

32 位微控制器

HC32F4A0 系列的电压比较器 CMP

F 系列	HC32F4A0
------	----------

目 录

1	摘要	3
2	CMP 简介.....	3
3	HC32F4A0 系列的 CMP	4
3.1	简介	4
3.2	系统框图	5
3.3	功能说明	6
3.3.1	普通比较模式.....	6
3.3.2	窗口比较模式.....	7
3.3.3	定时器窗口输出.....	8
3.4	输出信号处理.....	9
3.4.1	数字滤波	9
3.4.2	输出极性	9
3.4.3	VCOUT 输出	9
3.4.4	中断及外设触发事件	9
3.5	注意事项	10
3.6	寄存器说明	10
4	样例代码	11
4.1	代码介绍	11
4.2	代码运行	13
5	版本信息 & 联系方式	15

1 摘要

本篇应用笔记主要介绍 HC32F4A0 系列芯片的电压比较器（CMP）模块，并通过展示 CMP 样例代码简要说明如何使用 CMP 模块。

2 CMP 简介

电压比较器（Comparator，简称 CMP）是将两个模拟电压进行比较的外设模块，广泛应用于自动控制电路、测量技术、电源电压监测电路等。

3 HC32F4A0 系列的 CMP

3.1 简介

本系列芯片共有 4 个比较器通道 CMP1~CMP4。

本系列芯片 CMP 主要特性：

- CMP1、2 或者 CMP3、4 同时使用可完成窗口比较
- 正端电压和负端电压均有多个输入电压源供选择
- 能使用定时器 PWM 输出作为比较器输出开关信号
- 可产生触发其他外设启动的事件
- 可产生中断并可以唤醒系统停止模式
- 能将比较结果输出到外部管脚 VCOUT

