

FT62F08X

Application note

目录

1. 高级定时器 TIM1	4
1.1. 特性	4
1.2. 原理框图	5
1.3. 功能描述	5
1.3.1. 计数基本单元	6
1.3.1.1. 计数时钟预分频	6
1.3.1.2. 向上计数模式	7
1.3.1.3. 向下计数模式	7
1.3.1.4. 中心对齐模式	7
1.3.1.5. 重复向下计数器	8
1.3.2. 计数控制器	9
1.3.2.1. 计数器时钟源	10
1.3.2.2. 计数触发源	10
1.3.2.3. slave 模式控制	10
1.3.3. 捕捉比较通道	12
1.3.3.1. 捕捉输入通道	13
1.3.3.2. 输出比较通道	14
1.3.3.3. 死区产生	15
1.3.3.4. 输出控制	15
1.3.4. TIM1 中断	16
1.3.5. 故障刹车源	17
1.3.6. 前沿消隐	18
1.4. 与 TIMER1 相关寄存器汇总	19
1.4.1. TIM1CR1, 地址: 0x211	20
1.4.2. TIM1SMCR, 地址: 0x213	21
1.4.3. TIM1IER, 地址: 0x215	22
1.4.4. TIM1SR1, 地址: 0x216	22
1.4.5. TIM1SR2, 地址: 0x217	23
1.4.6. TIM1EGR, 地址: 0x218	24
1.4.7. TIM1CCMR1, 地址: 0x219	25
1.4.8. TIM1CCMR2, 地址: 0x21A	26
1.4.9. TIM1CCMR3, 地址: 0x21B	27
1.4.10. TIM1CCMR4, 地址: 0x21C	28
1.4.11. TIM1CCER1, 地址: 0x21D	29
1.4.12. TIM1CCER2, 地址: 0x21E	30
1.4.13. TIM1CNTRH, 地址: 0x28C	30
1.4.14. TIM1CNTRL, 地址: 0x28D	30
1.4.15. TIM1PSCRH, 地址: 0x28E	31
1.4.16. TIM1PSCRL, 地址: 0x28F	31
1.4.17. TIM1ARRH, 地址: 0x290	31
1.4.18. TIM1ARRL, 地址: 0x291	32
1.4.19. TIM1RCR, 地址: 0x292	32



1.4.20.	TIM1CCR1H, 地址: 0x293	32
1.4.21.	TIM1CCR1L, 地址: 0x294	33
1.4.22.	TIM1CCR2H, 地址: 0x295	33
1.4.23.	TIM1CCR2L, 地址: 0x296	33
1.4.24.	TIM1CCR3H, 地址: 0x297	33
1.4.25.	TIM1CCR3L, 地址: 0x298	34
1.4.26.	TIM1CCR4H, 地址: 0x299	34
1.4.27.	TIM1CCR4L, 地址: 0x29A	34
1.4.28.	TIM1BRK, 地址: 0x29B	35
1.4.29.	TIM1DTR, 地址: 0x29C	36
1.4.30.	TIM1OISR, 地址: 0x29D	36
1.4.31.	LEBCON 寄存器, 地址 0x41C	37

2 应用范例	37
--------------	----

62F08x_TIM1_CAPTURE 的应用

1. 高级定时器 TIM1

1.1. 特性

- 16bit 的向上计数、向下计数或者上/下计数器，支持自动重载；
- 支持可编程预分频的计数时钟；
- 支持 4 个独立的捕捉比较通道，通道可支持：
 - 输入捕捉
 - 输出比较
 - 边沿或中心对称 PWM
 - 单脉冲输出
 - 6 步 PWM
- PWM 互补输出和可编程死区时间；
- 支持多 timer 同步工作；
- 可编程的重复计数器；
- 刹车功能，使输出停止在一个复位态或者一个预设状态
- 中断事件：
 - 更新事件：计数器溢出，计数器初始化
 - 触发事件：触发计数开始与停止，计数器初始化或外部触发事件
 - 输入捕捉事件
 - 输出比较事件
 - 刹车输入有效事件
- 外部时钟的触发计数
- 前沿消隐

1.2. 原理框图

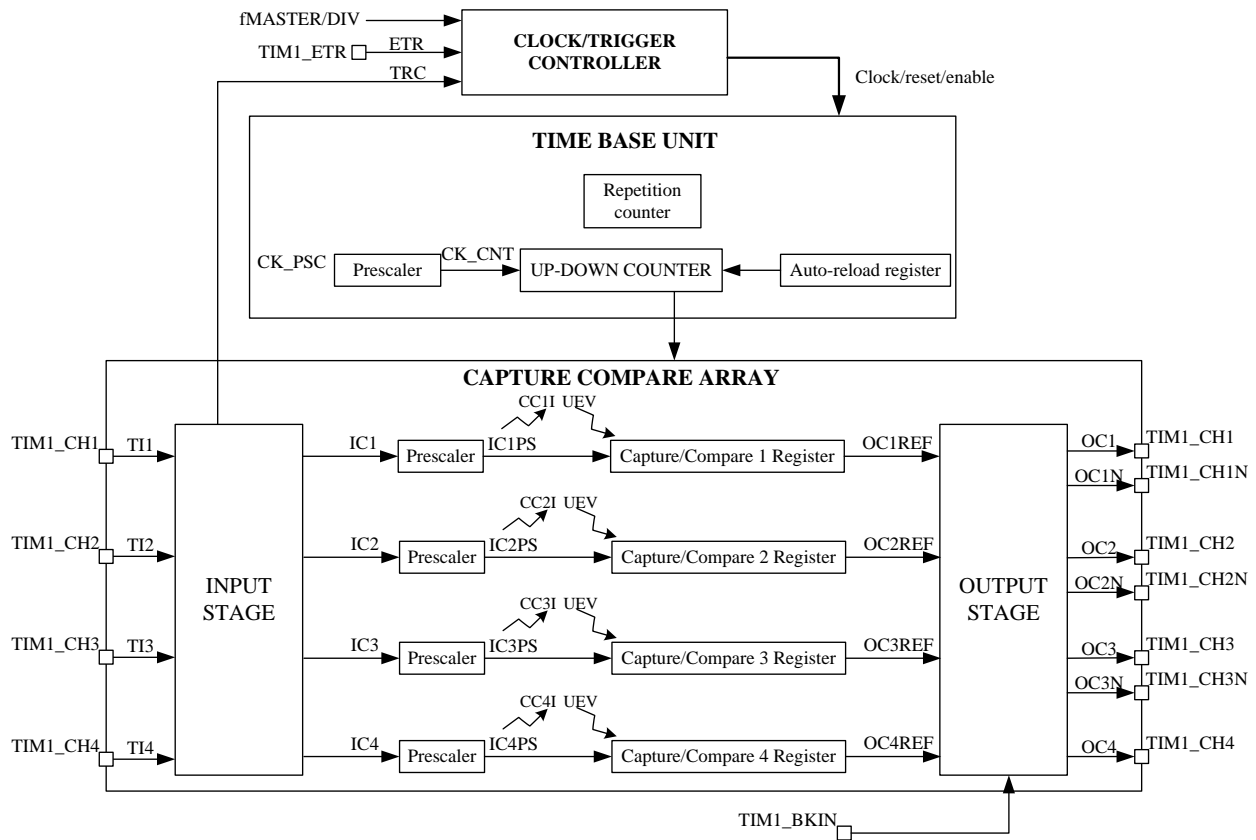


图 10.1 Timer1 原理框图

1.3. 功能描述

整个 Timer1 可以分为三个大的功能部分：计数基本单元、计数控制和捕捉比较通道。计数基本单元分为向上/向下计数器、自动加载寄存器、重复计数器和预分频器；计数控制器又分为计数触发源，模式控制；捕捉比较通道分为捕捉输入通道，输出比较通道，死区产生和输出控制。

1.3.1. 计数基本单元

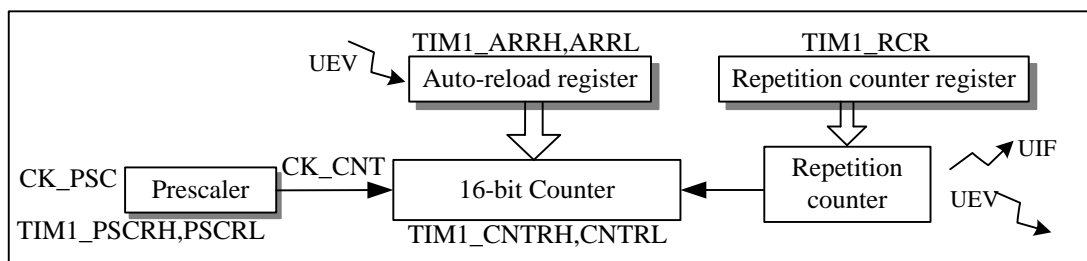


图 10.2 计数基本单元

16 位计数器，预分频器，自动重载寄存器和重复计数寄存器都能由软件进行读写。

自动重载寄存器由一个预加载寄存器和一个影子寄存器组成。

写自动重载寄存器的两种方式：

- 自动重载预加载使能(T1ARPE=1)。在这种模式下，写入自动重载寄存器的数值保存在预加载寄存器中，并在下一个更新事件到来时传送到影子寄存器中进行使用。
- 自动重载预加载关闭(T1ARPE=0)。在这种模式下，写入自动重载寄存器的数值直接传送到影子寄存器中进行使用。

