMP 的输出状态不确定。COMP 的输出可以重定向到其它模块作为输入信号使用。

APM32F411 内嵌有两个 COMP（COMP1 和 COMP2）。在 APM32F411 中，COMP 没有单独的时钟使能控制位，其时钟独立且同步于 PCLK 时钟，比较器只能通过系统复位实现对模块的复位。APM32F411 可以配置 COMP 的输出极性实现输出反相。

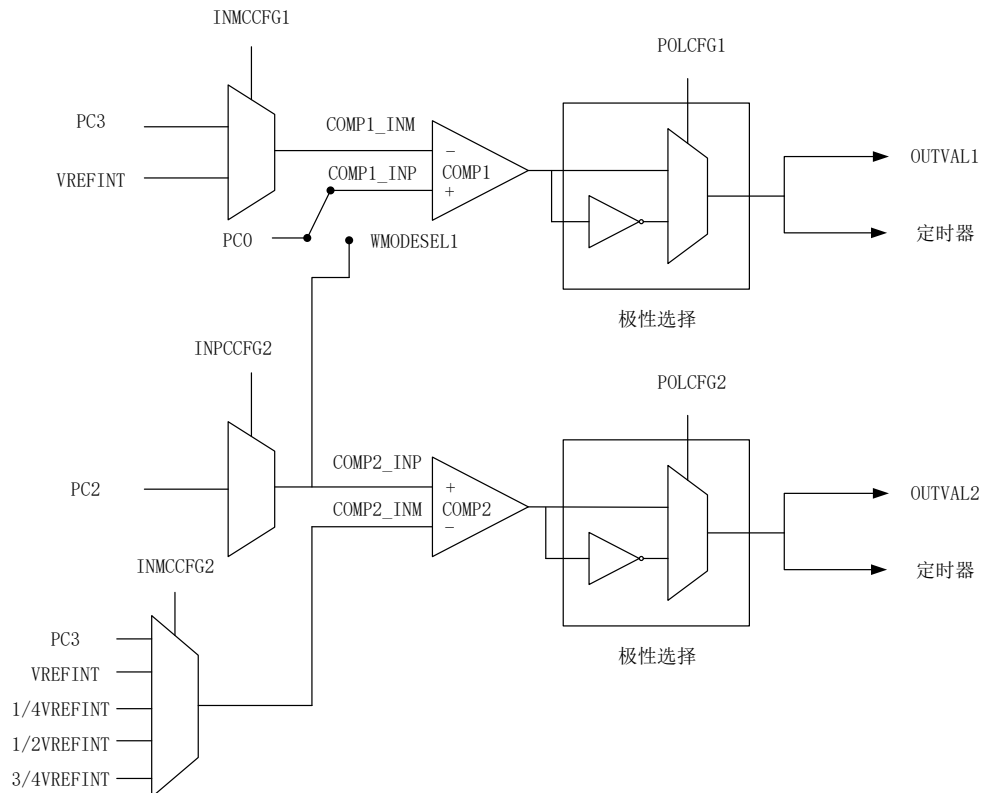


图 2.1 COMP 结构框图

2.1 COMP 输入

COMP1 的同相输入端连接到外部引脚 PC0，反相输入端可连接外部引脚 PC1 或参考电压 VREFINT；COMP2 的同相输入端连接到外部引脚 PC2，反相输入端可连接外部引脚 PC3、参考电压 VREFINT 或参考电压的 1/4、1/2、3/4。

表格 1 COMP 同相输入引脚

模块名	输入端	引脚
COMP1	同相输入	PC0
COMP2	同相输入	PC2

表格 2 COMP 反相输入引脚

模块名	输入端	引脚
COMP1	反相输入	PC1
COMP1	反相输入	VREFINT (参考电压)
COMP2	反相输入	PC3
COMP2	反相输入	VREFINT
COMP2	反相输入	$1/4 * VREFINT$
COMP2	反相输入	$1/2 * VREFINT$
COMP2	反相输入	$3/4 * VREFINT$