3.2 系统框图

CMP 的系统框图如下图所示：

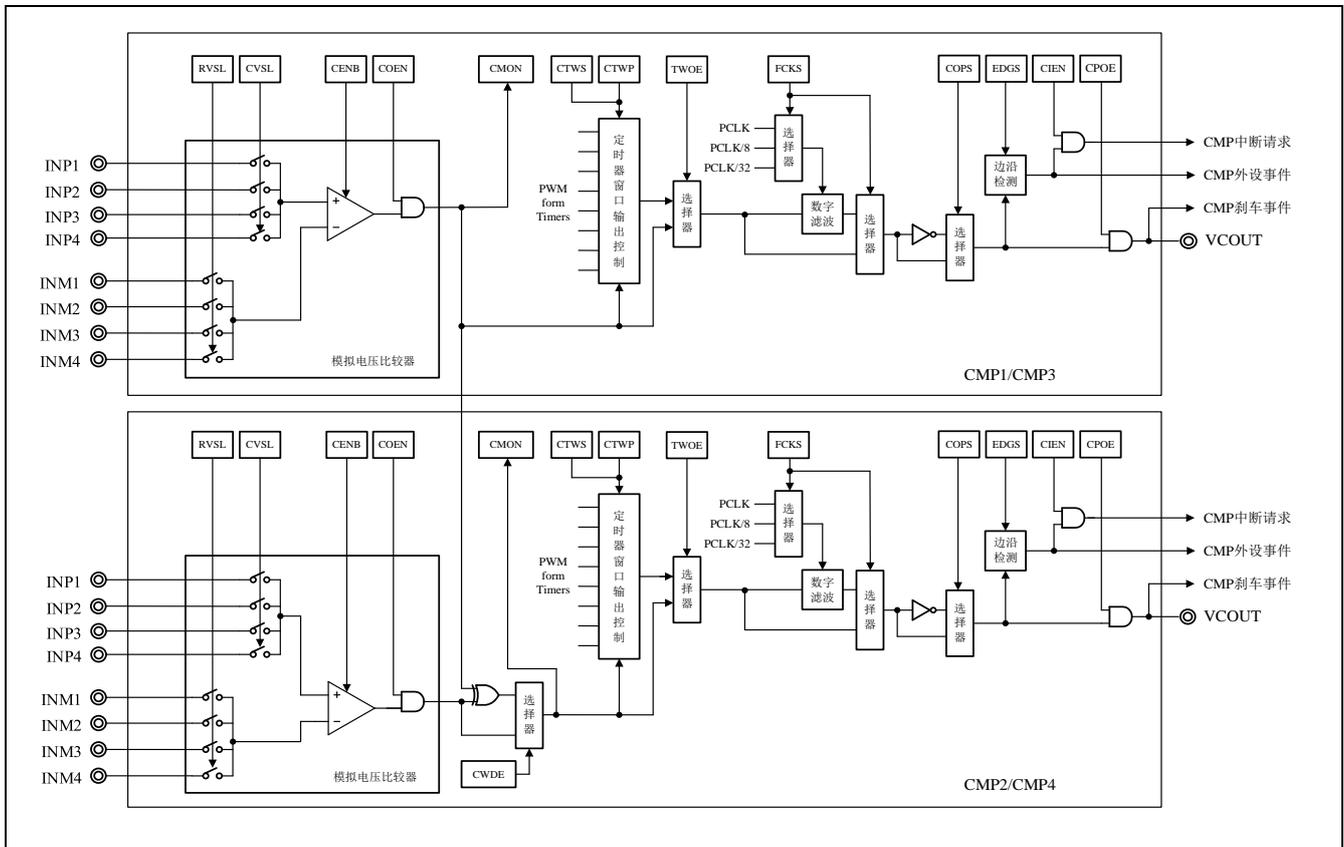


图 3-1 CMP 系统框图

CMP 由寄存器位 RVSL（reference voltage select）选择参考电压（负端电压）来源，由寄存器位 CVSL（compare voltage select）选择比较电压（正端电压）输入源，比较结果放在 CMON（Compare monitor）位，表示电压比较器的比较输出监视信号，该结果是一个实时状态。比较输出监视信号，可以通过 TWOE（Timer windows output enable）等寄存器位来配置是否采用定时器窗口功能；对于比较器的输出监视信号，可以通过 FCKS（Filter clock select）寄存器位来配置是否进行数字滤波；对于比较器的输出监视信号，可以通过 COPS（Compare output polarity select）寄存器位来配置是否反向输出。通过以上功能电路（定时器窗口输出、滤波、极性控制）的比较器输出信号可以通过 VCOUNT 端口输出；也可以通过寄存器位 EDGS（Edge select）选择产生中断或事件的有效边沿，产生相应的中断或事件。

CMP2/4 通道特有寄存器位 CWDE（Compare window mode）来选择是否采用窗口比较模式。

3.3 功能说明

本系列芯片的 CMP 可以配置为普通比较模式和窗口比较模式。

3.3.1 普通比较模式

比较器 CMP1~4 在普通比较模式下可以分别单独使用。普通比较模式实现输入比较电压（正端电压）与参考电压（负端电压）之间的比较并且输出比较结果，工作示意图如下图所示：