更新事件的产生：

- 计数器上溢或下溢
- 通过软件设置 TIM1_EGR 寄存器中的 T1UG 位
- 设置为复位模式时，触发事件的到来

更新事件的影响：

- 某些预加载的寄存器(具体寄存器可查看寄存器表格)在预加载使能的情况下都能被更新为最新值
- 更新标志位(T1UIF)被置位(需要根据 T1URS 的值)
- 自动重载的影子寄存器被更新为 TIM1_ARR 寄存器中的值

16 位计数器的读写：

- TIM1_CNTRH/L 能在任何时候进行写操作；但是建议为了避免出现不正确的中间状态，不要在计数器运行的时候进行写操作
- TIM1_CNTRH/L 的写操作是没有顺序限制的；可以先写高位也可以先写低位
- TIM1_CNTRH/L 能在任何时候进行读操作；但是因为此设计是异步设计，所以在计数器运行期间进行读操作可能读出不正确的数值，需要读两次，比较两次数值是否一致；如果一致，则读出的数值是正确的数值；否则，读出数值是错误的。

1.3.1.1. 计数时钟预分频

计数时钟可以进行 16bit 的时钟预分频：

$$f_{CK_CNT} = f_{CK_PSC} / (T1PSC[15:0] + 1)$$

预分频支持分频自动更新，即在更新事件发生后，能够自动改变预分频值。当 T1CEN 为 0 时，写入预分频寄存器的值也能直接加载实际应用的预分频寄存器中。

1.3.1.2. 向上计数模式

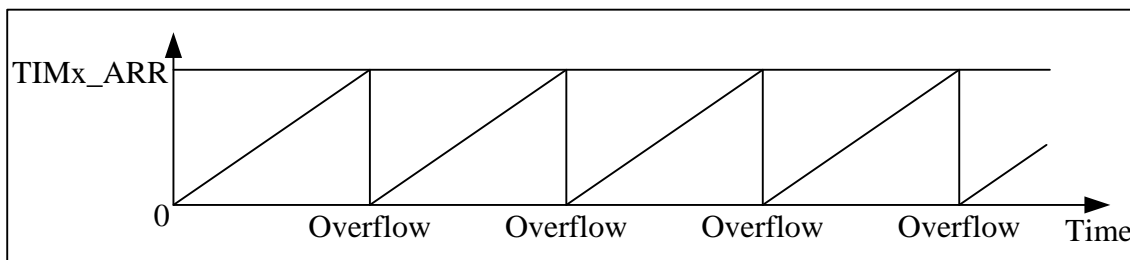


图 10.3 向上计数模式

在向上计数模式中，计数器从 0 开始计数向上计数，计到 TIM1_ARR 寄存器所设数值。然后重新从 0 开始计数并产生一个计数器上溢事件；如果 T1UDIS 设为 0，那么还会产生一个更新事件 UEV。

1.3.1.3. 向下计数模式

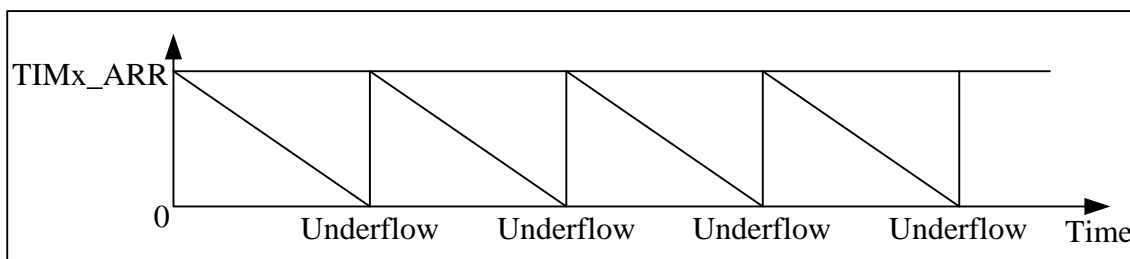


图 10.4 向下计数模式

在向下计数模式中，计数器从 TIM1_ARR 寄存器设置的自动重载值开始向下计数，直到计到 0。然后重新从自动重载值开始计数并产生一个计数器下溢事件；如果 T1UDIS 设为 0，那么还会产生一个更新事件 UEV。

1.3.1.4. 中心对齐模式

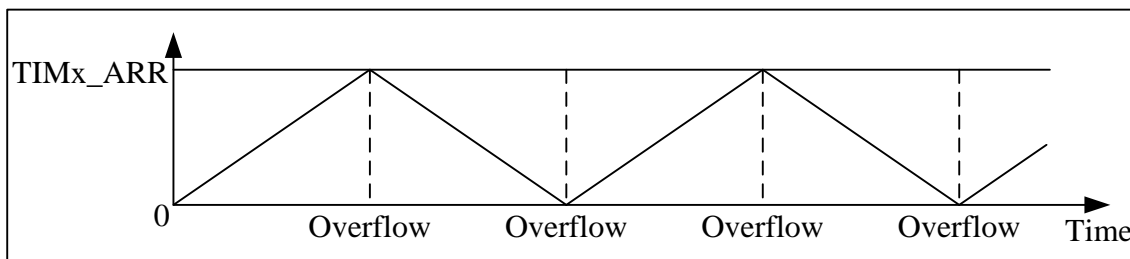


图 10.5 中心对齐模式

在中心对齐模式中，计数器从 0 开始向上计数，计到自动重载值。这时会产生一个计数器上溢事件。然后计数器开始向下计数计到 0，产生一个下溢事件。在这之后，计数器重新从 0 开始计数。

在这个模式下，方向位(T1DIR)不能进行写操作。方向位会由硬件设置成当前计数器的计数方向。

中心对齐模式所需注意事项：

- 当在中心对齐模式下开始计数时，当前的配置会被使用 – 计数开始值为写入 TIM1_CNTRH/L 中的值，计数开始方向决定于写入 TIM1_CR1 寄存器中的 T1DIR 位。注意 T1DIR 位和 T1CMS 值不能被软件同时改写。
- 运行在中心对齐模式下时，不建议写计数器值(TIM1_CNTRH/L)，因为可能会产生意想不到的结果。如果写入计数器的值大于自动加载值(TIM1_CNT > TIM1_ARR)，计数方向可能不会进行更新。如果写入计数器的值为 0 或为 TIM1_ARR，计数方向会进行更新但更新事件(UEV)不会产生。
- 中心对齐模式最安全的使用方式就是在开始计数前软件产生一个更新事件。将当前的配置加载到真正的控制信号当中进行使用。

1.3.1.5. 重复向下计数器

重复计数器是 8bit 的向下计数器，会在每次 timer 上溢或下溢时-1；只有当重复向下计数器减到 0 时，计数器上溢或下溢才会产生更新事件(UEV)；使用重复计数器能够设定更新事件的频率。

重复向下计数器自减事件：

- 计数器向上计数模式下的每个计数上溢事件
- 计数器向下计数模式下的每个计数下溢事件
- 计数器中心对齐模式下的每个计数上溢或下溢事件

重复向下计数器是自动重载的，当发生了软件更新事件或硬件更新事件(UEV)时，会将 TIM1_RCR 中的值自动重载到重复向下计数器中。

除外部时钟模式 1 外的其他模式：

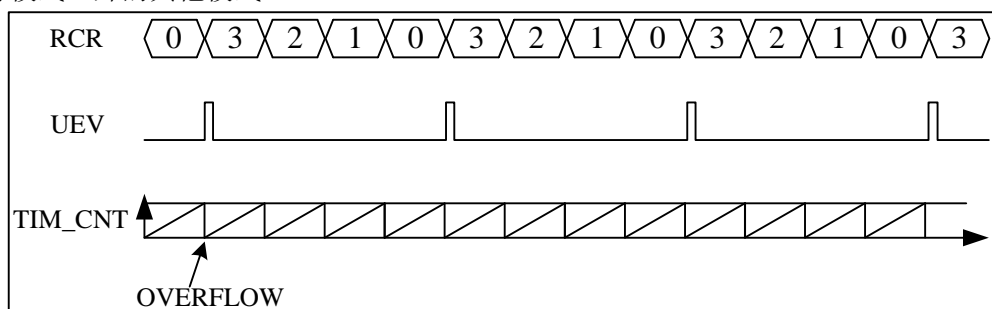


图 10.21 除外部时钟模式 1 外的其他情况 vs 重复计数器

外部时钟模式 1:

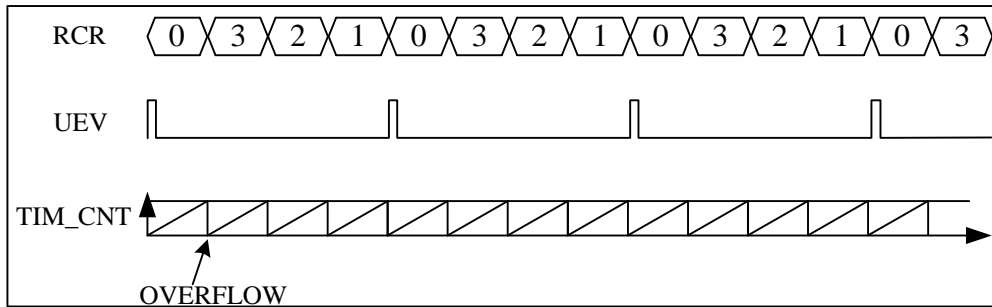


图 10.22 外部时钟模式 vs 重复计数器

在外部时钟模式 1 下, 当配置 T1RCR 不为 0 时, 在重复计数器减为 0 的那个 timer 时钟产生更新事件(UEV)。在除外部时钟模式 1 的其他模式下, 当配置 T1RCR 不为 0 时, 在重复计数器重新加载 T1RCR 寄存器值的那个 timer 时钟产生更新事件(UEV)。

在第一个硬件更新事件产生之前, 不管有没有软件设置产生更新事件(T1UG=1), 第一个更新事件的产生所需时间都与配置 RCR 数值不相符合。建议当配置 T1RCR 不为 0 时, 软件需要在第一个硬件更新事件之后再打开更新事件中断。

1.3.2. 计数控制器

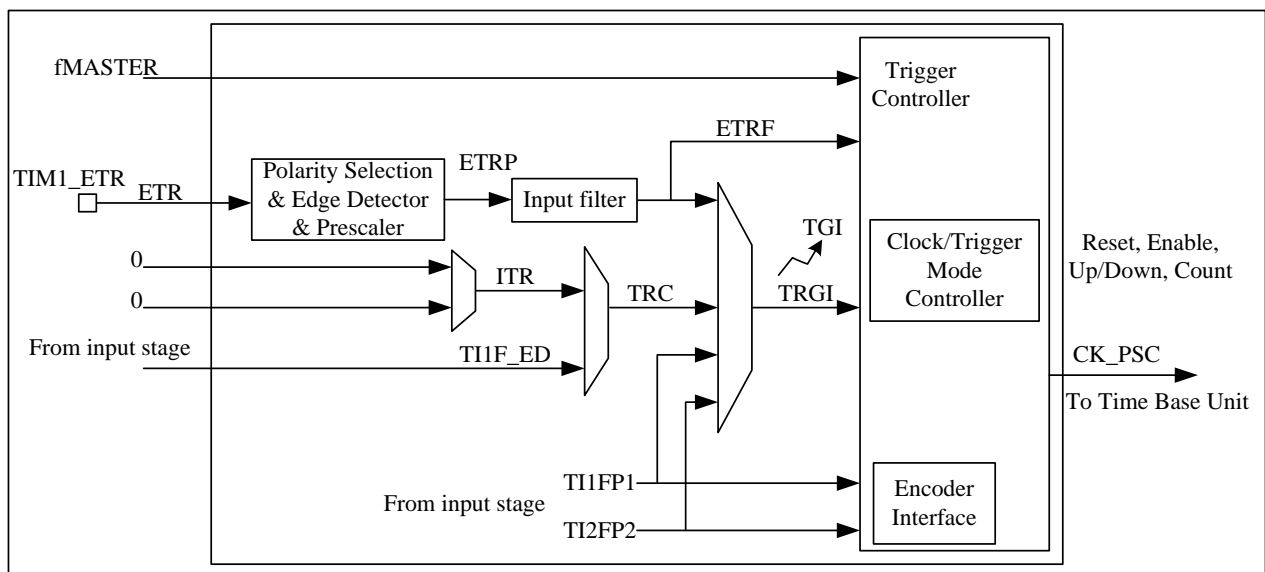


图 10.6 时钟/触发控制器框图

时钟/触发控制器允许配置各种计数器时钟源, 输入触发和输出触发。

1.3.2.1. 计数器时钟源

计数器一共有 3 个时钟源：

- 内部的 `clk_int` 分频时钟
- 外部时钟源 `etr`
- 触发源 `trgi`

其中 `etr` 是外部输入的异步信号，由于时钟同步的需求，因此 `etr` 的时钟周期必须大于内部的计数器时钟周期的 2 倍。

当计数器使用内部时钟时，由于时钟频率可能是 `cpu` 频率的 $n(n=1\sim128)$ 倍相等，则需要注意产生中断标志位的事件的时间长度，否则可能误产生 `T1CCxOF` 中断。

1.3.2.2. 计数触发源

在普通情况下（不是 `slave` 模式中的某个），使能 `T1CEN` 即可触发计数，但是在 `slave` 模式下，需要触发源，这些触发源包括：

- (1) 输入源 `T1` 的边沿检测，`TI1F_ED`；
- (2) 滤波后的通道 1 输入；
- (3) 滤波后的通道 2 输入；
- (4) 外部触发输入 `ETRF`；

其中外部触发输入 `ETRF` 可配置极性、预分频和沿检测，预分频为 $2n(n=0\sim3)$ 。

1.3.2.3. slave 模式控制

一共有 7 中 `slave` 模式，由寄存器 `T1SMS[2:0]` 控制：

- (1) 译码模式 1：根据 `TI1FP1` 的电平，在 `TI2FP2` 的边沿进行上计数或下计数；
- (2) 译码模式 2：根据 `TI2FP2` 的电平，在 `TI1FP1` 的边沿进行上计数或下计数；
- (3) 译码模式 3：根据其他输入电平，在 `TI1FP1` 和 `TI2FP2` 的边沿进行上计数或下计数；
- (4) 复位模式：复位模式下，计数值 `cnt` 会被配置的有效事件复位；
- (5) 门控模式：计数值 `cnt` 会被配置的有效事件暂停；
- (6) 触发模式：只有当配置的有效事件发生时，`cnt` 才开始计数；
- (7) 外部时钟模式 1：选择的触发输入(`TRGI`)的上升沿驱动计数器

各个模式更详细的描述参照 `TIMx_SMCR` 寄存器的描述。

设计细节：

- 1.在配置复位模式和触发模式时，可以通过配置 `T1CC1NP/T1CC1P` 来配置选择触发的边沿，具体参照 `T1CC1NP/T1CC1P` 的描述。
- 2.译码模式下，其输入的信号可以选择极性，通过配置 `T1CCxP` 选择。
- 3.门控模式只能选择部分触发源，因为一部分触发源是脉冲的。
- 4.外部时钟模式 1 比外部时钟模式 2 的优先级更低。

译码模式：

有效计数沿	相对信号的电平值	TI1FP1 信号	TI2FP2 信号
-------	----------	-----------	-----------

	(TI1FP1 for TI2) (TI2FP2 for TI1)	上升	下降	上升	下降
TI1	高	向下	向上	——	——
	低	向上	向下	——	——
TI2	高	——	——	向上	向下
	低	——	——	向下	向上
TI1 or TI2	高	向下	向上	向上	向下
	低	向上	向下	向下	向上

表 10.1 计数方向 vs 译码信号

复位模式：

当触发输入事件到来时，计数器和计数器预分频都会被初始化。如果此时 T1URS 为 0，则会产生一个更新事件，同时所有的预加载寄存器都会被更新。

复位模式的步骤示例：

1. 配置输入捕捉寄存器的值 – 配置输入捕捉滤波器 T1IC1F=000；配置捕捉预分频器 T1IC1PSC=0
2. 将通道配置为输入捕捉通道 T1CC1S=01，并将 IC1 映射在 TI1FP1 上
3. 写 T1CC1P=0，选择检测触发上升沿的到来
4. 通过写 T1SMS=100，将 timer 配置为复位模式。同时写 T1TS=101，选择 TI1 为输入触发源
5. 置位 T1CEN，启动计数器

当 TI1 的上升沿到来时，计数器被清 0 并从 0 开始重新计数。与此同时与此同时，触发标志位(TIF)会被置位，在触发中断使能的情况下还会产生一个中断请求。

门控模式：

依据选择的触发输入的电平值，计数器会被使能。此模式下，计数器的运行和停止都是受控的。

门控模式的步骤示例：

1. 配置输入捕捉寄存器的值 – 配置输入捕捉滤波器 T1IC1F=000；配置捕捉预分频器 T1IC1PSC=0
2. 将通道配置为输入捕捉通道 T1CC1S=01，并将 IC1 映射在 TI1FP1 上
3. 写 T1CC1P=1，选择检测输入低电平的到来
4. 通过写 T1SMS=101，将 timer 配置为门控模式。同时写 T1TS=101，选择 TI1 为输入源
5. 置位 T1CEN，使能计数器(在门控模式下，需要开启 T1CEN；在此基础上，才能由输入源控制计数器的运行与停止)

当 TI1 为低电平时，计数器在内部时钟的驱动下进行计数；当 TI1 变为高电平时，计数器停止计数。触发标志位(T1TIF)会在计数器启动或停止时被置位

触发模式：

依据选择的触发输入的电平值，计数器会被启动(T1CEN 被置位)。

触发模式的步骤示例：

1. 配置输入捕捉寄存器的值 – 配置输入捕捉滤波器 T1IC1F=000；配置捕捉预分频器 T1IC1PSC=0
2. 将通道配置为输入捕捉通道 T1CC1S=01，并将 IC1 映射在 TI1FP1 上
3. 写 T1CC1P=0，选择检测触发上升沿的到来
4. 通过写 T1SMS=110，将 timer 配置为触发模式。同时写 T1TS=101，选择 TI1 为输入触发源