2.2 COMP 输出

APM32F411 的 COMP 模块输出可选不重定向到任何设备, 或是重定向到定时器作为定时器刹车信号、输入捕获信号或是计数器的 ETRF (外部参考时钟) 信号使用。

TMR1 (定时器 1)、TMR8 (定时器 8)、TMR3 (定时器 3)、TMR4 (定时器 4) 都是 COMP 输出重定向的可选对象。需要注意, 在 APM32F411 中, 只有高级定时器 TMR1 和 TMR8 有刹车功能, 当 COMP 的输出在作为定时器刹车信号使用时, 只能连接到 TMR1 和 TMR8。

COMP 输出重定向状态共有以下 12 种:

- (1) 无重定向对象
- (2) TMR1 刹车信号
- (3) TMR1 通道 1 输入捕获
- (4) TMR1 外部参考时钟
- (5) TMR8 刹车信号
- (6) TMR8 通道 4 输入捕获

- (7) TMR8 外部参考时钟
- (8) TMR2 通道 4 输入捕获
- (9) TMR2 外部参考时钟
- (10) TMR3 通道 1 输入捕获
- (11) TMR3 外部参考时钟
- (12) TMR4 通道 1 输入捕获

2.3 COMP 引脚配置

APM32F411 的 COMP 模块分别将引脚 PC0 和 PC2 作为同相输入端使用，将 PC1 和 PC3 作为反相输入端使用。

当 COMP 用到这些引脚作输入端时，需要将引脚配置成模拟输入模式，例如再使用外部引脚 PC2 作为 COMP2 的同相输入端时，其配置示例代码如下：

```
void GPIO_Init(void)
{
    GPIO_Config_T gpioConfig;
    RCM_EnableAHB1PeriphClock(RCM_AHB1_PERIPH_GPIOC);

    GPIO_ConfigStructInit(&gpioConfig);
    gpioConfig.mode = GPIO_MODE_AN;
    gpioConfig.pupd = GPIO_PUPD_NOPULL;
    gpioConfig.pin = GPIO_PIN_2;
    GPIO_Config(GPIOC, &gpioConfig);
}
```

3 COMP 应用程序示例

本节描述在示例场景中（PWM 控制、频率测量、窗口比较器等）如何应用 APM32F411 的 COMP 模块。

3.1 PWM 控制

在 APM32F411 中，COMP 的输出可以作为高级定时器的刹车信号使用。

当同相输入大于反相输入时，COMP 输出高电平刹车信号，定时器停止输出 PWM 波；当同相输入小于反相输入时，COMP 输出恢复低电平，定时器在下一个时钟周期重新开始输出 PWM 波。如果配置 COMP 的输出极性反相，那么 COMP 对 PWM 的控制效果会恰好相反。