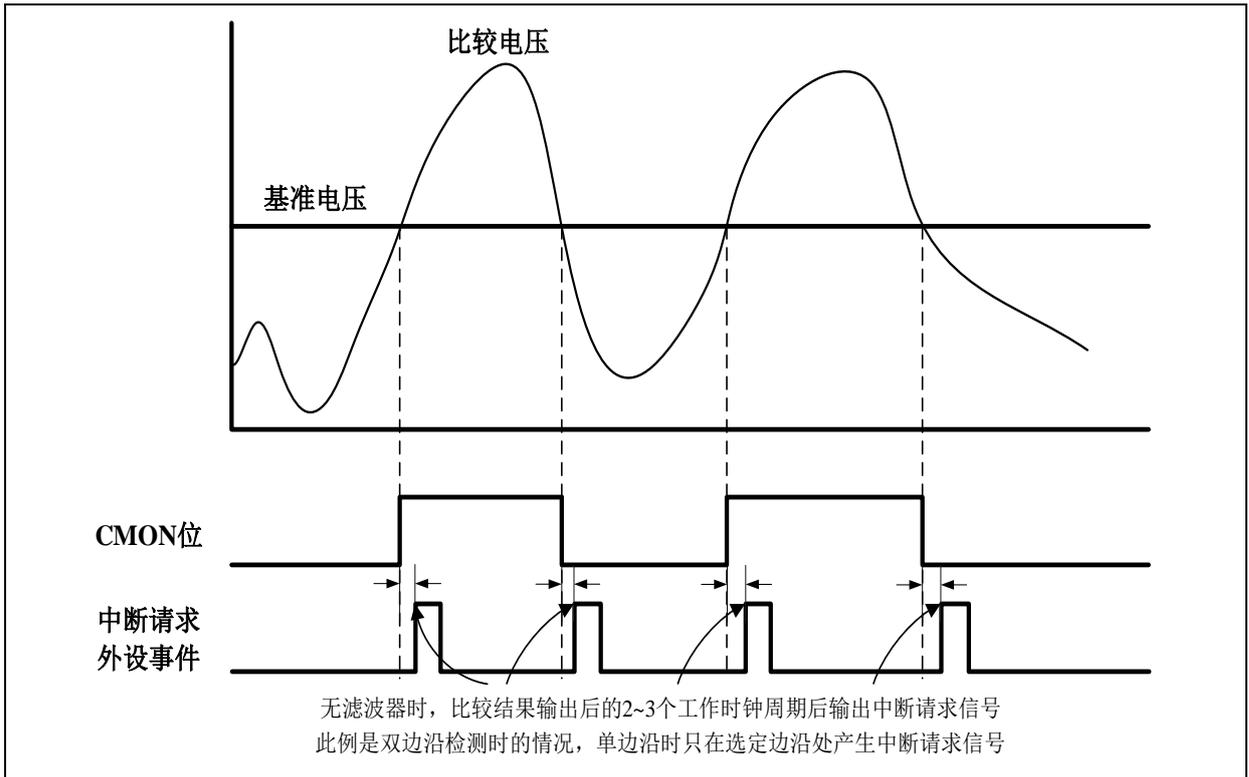


图 3-2 普通比较模式工作示意图

上图所示是在普通比较模式下，选择双边沿检测。当比较电压（正端电压）高于基准电压（负端电压）时，CMON 由 0 变为 1，产生上升沿，产生外设请求事件。当比较电压（正端电压）低于基准电压（负端电压）时，CMON 由 1 变为 0，产生下降沿，也产生外设请求事件。在单边沿检测时，只在选定边沿处产生外设请求事件。