当 TI1 的上升沿到来时，计数器在内部时钟的驱动下启动计数，并且触发标志位(T1TIF)被置位。

外部时钟模式 1:

当 T1SMS=111 时，此模式被选中。计数器会在每个输入触发的上升沿或下降沿进行计数。

外部时钟模式 1 的步骤示例:

1. 配置输入捕捉寄存器的值 – 配置输入捕捉滤波器 T1IC1F=000; 配置捕捉预分频器 T1IC1PSC=0
2. 将通道配置为输入捕捉通道 T1CC1S=01, 并将 IC1 映射在 TI1FP1 上
3. 写 T1CC1P=0, 选择检测触发上升沿的到来
4. 通过写 T1SMS=111, 将 timer 配置为外部时钟模式 1; 同时写 T1TS=110, 选择 TI1 为输入源
5. 置位 T1CEN, 使能计数器

当 TI1 上升沿到来时，计数器进行一次计数，并且触发标志位(T1TIF)被置位；如果触发中断使能(T1TIE)，则会产生一个中断请求。

外部时钟模式 2:

当 T1ECE=1 时，此模式被选中。计数器会在每个输入触发的上升沿或下降沿进行计数。

外部时钟模式 2 的步骤示例:

1. 配置外部触发寄存器的值 – 配置外部触发滤波器 T1ETF=000; 配置外部触发预分频器 T1ETPS=0
2. 写 T1ETP=0, 选择上升沿为检测沿
3. 通过写 T1ECE=1, 将 timer 配置为外部时钟模式 2
4. 置位 T1CEN, 使能计数器

当 TI1 上升沿到来时，计数器进行一次计数，并且触发标志位(T1TIF)被置位；如果触发中断使能(T1TIE)，则会产生一个中断请求。

1.3.3. 捕捉比较通道

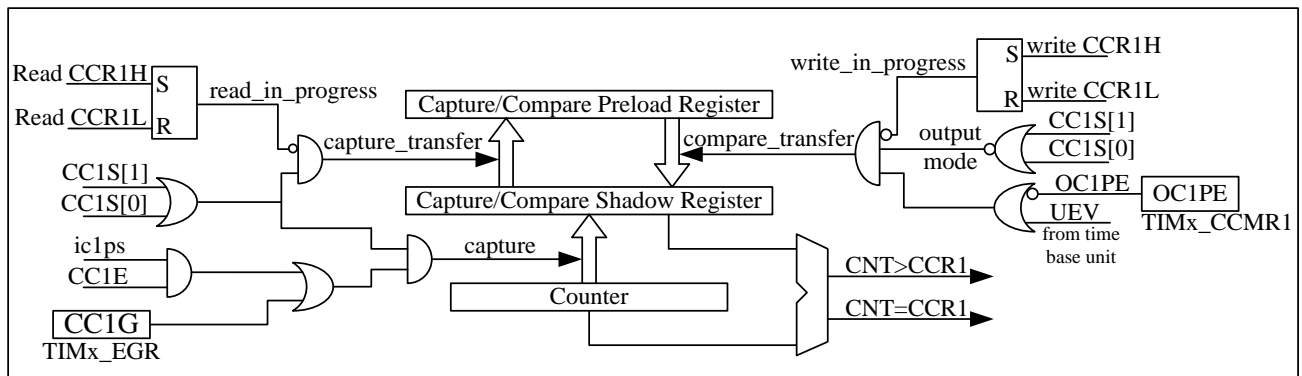


图 10.7 捕捉/比较通道 1 框图

timer 的 I/O 口能被配置为输入捕捉或输出比较功能。这个配置由 CCiS 通道选择位进行设定。

1.3.3.1. 捕捉输入通道

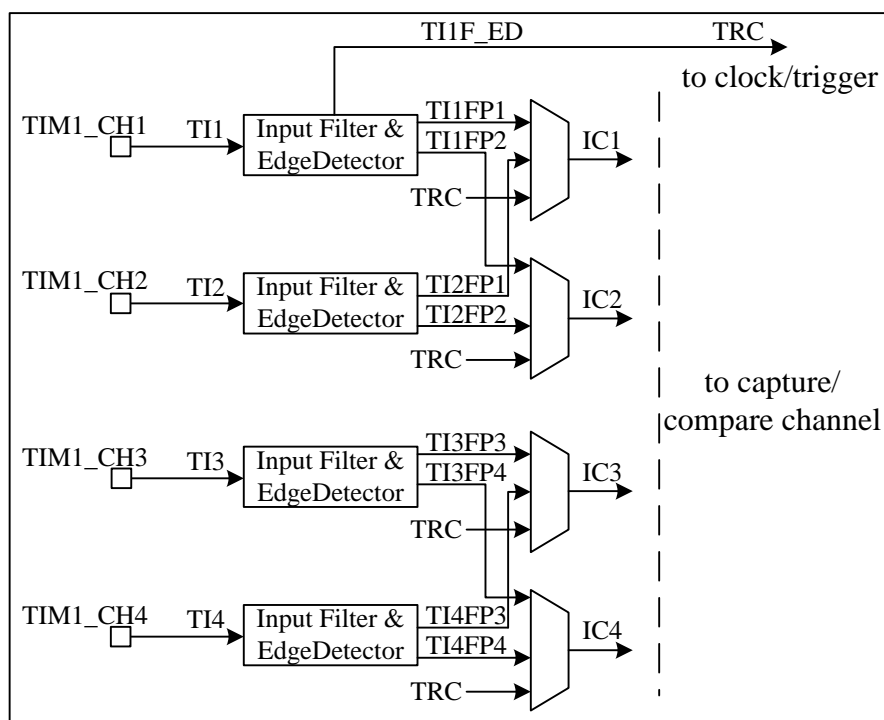


图 10.8 通道输入框图

当一个通道被配置成捕捉通道的时候，当发生的配置好的输入事件时，可以将当前的计数值保存在 CCRx 寄存器。

当一个输入捕捉发生时：

- TIM1_CCR1 寄存器得到捕捉发生时计数器的值
- 输入捕捉标志位(T1CCxIF)被置位。如果当 T1CCxIF 保持为 1 时，有一次发生了输入捕捉事件，那么溢出捕捉标志位(T1CCxOF)也会被置位
- 如果 T1CCxIE 为 1，那么捕捉将产生一个中断事件

每个通道都由一个数字滤波单元，配置 T1ICXF[3:0]可以配置滤波的各个参数，具体参考相应寄存器位的描述。

有效事件源：

- (1) TI1
- (2) TI2
- (3) TRC

捕捉触发，可以配置连续采样到 $n(n=0,1,2,4,8)$ 次有效事件后进行捕捉操作。

1.3.3.2. 输出比较通道

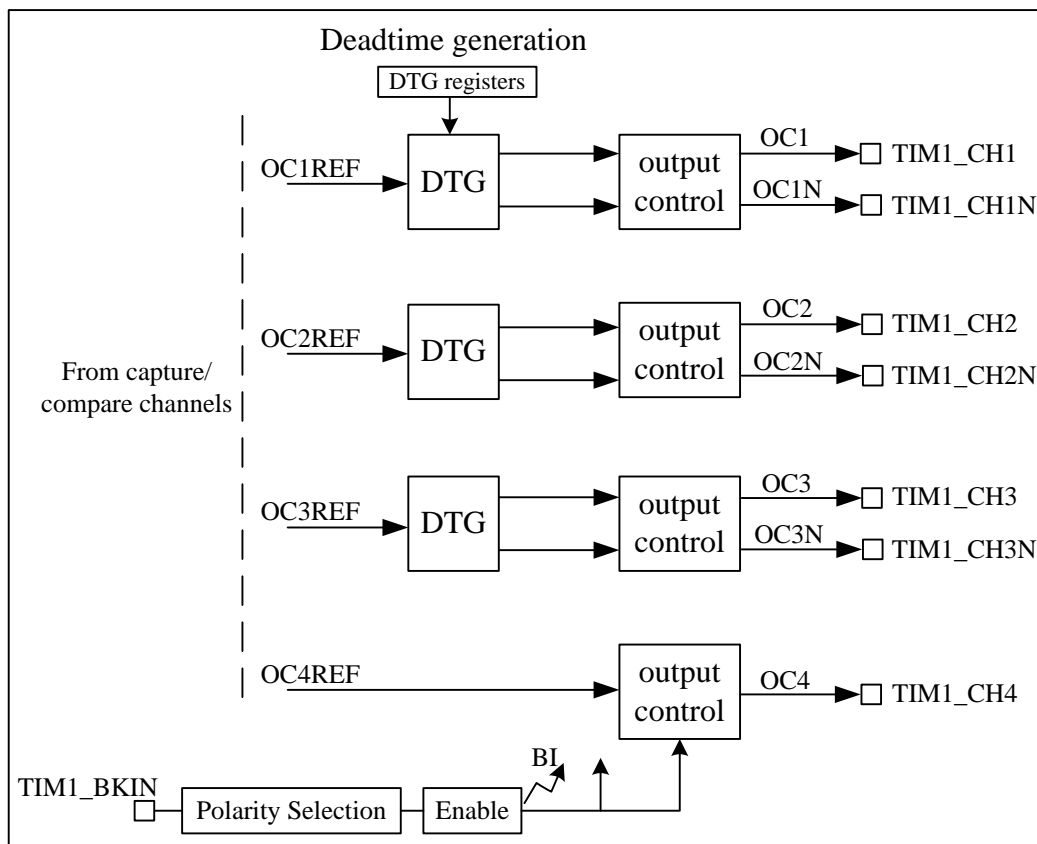


图 10.9 通道输出框图

输出阶段产生立即响应的波形，用来作为参考波形，叫做 OCxREF 信号(高有效)。刹车功能和极性选择都在参考波形之后去做控制。

输出比较通道根据计数值与比较值 CCRx，产生 ocxref 输出并送到死区产生模块。一共有 8 种比较模式，每个模式的输出 ocxref:

- (1) 冻结模式：输出保持无效；
- (2) 匹配有效：当计数值 cnt 与 ccr 匹配时，输出有效；
- (3) 匹配无效：当计数值 cnt 与 ccr 匹配时，输出无效；
- (4) 翻转：当计数值 cnt 与 ccr 匹配时，输出翻转；
- (5) 强制无效：输出强制低电平。
- (6) 强制有效：输出强制高电平。
- (7) PWM1：向上计数时，当 $CNT < CCR$ 时，输出有效；向下计数时， $CNT > CCR$ 时，输出无效。
- (8) PWM2：向上计数时，当 $CNT < CCR$ 时，输出无效；向下计数时， $CNT > CCR$ 时，输出有效。

输出配置所需注意事项：

- 刹车事件的到来只会影响输出，并不影响 timer 本身的计数。
- 建议在计数器运行过程中重新配置 T1CCxE 和 T1CCxNE 寄存器需要开启捕捉/比较预加载控制位 (CCPC)
- 为了避免在计数器还未开启时，就开始输出脉冲；需要在计数器使能之后在开启 T1CCxE 和 T1CCxNE

1.3.3.3. 死区产生

当把通道的互补输出使能时，就自动使能死区功能。每当一个输出信号出现下降沿时，就会将另一个信号的上升沿后延一个死区时间长度。

死区时间可以编程:根据寄存器位 T1DTG[7:0]，可以配置死区时间长度，具体参考寄存器描述 TIM1_BDTR 的 T1DTG[7:0]。

有的 ocxref 输出的脉冲时间很短（小于死区时间），有可能某一互补信号的脉冲会被死区覆盖，导致输出不变化。

当一个刹车事件发生时：

- T1MOE 位会被异步清 0，强制输出进入无效状态，空闲状态或复位状态。甚至在 MCU 振荡器关闭的情况下，T1MOE 也会被刹车事件清 0。
- 在 T1MOE=0 之后，每个输出通道都会先将输出值置为无效值，等死区时间到之后变成提前设置好的 T1OISi 位的值。如果 T1OSSI=0，timer 会将输出关闭。
- 当互补输出使能时：
输出首先会设置为无效值(根据极性选择位)。此操作是异步清 0 的，所以即使 timer 没有时钟驱动也能进行。
如果 timer 是有时钟进行驱动的，那么死区时间到来之后就会进入由 T1OISi 和 T1OISiN 提前设定的预设状态。(由于 T1MOE 的同步，所以此情况下真正的死区时间会比死区设置值长 2 个 timer 时钟)
- 刹车状态标志位(T1BIF)被置位。如果 T1BIE 位为 1，那么将会产生一个中断事件。
- 如果 T1AOE 位配置为 1，那么 T1MOE 位在下次更新事件(UEV)到来时，将会由硬件自动置位。如果 T1AOE 位为 0，那么只能由软件将 T1MOE 位重新置位。

1.3.3.4. 输出控制

控制位					输出状态	
T1MOE	T1OSSI	T1OSSR	T1CCxE	T1CCxNE	OCx 输出状态	OCxN 输出状态
1	X	0	0	0	输出关闭(不由 timer 驱动) OCx=0, OCx_EN=0	输出关闭(不由 timer 驱动) OCxN=0, OCxN_EN=0
		0	0	1	输出关闭(不由 timer 驱动) OCx=0, OCx_EN=0	OCxREF + 极性选择 OCxN=OCxREF ^ T1CCxNP OCxN_EN=1
		0	1	0	OCxREF + 极性选择 OCx=OCxREF ^ T1CCxNP OCx_EN=1	输出关闭(不由 timer 驱动) OCxN=0, OCxN_EN=0
		0	1	1	OCxREF + 极性选择 + 死区时间 OCx_EN=1	OCREF 的互补信号 + 极性选择 + 死区时间 OCxN_EN=1
		1	0	0	输出关闭(不由 timer 驱动) OCx=T1CCxP, OCx_EN=0	输出关闭(不由 timer 驱动) OCxN=T1CCxNP, OCxN_EN=0

		1	0	1	关闭状态(运行模式下输出使能) OCx=T1CCxP, OCx_EN=1	OCxREF + 极性选择 OCxN=OCxREF ^ T1CCxNP OCxN_EN=1
		1	1	0	OCxREF + 极性选择 OCx=OCxREF ^ T1CCxNP OCx_EN=1	关闭状态(运行模式下输出使能) OCxN=T1CCxNP, OCx_EN=1
		1	1	1	OCxREF + 极性选择 + 死区时间 OCx_EN=1	OCREF 的互补信号 + 极性选择 + 死区时间 OCxN_EN=1
0	0	X	0	0	输出关闭(不由 timer 驱动) OCx=T1CCxP, OCx_EN=0	
	0		0	1	输出关闭(不由 timer 驱动)	
	0		1	0	一开始	
	0		1	1	OCx=T1CCxP,OCx_EN=0,OCxN=T1CCxNP,OCxN_EN=0 在死区时间之后 OCx=T1OISx,OCxN=T1OISxN	
	1		0	0	输出关闭(不由 timer 驱动) OCx=T1CCxP, OCx_EN=0	输出关闭(不由 timer 驱动) OCxN=T1CCxNP,OCxN_EN=0
	1		0	1	关闭状态(空闲模式下输出使能)	
	1		1	0	一开始	
	1		1	1	OCx=T1CCxP,OCx_EN=1,OCxN=T1CCxNP,OCxN_EN=1 在死区时间之后 OCx=T1OISx,OCxN=T1OISxN	

表 10.2 输出控制与输出状态

可以根据 TIMOE、TIOSSI、TIOSSR、T1CCxNE 和 CcxE 进行输出的控制。

设计：输出的状态转换实际是 break 事件异步将 TIMOE 清零实现的，TIMOE 异步清零，但要注意释放时需要同步 2 个 Timer1 时钟（如果有时钟的情况下）。

1.3.4. TIM1 中断

Timer1 有以下 8 个中断请求源：

- 刹车中断
- 触发中断
- 换相中断
- 捕捉/比较 4 中断
- 捕捉/比较 3 中断
- 捕捉/比较 2 中断
- 捕捉/比较 1 中断
- 更新中断(例如：上溢、下溢、计数初始化)