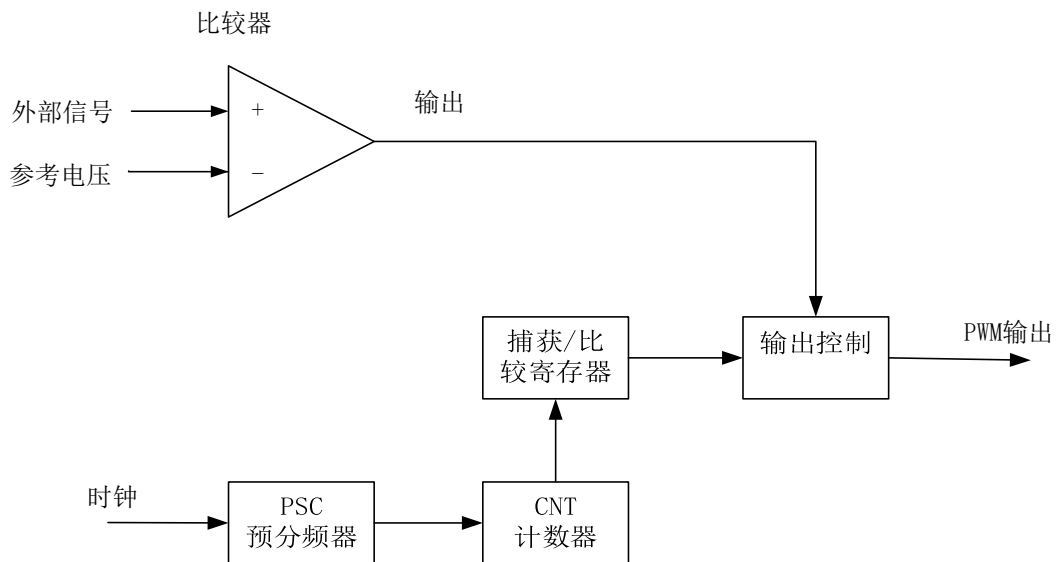


图 3.1 结构图

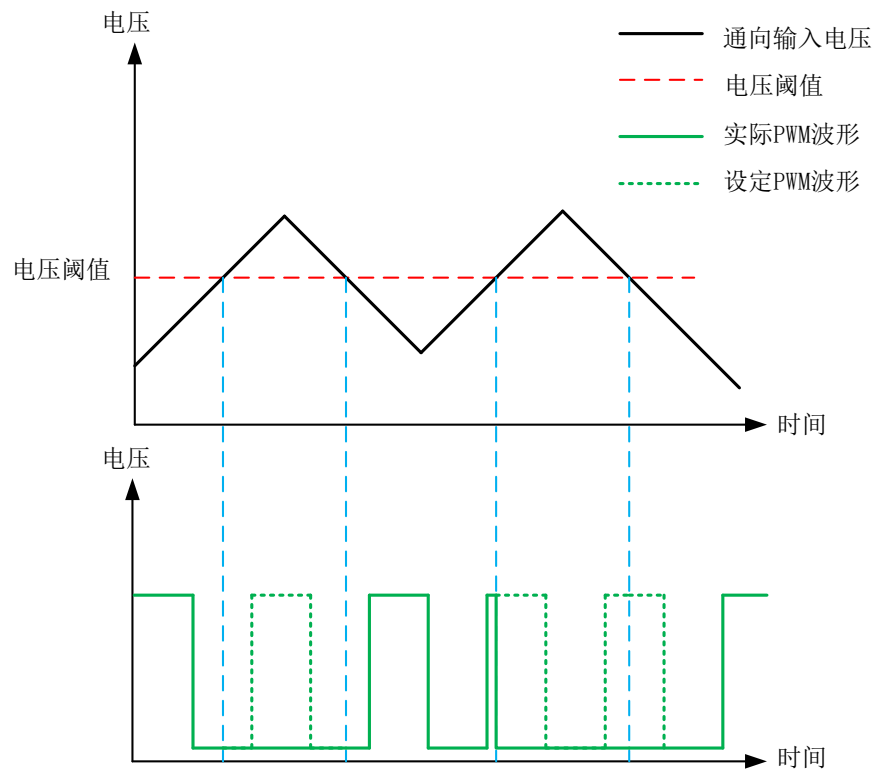


图 3.2 输入电压与 PWM 输出效果

例如将 COMP2 的输出重定向至 TMR1，作为 TMR1 的刹车信号，COMP 的配置代码如下：

```
void COMP_Init(void)
{
    COMP_Config_T compConfig;

    RCM_EnableAPB2PeriphClock(RCM_APB2_PERIPH_SYSCFG);

    COMP_ConfigStructInit(&compConfig);
    compConfig.invertingInput = COMP_INVERTING_INPUT_VREFINT;
    compConfig.mode = COMP_MODE_LOWSPEED;
    compConfig.output = COMP_OUTPUT_TIM1BKIN;
    compConfig.outputPol = COMP_OUTPUTPOL_NONINVERTED;
    COMP_Config(COMP_SELECT_COMP2, &compConfig);

    COMP_Enable(COMP_SELECT_COMP2);
}
```