3.3.2 窗口比较模式

窗口比较模式实现对输入电压是否在窗口电压范围的比较判断并且输出比较结果，其工作示意图如下图所示：

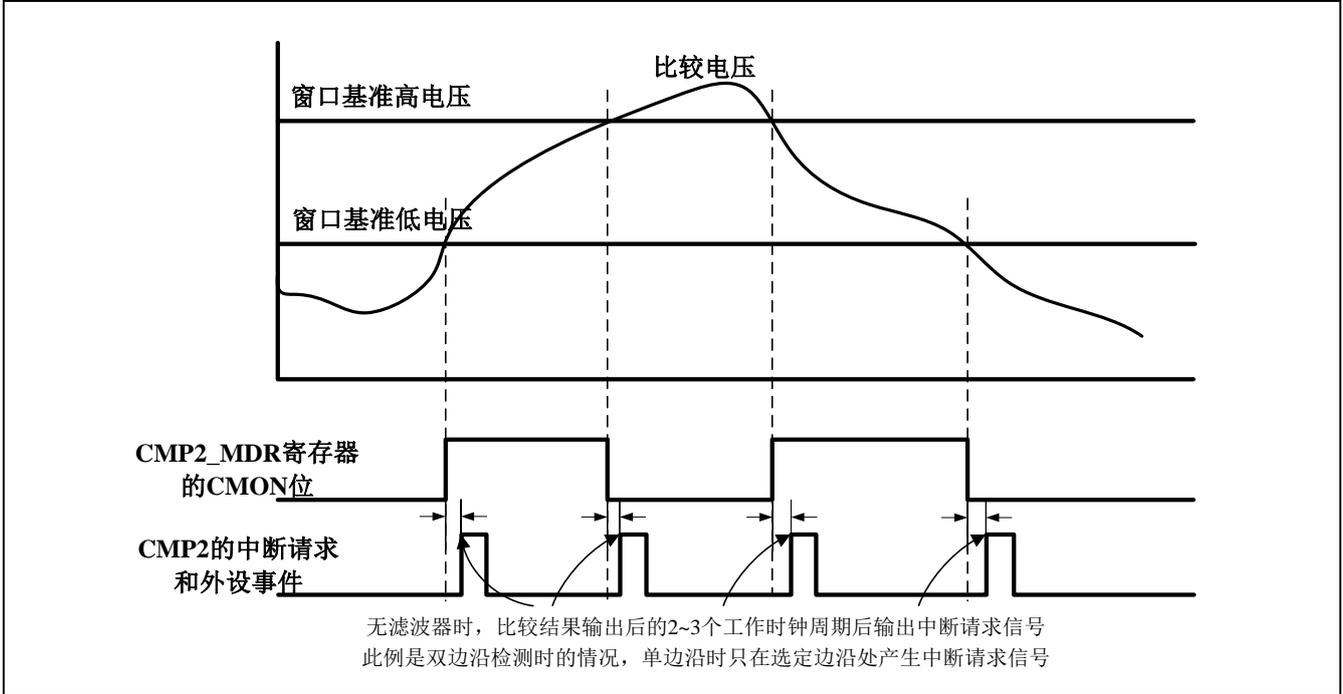


图 3-3 窗口比较模式工作示意图

窗口比较模式使用 CMP1、2 或者 CMP3、4 两个比较器来实现。例如选择 CMP1 和 CMP2 进行窗口电压比较，CMP1 和 CMP2 的正端电压选择需要同一比较电压（CMP2_INP3），负端电压分别选择窗口电压的上限值和下限值。当正端的比较电压在窗口范围内，CMP2 的 CMON 位由 0 变为 1。当正端的比较电压在窗口范围之外，CMP2 的 CMON 位由 1 变为 0。在选定的边沿会产生外设请求事件。

3.3.3 定时器窗口输出

定时器窗口输出功能使能时，电压比较器的输出除了受到 COEN 位控制外，还受到 CMP_TWSR 寄存器选择的 PWM 信号控制。可以通过 CMP_TWPR 寄存器设定允许输出时定时器窗口信号的定时器极性，并通过 CMP_OCR 寄存器的 TWOL 位设定禁止输出时定时器窗口输出端的固定电平。定时器窗口输出如图 3-4 所示：

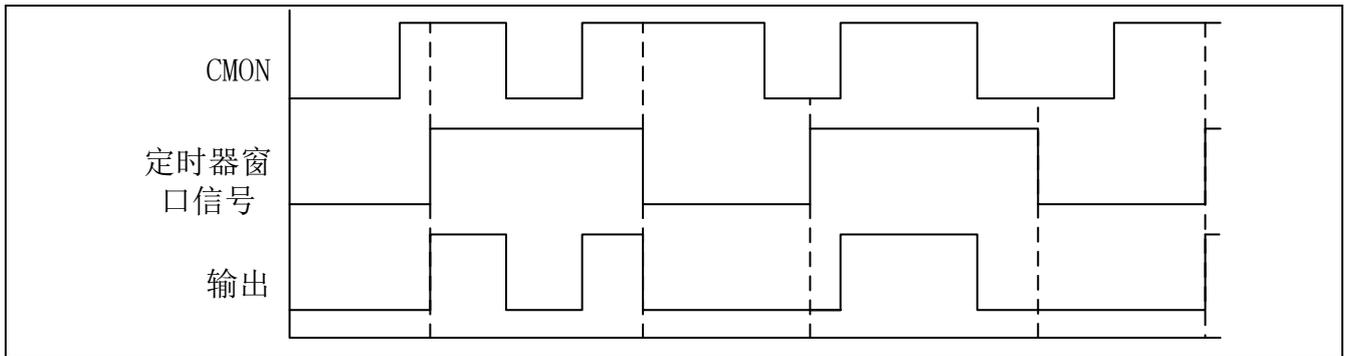


图 3-4 定时器窗口输出模式

上图所示在 PWM 输出高电平时，CMP 输出有效。在 PWM 输出低电平时，即使 CMON 为 1，CMP 输出依旧是低电平。定时器 TMR4、TMR6、TMEA 产生的 PWM 可作为 CMP 的窗口输出信号，具体 PWM 信号请参照用户手册表 16-5。