在用这些中断之前需要提前打开 TIM1_IER 寄存器中的中断使能位(T1BIE、T1TIE、T1COMIE、T1CCiIE 和 T1UIE)。

不同的中断源还可以配置通过 TIM1_EGR 寄存器来产生(软件产生中断)。

1.3.5. 故障刹车源

Timer1 有以下 3 种刹车事件：

- BKIN 管脚事件
- LVD 事件
- ADC 比较事件

当故障事件有效且被选择为刹车源（由 BKS0~2 决定），如果 BKE 位为 1，PWM 输出管脚将被置于预设的状态，预设状态由寄存器 TIM1OISR 决定。

刹车功能有以下特性：

当故障事件有效时，T1MOE 清 0，PWM 输出将一直置于预设状态；

故障事件撤消后，如果 T1AOE=1，PWM 将在下一 PWM 周期恢复正常输出，否则，软件需要自动打开 T1MOE；

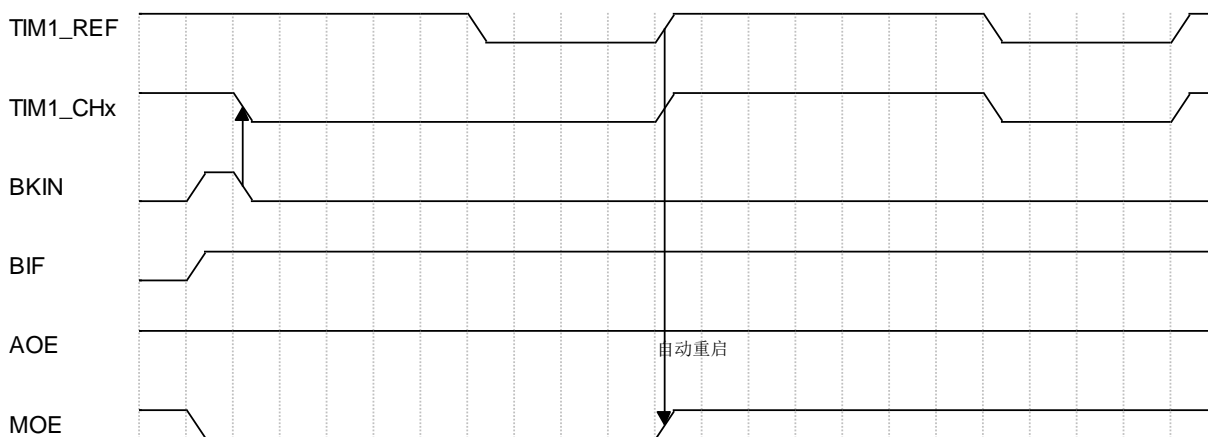


图 10.10 PWM 的自动重启

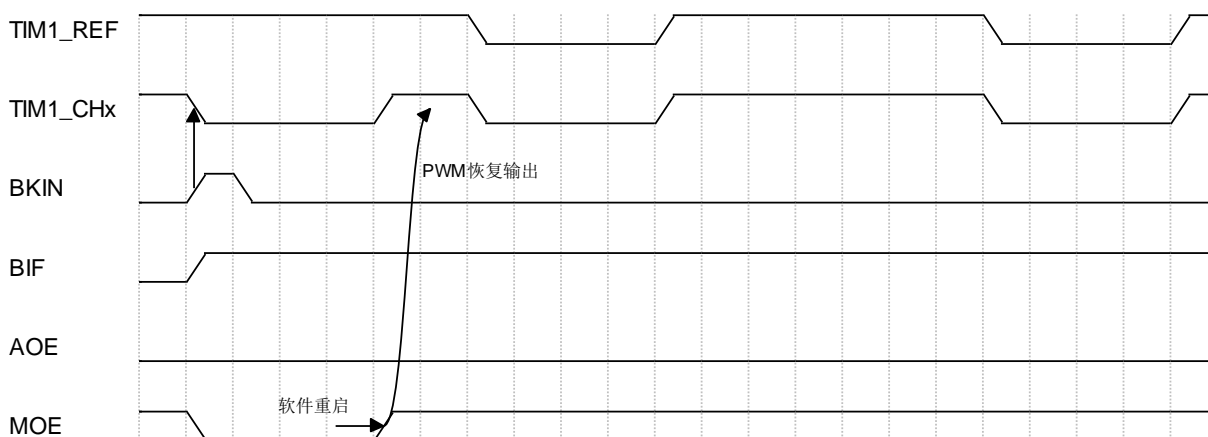


图 10.11 PWM 的软件重启

1.3.6. 前沿消隐

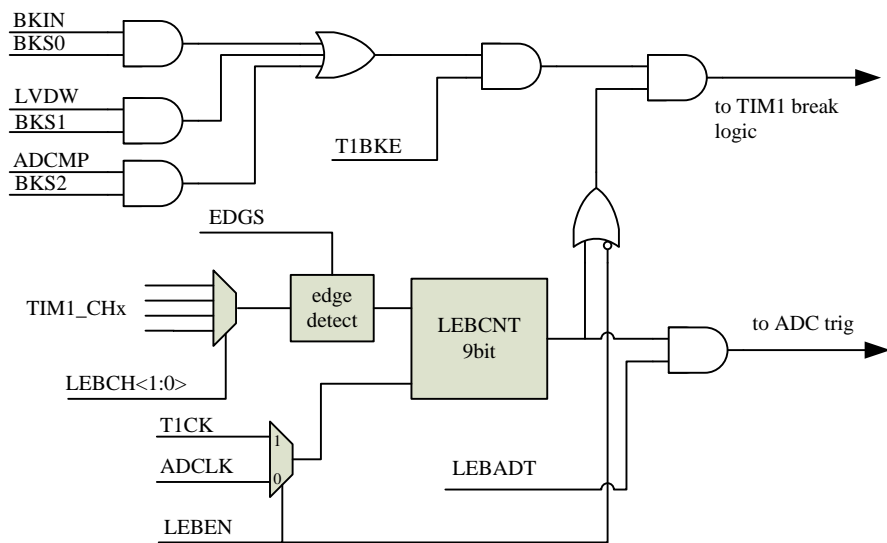


图 10.12 前沿消隐原理框图

在高速开关应用中，开关通常会产生极大的瞬变，这些瞬变可能会导致测量误差。利用前沿消隐（LEB）功能，应用程序可以忽略 PWM 输出边沿附近发生的瞬变。

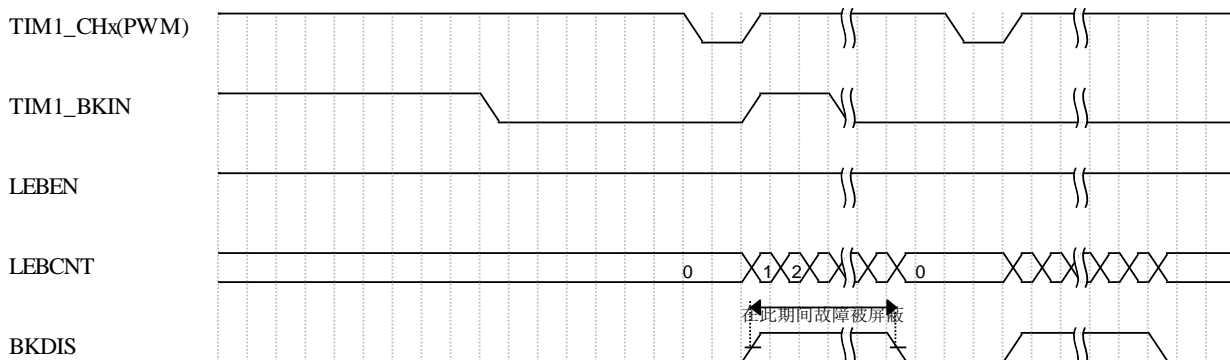


图 10.13 前沿消隐时序示意图

LEBCH 用于选择被消隐 Timer1 的 PWM 通道，EDGS 选择边沿类型。当 LEBEN 为 1，PWM 边沿将触发 LEB 定时器计数，时钟源为 Timer1 时钟，直到计数值等于 LEBPR，LEB 定时器停止计数，这段时间为消隐周期，期间所发生的刹车事件将被忽略；在消隐周期内如果再次发生有效的 PWM 边沿，则 LEB 定时器将清 0，重新开始计数。

注意：

LEB 定时器和 ADC 延时定时器复用了同一个 9bit 计数器，当 LEBEN 为 1 时，原 ADC 的延时触发功能被禁止，但如果 LEBADT 为 1，LEB 定时器溢出将触发一次 AD 转换。

1.4. 与 TIMER1 相关寄存器汇总

名称	地址	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	复位值
TIM1CR1	0x211	T1ARPE	T1CMS[1:0]		T1DIR	T1OPM	T1URS	T1UDIS	T1CEN	0000 0000
TIM1SMCR	0x213	—	T1TS[2:0]			—	T1SMS[2:0]			-000 -000
TIM1IER	0x215	T1BIE	T1TIE	—	T1CC4IE	T1CC3IE	T1CC2IE	T1CC1IE	T1UIE	00-0 0000
TIM1SR1	0x216	T1BIF	T1TIF	—	T1CC4IF	T1CC3IF	T1CC2IF	T1CC1IF	T1UIF	00-0 0000
TIM1SR2	0x217	—	—	—	T1CC4OF	T1CC3OF	T1CC2OF	T1CC1OF	—	---0 000-
TIM1EGR	0x218	T1BG	—	—	T1CC4G	T1CC3G	T1CC2G	T1CC1G	—	0--0 000-
TIM1CCMR1 (output mode)	0x219	—	T1OC1M[2:0]			T1OC1PE	—	T1CC1S[1:0]		-000 0-00
TIM1CCMR1 (input mode)		T1IC1F[3:0]			T1IC1PSC[1:0]		T1CC1S[1:0]		0000 0000	
TIM1 CCMR2 (output mode)	0x21A	—	T1OC2M[2:0]			T1OC2PE	—	T1CC2S[1:0]		-000 0-00
TIM1CCMR2 (input mode)		T1IC2F[3:0]			T1IC2PSC[1:0]		T1CC2S[1:0]		0000 0000	
TIM1CCMR3 (output mode)	0x21B	—	T1OC3M[2:0]			T1OC3PE	—	T1CC3S[1:0]		-000 0-00
TIM1CCMR3 (input mode)		T1IC3F[3:0]			T1IC3PSC[1:0]		T1CC3S[1:0]		0000 0000	
TIM1CCMR4 (output mode)	0x21C	—	T1OC4M[2:0]			T1OC4PE	—	T1CC4S[1:0]		-000 0-00
TIM1CCMR4 (input mode)		T1IC4F[3:0]			T1IC4PSC[1:0]		T1CC4S[1:0]		0000 0000	
TIM1CCER1	0x21D	T1CC2NP	T1CC2NE	T1CC2P	T1CC2E	T1CC1NP	T1CC1NE	T1CC1P	T1CC1E	0000 0000
TIM1CCER2	0x21E	—	—	T1CC4P	T1CC4E	T1CC3NP	T1CC3NE	T1CC3P	T1CC3E	--00 0000
TIM1CNTRH	0x28C	T1CNT[15:8]								0000 0000
TIM1CNTRL	0x28D	T1CNT[7:0]								0000 0000
TIM1PSCRH	0x28E	T1PSC[15:8]								0000 0000
TIM1PSCRL	0x28F	T1PSC[7:0]								0000 0000
TIM1ARRH	0x290	T1ARR[15:8]								1111 1111
TIM1ARRL	0x291	T1ARR[7:0]								1111 1111
TIM1RCR	0x292	T1REP[7:0]								0000 0000
TIM1CCR1H	0x293	T1CCR1[15:8]								0000 0000
TIM1CCR1L	0x294	T1CCR1[7:0]								0000 0000
TIM1CCR2H	0x295	T1CCR2[15:8]								0000 0000
TIM1CCR2L	0x296	T1CCR2[7:0]								0000 0000
TIM1CCR3H	0x297	T1CCR3[15:8]								0000 0000
TIM1CCR3L	0x298	T1CCR3[7:0]								0000 0000
TIM1CCR4H	0x299	T1CCR4[15:8]								0000 0000
TIM1CCR4L	0x29A	T1CCR4[7:0]								0000 0000
TIM1BKR	0x29B	T1MOE	T1AOE	T1BKP	T1BKE	T1OSSR	T1OSSI	T1LOCK[1:0]		0000 0000
TIM1DTR	0x29C	T1DTG[7:0]								0000 0000
TIM1OISR	0x29D	—	T1OIS4	T1OIS3N	T1OIS3	T1OIS2N	T1OIS2	T1OIS1N	T1OIS1	-000 0000
LEBCON	0x41C	LEBEN	LEBCH[1:0]		—	EDGS	BKS[2:0]			000- 0000