详细源码可以参见 APM32F4xx_SDK/Example/COMP/COMP_PWMSignalControl。

3.2 频率测量

APM32F411 的 COMP 模块可以测量超过电压阈值部分的频率和脉宽。电压阈值由反相输入端确定。测量频率时需配置定时器为输入捕获模式，测量脉宽则需配置定时器为 PWM 输入捕获模式。

配置定时器为输入捕获模式、输入通道在上升沿时捕获；配置 COMP 的输入重定向至定时器输入捕获通道，输出极性不反相。

当同相输入大于反相输入时，COMP 输出高电平至定时器输入通道，定时器输入通道捕捉到上升沿信号并记录计数器的值至 CC1（通道 1 捕获/比较寄存器），置位定时器 STS（状态寄存器）的 CC1IFLG 位，如果开启了定时器中断，则会进入到定时器中断服务函数；在第二次同相输入大于反相输入，定时器输入通道捕捉到上升沿，再次记录计数器的值至 CC1 以及置位 STS 的 CC1IFLG 位。

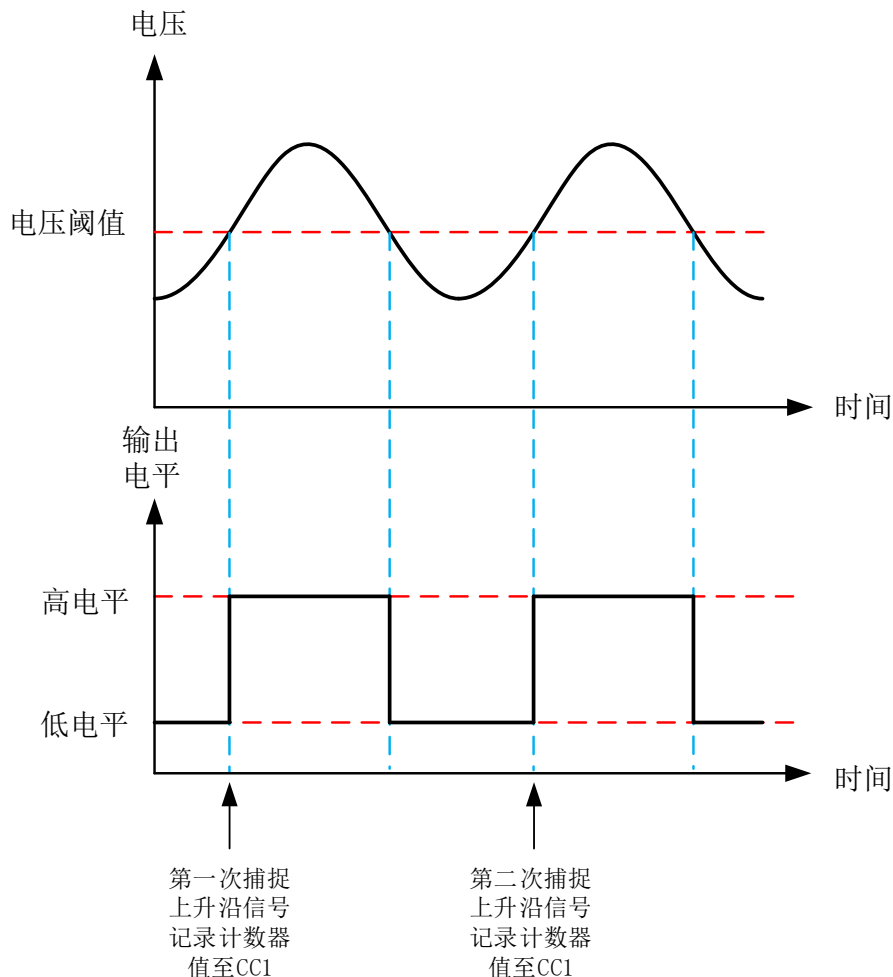


图 3.3 COMP 与定时器频率测量示意图

根据两次记录到 CC1 的值 value[0]和 value[1]、以及在两次采集的间隔中所产生的更新事件次数 update，结合定时器时钟源频率 pclk1，可以算出 COMP 输出波形的频率为：

$$\text{frequency} = \text{pclk1} * 2 / (\text{value}[1] + \text{update} * 0\text{xffff} - \text{value}[0])$$