3.4 输出信号处理

本系列芯片的 CMP 外设对模拟电压比较器输出的比较监测结果 CMON 通过以下电路对输出结果进行加工处理。输出信号处理电路后的 CMP 比较输出用于 VCOUT 的输出和产生中断及外设触发事件。

3.4.1 数字滤波

数字滤波器对电压比较器的输出信号进行噪声滤波，数字滤波器按照采样时钟对电压比较器的输出进行采样，在三次采样电平相同后的下一个采样时钟将采样值输出，否则保持输出不变。采样时钟可以通过 CMPn_FIR 寄存器的 FCKS 寄存器位进行配置。

3.4.2 输出极性

通过寄存器 CMPn_OCR 的寄存器位 COPS 可以对比较器的输出进行极性翻转。

3.4.3 VCOUT 输出

通过寄存器 CMPn_OCR 的寄存器位 CPOE (Compare port output enable) 可以使能或禁止 VCOUT 管脚输出经过输出处理的比较器输出结果。在使能 VCOUT 输出同时还应该通过端口寄存器将 VCOUT 端口设置为 VCOUT 输出功能，详细配置方法请参考本系列芯片数据手册引脚配置和引脚功能章节。

3.4.4 中断及外设触发事件

通过寄存器 CMPn_FIR 的寄存器位 EDGS 选择产生中断及外设事件的边沿，可以配置为上升沿、下降沿和双边沿。

寄存器 CMPn_FIR 的寄存器位 CIEN (Compare interrupt enable) 使能 CMP 中断。当 CMP 用于唤醒停止低功耗模式时，将数字滤波和边沿检测功能关闭，同时通过 CIEN 使能 CMP 中断。如此配置后，中断将在输出为高电平时发生并唤醒停止低功耗模式。回到工作模式后，请先设定中断无效 (CIEN=0) 后再进行后续操作。

3.5 注意事项

使用 CMP 模块时，需要注意以下几点：

- 1) 对于 CMP1, CMP3, 如果选择 INP2 或者 INP3 作为正端电压，还需要通过设定 CMP_VISR 选择电压输入源。具体参考用户手册表 16-3。
- 2) CMP 的寄存器某些标志位在有写操作时会引起中断或外设触发事件，因此请在中断禁止或外设触发功能无效状态下设定该寄存器。寄存器设定好后，清除相应中断标志。

3.6 寄存器说明

以下为 CMP 模块的寄存器列表，详细寄存器说明请查看本系列芯片用户手册相关章节。

CMP1 基准地址：0x4000A000

CMP2 基准地址：0x4000A010

CMP3 基准地址：0x4000A400

CMP4 基准地址：0x4000A410

寄存器名	符号	偏移地址	位宽	复位值
比较器模式设定寄存器	CMPx_MDR, x=1~4	0x00	8	0x00
比较器滤波控制寄存器	CMPx_FIR, x=1~4	0x01	8	0x00
比较器输出控制寄存器	CMPx_OCR, x=1~4	0x02	8	0x00
比较器正负端输入选择寄存器	CMPx_PMSR, x=1~4	0x03	8	0x00
比较器定时器窗口选择寄存器	CMPx_TWSR, x=1~4	0x04	16	0x0000
比较器定时器窗口极性寄存器	CMPx_TWPR, x=1~4	0x06	16	0x0000
比较器电压输入源选择寄存器	CMPx_VISR, x=1, 3	0x08	16	0x0000

4 样例代码

4.1 代码介绍

用户可写自己的代码来学习验证该模块，也可以直接通过华大半导体的网站下载到设备驱动库（Device Driver Library, DDL）的样例代码并使用其中的 CMP 的样例进行验证。