1.4.1. TIM1CR1, 地址: 0x211

Bit	7	6	5	4	3	2	1	0
Name	T1ARPE	T1CMS[1:0]		T1DIR	T1OPM	T1URS	T1UDIS	T1CEN
Reset	0	0	0	0	0	0	0	0
Type	RW	RW	RW	RW	RW	RW	RW	RW
7	T1ARPE: 自动预装载允许位 0: TIM1_ARR 寄存器没有缓冲, 它可以被直接写入; 1: TIM1_ARR 寄存器由预装载缓冲器缓冲。							
6:5	T1CMS[1:0]: 选择中央对齐模式 00: 边沿对齐模式。计数器依据方向位(DIR)向上或向下计数。 01: 中央对齐模式1。计数器交替地向上和向下计数。配置为输出的通道(TIM1_CCMRx寄存器中CciS=00)的输出比较中断标志位, 只在计数器向下计数时被置1。 10: 中央对齐模式2。计数器交替地向上和向下计数。配置为输出的通道(TIM1_CCMRx寄存器中CciS=00)的输出比较中断标志位, 只在计数器向上计数时被置1。 11: 中央对齐模式3。计数器交替地向上和向下计数。配置为输出的通道(TIM1_CCMRx寄存器中CciS=00)的输出比较中断标志位, 在计数器向上和向下计数时均被置1。 注1: 在计数器开启时(CEN=1), 不允许从边沿对齐模式转换到中央对齐模式。 注2: 在中央对齐模式下, 编码器模式 (GPT_SMCR寄存器中的SMS=001, 010, 011) 必须被禁止。							
4	T1DIR: 方向 0: 计数器向上计数; 1: 计数器向下计数。 注: 当计数器配置为中央对齐模式或编码器模式时, 该位为只读。							
3	T1OPM: 单脉冲模式 0: 在发生更新事件时, 计数器不停止; 1: 在发生下一次更新事件(清除CEN位)时, 计数器停止。							
2	T1URS: 更新请求源 0: 如果UDIS允许产生更新事件, 则下述任一事件产生一个更新中断: 寄存器被更新(计数器上溢/下溢) 软件设置UG位 时钟/触发控制器产生的更新 1: 如果UDIS允许产生更新事件, 则只有当下列事件发生时才产生更新中断, 并UIF置1: 寄存器被更新(计数器上溢/下溢)							
1	T1UDIS: 禁止更新 0: 一旦下列事件发生, 产生更新(UEV)事件: 计数器溢出/下溢 产生软件更新事件 时钟/触发模式控制器产生的硬件复位被缓存的寄存器被装入它们的预装载值。 1: 不产生更新事件, 影子寄存器(ARR、PSC、CCRx)保持它们的值。如果设置了UG位或时钟/触发控制器发出了一个硬件复位, 则计数器和预分频器被重新初始化。							

0	TICEN: 允许计数器 0: 禁止计数器; 1: 使能计数器。 注: 在软件设置了CEN位后, 外部时钟、门控模式和编码器模式才能工作。然而触发模式可以自动地通过硬件设置CEN位。
---	---

1.4.2. TIM1SMCR, 地址: 0x213

Bit	7	6	5	4	3	2	1	0
Name	T1MSM	T1TS[2:0]			reserved	T1SMS[2:0]		
Reset	0	0	0	0	—	0	0	0
Type	RW	RW	RW	RW	RO-0	RW	RW	RW
7	T1MSM: 主/从模式 0: 无作用; 1: 触发输入(TRGI)上的事件被延迟了, 以允许定时器1与它的从定时器间的完美同步(通过TRGO)。 此功能在此设计中没有作用, 因为此设计没有级联的时钟							
6:4	T1TS[2:0]: 触发选择 这3位选择用于选择同步计数器的触发输入。 000: 内部触发ITR0连接到TIM6 TRGO (此设计没有TIM6, 所以固定接0) 001: 保留 010: 内部触发ITR2连接到TIM5 TRGO(此设计没有TIM5, 所以固定接0) 011: 保留 100: TI1的边沿检测器(TI1F_ED) 101: 滤波后的定时器输入1(TI1FP1) 110: 滤波后的定时器输入2(TI2FP2) 111: 外部触发输入(ETRF) 注: 这些位只能在未用到(如SMS=000)时被改变, 以避免在改变时产生错误的边沿检测。							
2:0	T1SMS: 时钟/触发/从模式选择 当选择了外部信号, 触发信号(TRGI)的有效边沿与选中的外部输入极性相关(见输入控制寄存器和控制寄存器的说明) 000: 时钟/触发控制器禁止 – 如果CEN=1, 则预分频器直接由内部时钟驱动。 001: 编码器模式1 – 根据TI1FP1的电平, 计数器在TI2FP2的边沿向上/下计数。 010: 编码器模式2 – 根据TI2FP2的电平, 计数器在TI1FP1的边沿向上/下计数。 011: 编码器模式3 – 根据另一个输入的电平, 计数器在TI1FP1和TI2FP2的边沿向上/下计数。 100: 复位模式 – 在选中的触发输入(TRGI)的上升沿时重新初始化计数器, 并且产生一个更新寄存器的信号。 101: 门控模式 – 当触发输入(TRGI)为高时, 计数器的时钟开启。一旦触发输入变为低, 则计数器停止(但不复位)。计数器的启动和停止都是受控的。 110: 触发模式 – 计数器在触发输入TRGI的上升沿启动(但不复位), 只有计数器的启动是受控的。 111: 外部时钟模式1 – 选中的触发输入(TRGI)的上升沿驱动计数器。 注: 如果TI1F_ED被选为触发输入(TS=100)时, 不要使用门控模式。这是因为TI1F_ED在每次TI1F变化时只是输出一个脉冲, 然而门控模式是要检查触发输入的电平。							

1.4.3. TIM1IER, 地址: 0x215

Bit	7	6	5	4	3	2	1	0
Name	T1BIE	T1TIE	T1COMIE	T1CC4IE	T1CC3IE	T1CC2IE	T1CC1IE	T1UIE
Reset	0	0	0	0	0	0	0	0
Type	RW	RW	RW	RW	RW	RW	RW	RW
7	T1BIE: 允许刹车中断 0: 禁止刹车中断; 1: 允许刹车中断。							
6	T1TIE: 触发中断使能 0: 禁止触发中断; 1: 使能触发中断。							
5	T1COMIE: 允许COM中断 0: 禁止COM中断; 1: 允许COM中断。							
4	T1CC4IE: 允许捕获/比较4中断 0: 禁止捕获/比较4中断; 1: 允许捕获/比较4中断。							
3	T1CC3IE: 允许捕获/比较3中断 0: 禁止捕获/比较3中断; 1: 允许捕获/比较3中断。							
2	T1CC2IE: 允许捕获/比较2中断 0: 禁止捕获/比较2中断; 1: 允许捕获/比较2中断。							
1	T1CC1IE: 允许捕获/比较1中断 0: 禁止捕获/比较1中断; 1: 允许捕获/比较1中断。							
0	T1UIE: 允许更新中断 0: 禁止更新中断; 1: 允许更新中断。							

1.4.4. TIM1SR1, 地址: 0x216

Bit	7	6	5	4	3	2	1	0
Type	T1BIF	T1TIF	T1COMIF	T1CC4IF	T1CC3IF	T1CC2IF	T1CC1IF	T1UIF
Reset	0	0	0	0	0	0	0	0
Type	R-W0	R-W0	R-W0	R-W0	R-W0	R-W0	R-W0	R-W0
7	T1BIF: 刹车中断标记 一旦刹车输入有效, 由硬件对该位置1。如果刹车输入无效, 则该位可由软件清0。 0: 无刹车事件产生; 1: 刹车输入上检测到有效电平。							
6	T1TIF: 触发器中断标记 当发生触发事件(当从模式控制器处于除门控模式外的其它模式时, 在TRGI输入端检测到有效边沿, 或门控模式下的任一边							