例如使用 COMP2 与 TMR3（定时器 3）进行频率测量，需将 COMP2 的输出重定向到 TMR3 的输入捕获通道 1 中，COMP 配置示例代码如下：

```
void COMP_Init(void)
{
    COMP_Config_T compConfig;

    /* Enable GPIOC clock */
    RCM_EnableAHB1PeriphClock(RCM_AHB1_PERIPH_GPIOC);

    /* Configure COMP2 */
    COMP_ConfigStructInit(&compConfig);
    compConfig.invertingInput = COMP_INVERTING_INPUT_VREFINT;
    compConfig.mode = COMP_MODE_HIGHSPEED;
    compConfig.output = COMP_OUTPUT_TIM3IC1;
    compConfig.outputPol = COMP_OUTPUTPOL_NONINVERTED;
    COMP_Config(COMP_SELECT_COMP2, &compConfig);

    /* Enable COMP2 */
    COMP_Enable(COMP_SELECT_COMP2);
}

```

TMR3 配置为输入捕获模式，配置示例代码如下：

```
void TMR_Init()
{
    TMR_ICConfig_T ICConfig;

    RCM_EnableAPB1PeriphClock(RCM_APB1_PERIPH_TMR3);

    /* TMR3 Input mode */
    ICConfig.channel = TMR_CHANNEL_1;
    ICConfig.polarity = TMR_IC_POLARITY_RISING;
    ICConfig.selection = TMR_IC_SELECTION_DIRECT_TI;
    ICConfig.prescaler = TMR_IC_PSC_1;
    ICConfig.filter = 0x00;
    TMR_ConfigPWM(TMR3, &ICConfig);
}

```

```

    /* Enable TMR3 Interrupt */
    TMR_Enable(TMR3);
    TMR_EnableInterrupt(TMR3, TMR_INT_UPDATE);
    TMR_EnableInterrupt(TMR3, TMR_INT_CC1);
    TMR_ClearIntFlag(TMR3, TMR_INT_CC1);
    NVIC_EnableIRQRequest(TMR3_IRQn, 0, 0);
}
    
```

3.3 窗口比较器

APM32F411 的 COMP1 和 COMP2 可以组合成一个窗口比较器使用，窗口比较器的使能开关位于 COMP1 的状态控制寄存器中，使能窗口比较器功能会使 COMP1 和 COMP2 的同相输入端短路并连接到外部引脚 PC2 上。

当输入信号从同相输入端传入时，会与 COMP1 和 COMP2 的反相输入进行比较，当同相输入 (INP) 大于 COMP1 的反相输入 (INM_{COMP1}) 时，COMP1 输出高电平；当同相输入大于 COMP2 的反相输入时，COMP2 输出高电平，由此可得，当 COMP1 和 COMP2 的反相输入大小不一致时，窗口比较器会出现 3 种状态。

表格 3 窗口比较器输出状态表

COMP1 输出状态	COMP2 输出状态	说明
0	0	当 $INP < INM_{COMP1}$ 且 $INP < INM_{COMP2}$
1	0	当 $INP > INM_{COMP1}$ 且 $INP < INM_{COMP2}$
1	1	当 $INP > INM_{COMP1}$ 且 $INP > INM_{COMP2}$

可通过 COMP 的输出状态来判断输入信号所处阶段。

COMP1 和 COMP2 开启窗口比较器功能的配置示例代码如下：

```

void COMP_Init(void)
{
    COMP_Config_T    compConfig;

    /* Enable COMP clock */
    RCM_EnableAPB2PeriphClock(RCM_APB2_PERIPH_SYSCFG);

    /* Configure COMP1 */
    COMP_ConfigStructInit(&compConfig);
    compConfig.invertingInput = COMP_INVERTING_INPUT_VREFINT;
}
    
```

```
compConfig.mode = COMP_MODE_LOWSPEED;
compConfig.output = COMP_OUTPUT_NONE;
compConfig.outputPol = COMP_OUTPUTPOL_NONINVERTED;
COMP_Config(COMP_SELECT_COMP1, &compConfig);

compConfig.invertingInput = COMP_INVERTING_INPUT_1_2VREFINT;
COMP_Config(COMP_SELECT_COMP2, &compConfig);

COMP_EnableWindow();

/* Enable COMP */
COMP_Enable(COMP_SELECT_COMP1);
COMP_Enable(COMP_SELECT_COMP2);
}
```