以下部分简要介绍基于 DDL 的 CMP 模块样例（cmp_timerwindow_output）代码所涉及的各项配置。

- 1) 解锁 GPIO、CLK、PWC、EFM 等寄存器的写保护

```
/* Unlock peripherals or registers */  
Peripheral_WE();
```

- 2) 配置系统时钟和 PCLK3 时钟

```
/* Configure system clock. HCLK = 240MHZ PCLK3 = 60MHZ*/  
BSP_CLK_Init();
```

- 3) 配置 TMRA 的 PWM 输出

```
/* Enable TMRA peripheral clock. */  
PWC_Fcg2PeriphClockCmd(TMRA_PERIP_CLK, Enable);  
/* Set a default initialization value for stcInit. */  
(void)TMRA_StructInit(&stcTmraInit);  
/* Modifies the initialization values depends on the application. */  
stcTmraInit.u32ClkSrc = TMRA_CLOCK;  
stcTmraInit.u32PCLKDiv = TMRA_PCLK_DIV1024;  
stcTmraInit.u32CntDir = TMRA_DIR;  
stcTmraInit.u32CntMode = TMRA_MODE;  
stcTmraInit.u32PeriodVal = TMRA_PERIOD_VAL;  
(void)TMRA_Init(TMRA_UNIT, &stcTmraInit);  
/* Set compare value */  
TMRA_SetCmpVal(TMRA_UNIT, TMRA_CH, TMRA_PWM_A_CMP_VAL);  
GPIO_SetFunc(TMRA_PWM_A_PORT, TMRA_PWM_A_PIN, TMRA_PWM_A_PIN_FUNC,  
PIN_SUBFUNC_DISABLE);  
GPIO_OE(TMRA_PWM_A_PORT, TMRA_PWM_A_PIN, Enable); /* GPIO_E_09 output Enable */  
(void)TMRA_PWM_StructInit(&stcTrmaPwmCfg);  
(void)TMRA_PWM_Config(TMRA_UNIT, TMRA_CH, &stcTrmaPwmCfg);  
/* Enable PWM output */  
TMRA_PWM_Cmd(TMRA_UNIT, TMRA_CH, Enable);  
/* Starts TIMERA. */  
TMRA_Start(TMRA_UNIT);
```

4) 配置 CMP，选择正负端电压，检测边沿以及定时器窗口输出

```

/* Enable peripheral Clock */
PWC_Fcg3PeriphClockCmd(PWC_FCG3_CMBIAS, Enable);
PWC_Fcg3PeriphClockCmd(CMP_PERIP_CLK, Enable);
/* Port function configuration for CMP*/
(void)GPIO_StructInit(&stcGpioInit);
stcGpioInit.u16PinAttr = PIN_ATTR_ANALOG;
(void)GPIO_Init(CMP1_INP3_PORT, CMP1_INP3_PORT, &stcGpioInit);
(void)GPIO_Init(CMP1_INM4_PORT, CMP1_INM4_PIN, &stcGpioInit);
GPIO_SetFunc(CMP_VCOUT1_PORT, CMP_VCOUT1_PIN, GPIO_FUNC_1,
PIN_SUBFUNC_DISABLE);
/* Clear structure */
(void)CMP_StructInit(&stcCmpInit);
/* Configuration for normal compare function */
/*For CMP1 and CMP3, when selecting channels inp2 and inp3, there are
variety of compare voltage available, so you need to configure channels
and voltage sources, and for others, you only need to configure channels*/
stcCmpInit.u8CmpCh      = CMP_CVSL_INP3;
stcCmpInit.u16CmpVol    = CMP1_INP3_CMP1_INP3;
stcCmpInit.u8RefVol     = CMP_RVSL_INM4;
stcCmpInit.u8OutDetectEdges = CMP_DETECT_EDGS_BOTH;
stcCmpInit.u8OutFilter   = CMP_OUT_FILTER_PCLK3_DIV32;
stcCmpInit.u8OutPolarity = CMP_OUT_REVERSE_OFF;
(void)CMP_NormalModeInit(CMP_TEST_UNIT, &stcCmpInit);
/* Configuration for timer window function */
stcCmptimerWindow.u16TWSelect = CMP_TW_PWM;
stcCmptimerWindow.u8TWInvalidLevel = CMP_TIMERWIN_INVALID_LEVEL_LOW;
stcCmptimerWindow.u8TWOutLevel = CMP_TIMERWIN_OUT_LEVEL_HIGH;
(void)CMP_TimerWindowConfig(CMP_TEST_UNIT, &stcCmptimerWindow);
/* Enable CMP output */
CMP_OutputCmd(CMP_TEST_UNIT, Enable);
/* Enable VCOUT */
CMP_VCOUTCmd(CMP_TEST_UNIT, Enable);
}