4 版本历史

表格 4 文件版本历史

日期	版本	变更历史
2024.01.31	1.0	新建

声明

本手册由珠海极海半导体有限公司（以下简称“极海”）制订并发布，所列内容均受商标、著作权、软件著作权相关法律法规保护，极海保留随时更正、修改本手册的权利。使用极海产品前请仔细阅读本手册，一旦使用产品则表明您（以下称“用户”）已知悉并接受本手册的所有内容。用户必须按照相关法律法规和本手册的要求使用极海产品。

1、权利所有

本手册仅应当被用于与极海所提供的对应型号的芯片产品、软件产品搭配使用，未经极海许可，任何单位或个人均不得以任何理由或方式对本手册的全部或部分内容进行复制、抄录、修改、编辑或传播。本手册中所列带有“®”或“TM”的“极海”或“Geehy”字样或图形均为极海的商标，其他在极海产品上显示的产品或服务名称均为其各自所有者的财产。

2、无知识产权许可

极海拥有本手册所涉及的全部权利、所有权及知识产权。

极海不应因销售、分发极海产品及本手册而被视为将任何知识产权的许可或权利明示或默示地授予用户。

如果本手册中涉及任何第三方的产品、服务或知识产权，不应被视为极海授权用户使用前述第三方产品、服务或知识产权，除非在极海销售订单或销售合同中另有约定。

3、版本更新

用户在下单购买极海产品时可获取相应产品的最新版的手册。

如果本手册中所述的内容与极海产品不一致的，应以极海销售订单或销售合同中的约定为准。

4、信息可靠性

本手册相关数据经极海实验室或合作的第三方测试机构批量测试获得，但本手册相关数据难免会出现校正笔误或因测试环境差异所导致的误差，因此用户应当理解，极海对本手册中可能出现的该等错误无需承担任何责任。本手册相关数据仅用于指导用户作为性能参数参照，不构成极海对任何产品性能方面的保证。

用户应根据自身需求选择合适的极海产品，并对极海产品的应用适用性进行有效验证和测试，以确认极海产品满足用户自身的需求、相应标准、安全或其它可靠性要求；若因用户未充分对极海产品进行有效验证和测试而致使用户损失的，极海不承担任何责任。

5、合规要求

用户在使用本手册及所搭配的极海产品时，应遵守当地所适用的所有法律法规。用户应了解产品可能受到产品供应商、极海、极海经销商及用户所在地等各国有关出口、再出口或其它法律的限制，用户（代表其本身、子公司及关联企业）应同意并保证遵守所有关于取得极海产品及 / 或技术与直接产品的出口和再出口适用法律与法规。

6、免责声明

本手册由极海“按原样”（as is）提供，在适用法律所允许的范围内，极海不提供任何形式的明示或暗示担保，包括但不限于对产品适销性和特定用途适用性的担保。

对于用户后续在针对极海产品进行设计、使用的过程中所引起的任何纠纷，极海概不承担责任。

7、责任限制

在任何情况下，除非适用法律要求或书面同意，否则极海和/或以“按原样”形式提供本手册的任何第三方均不承担损害赔偿 responsibility，包括任何一般、特殊因使用或无法使用本手册相关信息而产生的直接、间接或附带损害（包括但不限于数据丢失或数据不准确，或用户或第三方遭受的损失）。

8、适用范围

本手册的信息用以取代本手册所有早期版本所提供的信息。

©2024 珠海极海半导体有限公司 - 保留所有权利