```

4.2 代码运行

用户可以通过华大半导体的网站下载到 HC32F4A0 的 DDL 的样例代码

（cmp_timerwindow_output），并配合评估用板（EV_F4A0_LQ176_V10）运行相关代码学习使用 CMP 模块。

以下部分主要介绍如何在评估板上运行 cmp_timerwindow_output 样例代码并观察结果：

- 确认安装正确的 IAR EWARM v7.7 工具（请从 IAR 官方网站下载相应的安装包，并参考用户手册进行安装）。
- 从华大半导体网站下载 HC32F4A0 DDL 代码。
- 下载并运行 cmp_timerwindow_output \ 中的工程文件：
 - 1) 打开 cmp_timerwindow_output \ 工程，并打开 ‘main.c’ 如下视图：

```

188  /**
189  * @brief Main function of cmp_timerwindow_output project
190  * @param None
191  * @retval int32_t return value, if needed
192  */
193  int32_t main(void)
194  {
195      /* Unlock peripherals or registers */
196      Peripheral_WE();
197      /* Configure system clock. HCLK = 240MHZ PCLK3 = 60MHZ*/
198      BSP_CLK_Init();
199      #if (CMP_TW_PWM_SOURCE == CMP_TW_PWM_TMRA)
200          /* Configure TMRA */
201          TMRAConfig();
202      #elif (CMP_TW_PWM_SOURCE == CMP_TW_PWM_TMR4)
203          /* Configure TMR4 */
204          TMR4Config();
205      #elif (CMP_TW_PWM_SOURCE == CMP_TW_PWM_TMR6)
206          /* Configure TMR6 */
207          TMR6Config();
208      #endif
209      /* Configure CMP */
210      CmpConfig();
211      /* Lock peripherals or registers */
212      Peripheral_WP();
213      for (;;)
214      {
215          ;
216      }
217  }
218
219  /**
220  * @brief Configure CMP.
221  * @param None
222  * @retval None
223  */
224  static void CmpConfig(void)
225  {
226      stc_cmp_init_t stcCmpInit;
227      stc_cmp_timerwindow_t stcCmptimerWindow;
228      stc_gpio_init_t stcGpioInit;
229      /* Enable peripheral Clock */
230      PWC_Fcg3PeriphClockCmd(PWC_FCG3_CMBIAS, Enable);
231      PWC_Fcg3PeriphClockCmd(CMP_PERIP_CLK, Enable);
    
```

- 2) 点击  重新编译整个项目；

3) 硬件连接：芯片引脚 PC3 接参考电压，引脚 PE10 接比较电压，示波器连接 PWM 输出引脚和 CMP_VCOOUT1(PB12)输出引脚；

4) 点击  将代码下载到评估板上，全速运行；

5) 调节输入比较电压值，示波器查看 VCOOUT1(PB12)输出是否随 PWM 变化。

当比较电压>参考电压，PWM 输出高电平时，CMP 输出有效且输出高电平。

当比较电压>参考电压，PWM 输出低电平时，CMP 输出无效。

当比较电压<参考电压，PWM 输出高电平时，CMP 输出有效且输出低电平。

当比较电压<参考电压，PWM 输出低电平时，CMP 输出无效。

5 版本信息 & 联系方式

日期	版本	修改记录
2021/7/27	Rev1.0	初版发布



如果您在购买与使用过程中有任何意见或建议，请随时与我们联系。

Email: mcu@hdsc.com.cn

网址: www.hdsc.com.cn

通信地址: 上海市浦东新区中科路 1867 号 A 座 10 层

邮编: 201203