	沿)时由硬件对该位置1。它由软件清0。 0: 无触发器事件产生; 1: 触发中断等待响应。
5	T1COMIF : COM中断标记 一旦产生COM事件(当捕获/比较控制位: CciE、CciNE、OciM已被更新)该位由硬件置1。它由软件清0。 0: 无COM事件产生; 1: COM中断等待响应。
4	T1CC4IF : 捕获/比较4中断标记 参考CC1IF描述。
3	T1CC3IF : 捕获/比较3中断标记 参考CC1IF描述。
2	T1CC2IF : 捕获/比较2中断标记 参考CC1IF描述。
1	T1CC1IF : 捕获/比较1中断标记 如果通道CC1配置为输出模式: 当计数器值与比较值匹配时该位由硬件置1, 但在中心对称模式下除外(参考TIM1_CR1寄存器的CMS位)。它由软件清0。 0: 无匹配发生; 1: TIMx_CNT的值与TIMx_CCR1的值匹配。 注: 在中心对称模式下, 当计数器值为0时, 向上计数, 当计数器值为ARR时, 向下计数(它从0向上计数到ARR-1, 再由ARR向下计数到1)。因此, 对所有的SMS位值, 这两个值都不置标记。但是, 如果CCR1>ARR, 则当CNT达到ARR值时, CC1IF置1。 如果通道CC1配置为输入模式: 当捕获事件发生时该位由硬件置1, 它由软件清0或通过读TIM1_CCR1L清0。 0: 无输入捕获产生; 1: 计数器值已被捕获(拷贝)至TIM1_CCR1(在IC1上检测到与所选极性相同的边沿)。
0	T1UIF : 更新中断标记 当产生更新事件时该位由硬件置1。它由软件清0。 0: 无更新事件产生; 1: 更新事件等待响应。当寄存器被更新时该位由硬件置1: 若TIM1_CR1寄存器的UDIS=0, 当计数器上溢或下溢时; 若TIM1_CR1寄存器的UDIS=0、URS=0, 当设置TIM1_EGR寄存器的UG位软件对计数器CNT重新初始化时; 若TIM1_CR1寄存器的UDIS=0、URS=0, 当计数器CNT被触发事件重新初始化时 (参考0从模式控制寄存器TIM1_SMCR)。

1.4.5. TIM1SR2, 地址: 0x217

Bit	7	6	5	4	3	2	1	0
Name	reserved			T1CC4OF	T1CC3OF	T1CC2OF	T1CC1OF	reserved
Reset	—	—	—	0	0	0	0	—
Type	RO-0	RO-0	RO-0	RW	RW	RW	RW	RO-0
4	T1CC4OF : 捕获/比较4重复捕获标记 参见CC1OF描述。							
3	T1CC3OF : 捕获/比较3重复捕获标记 参见CC1OF描述。							
2	T1CC2OF : 捕获/比较2重复捕获标记							

	参见CC10F描述。
1	<p>T1CC10F: 捕获/比较1重复捕获标记</p> <p>仅当相应的通道被配置为输入捕获时, 该标记可由硬件置1。写0可清除该位。</p> <p>0: 无重复捕获产生;</p> <p>1: 计数器的值被捕获到TIM1_CCR1寄存器时, CC1IF的状态已经为1。</p>

1.4.6. TIM1EGR, 地址: 0x218

Bit	7	6	5	4	3	2	1	0
Name	T1BG	T1TG	T1COMG	T1CC4G	T1CC3G	T1CC2G	T1CC1G	T1UG
Reset	0	0	0	0	0	0	0	0
Type	R0-W	R0-W	R0-W	R0-W	R0-W	R0-W	R0-W	R0-W
7	<p>T1BG: 产生刹车事件</p> <p>该位由软件置1, 用于产生一个刹车事件, 由硬件自动清0。</p> <p>0: 无动作;</p> <p>1: 产生一个刹车事件。此时MOE=0、BIF=1, 若开启对应的中断(BIE=1), 则产生相应的中断。</p>							
6	<p>T1TG: 产生触发事件</p> <p>该位由软件置1, 用于产生一个触发事件, 由硬件自动清0。</p> <p>0: 无动作;</p> <p>1: TIM1_SR寄存器的TIF=1, 若开启对应的中断(TIE=1), 则产生相应的中断。</p>							
5	<p>T1COMG: 捕获/比较事件, 产生控制更新该位由软件置1, 由硬件自动清0。</p> <p>0: 无动作;</p> <p>1: 当CCPC=1, 允许更新CCIE、CCINE、CciP, CciNP, OCIM位。</p> <p>注: 该位只对拥有互补输出的通道有效。</p>							
4	<p>T1CC4G: 产生捕获/比较4事件</p> <p>参考CC1G描述。</p>							
3	<p>T1CC3G: 产生捕获/比较3事件</p> <p>参考CC1G描述。</p>							
2	<p>T1CC2G: 产生捕获/比较2事件</p> <p>参考CC1G描述。</p>							
1	<p>T1CC1G: 产生捕获/比较1事件</p> <p>该位由软件置1, 用于产生一个捕获/比较事件, 由硬件自动清0。</p> <p>0: 无动作;</p> <p>1: 在通道CC1上产生一个捕获/比较事件: 若通道CC1配置为输出: 设置CC1IF=1, 若开启对应的中断, 则产生相应的中断。若通道CC1配置为输入: 当前的计数器值被捕获至TIM1_CCR1寄存器, 设置CC1IF=1, 若开启对应的中断, 则产生相应的中断。若CC1IF已经为1, 则设置CC10F=1。</p>							
0	<p>T1UG: 产生更新事件</p> <p>该位由软件置1, 由硬件自动清0。</p> <p>0: 无动作;</p> <p>1: 重新初始化计数器, 并产生一个更新事件。注意预分频器的计数器也被清0(但是预分频系数不变)。若在中心对称模式下或DIR=0(向上计数)则计数器被清0; 若DIR=1(向下计数)则计数器取TIM1_ARR的值。</p>							

1.4.7. TIM1CCMR1, 地址: 0x219

Bit	7	6	5	4	3	2	1	0
Name	T1OC1CE	T1OC1M[2:0]			T1OC1PE	T1OC1FE	T1CC1S[1:0]	
Reset	0	0	0	0	0	0	0	0
Type	RW	RW	RW	RW	RW	RW	RW	RW
7	T1OC1CE: 输出比较1清零使能 该位用于使能使用TIM1_TRIG引脚上的外部事件来清通道1的输出信号(OC1REF), 参考17.5.9在外部事件发生时清除OCREF信号 0: OC1REF 不受ETRF输入(来自TIM1_TRIG引脚)的影响; 1: 一旦检测到ETRF输入高电平, OC1REF=0。							
6:4	T1OC1M[2:0]: 输出比较1模式 该3位定义了输出参考信号OC1REF的动作, 而OC1REF决定了OC1的值。OC1REF是高电平有效, 而OC1的有效电平取决于CC1P位。 000: 冻结。输出比较寄存器TIM1_CCR1与计数器TIM1_CNT间的比较对OC1REF不起作用; 001: 匹配时设置通道1的输出为有效电平。当计数器TIM1_CNT的值与捕获/比较寄存器1(TIM1_CCR1)相同时, 强制OC1REF为高。 010: 匹配时设置通道1的输出为无效电平。当计数器TIM1_CNT的值与捕获/比较寄存器1(TIM1_CCR1)相同时, 强制OC1REF为低。 011: 翻转。当TIM1_CCR1=TIM1_CNT时, 翻转OC1REF的电平。 100: 强制为无效电平。强制OC1REF为低。 101: 强制为有效电平。强制OC1REF为高。 110: PWM模式1— 在向上计数时, 一旦TIM1_CNT<TIM1_CCR1时通道1为有效电平, 否则为无效电平; 在向下计数时, 一旦TIM1_CNT>TIM1_CCR1时通道1为无效电平(OC1REF=0), 否则为有效电平(OC1REF=1)。 111: PWM模式2— 在向上计数时, 一旦TIM1_CNT<TIM1_CCR1时通道1为无效电平, 否则为有效电平; 在向下计数时, 一旦TIM1_CNT>TIM1_CCR1时通道1为有效电平, 否则为无效电平。 注1: 一旦LOCK级别设为3(TIM1_BKR寄存器中的LOCK位)并且CC1S=00(该通道配置成输出) 则该位不能被修改。 注2: 在PWM模式1或PWM模式2中, 只有当比较结果改变了或在输出比较模式中从冻结模式切换到PWM模式时, OC1REF电平才改变。(参考17.5.7PWM模式) 注3: 在有互补输出的通道上, 这些位是预装载的。如果TIM1_CR2寄存器的CCPC=1, OCM 位只有在COM事件发生时, 才从预装载位取新值。							
3	T1OC1PE: 输出比较1预装载使能 0: 禁止TIM1_CCR1寄存器的预装载功能, 可随时写入TIM1_CCR1寄存器, 并且新写入的数值立即起作用。 1: 开启TIM1_CCR1寄存器的预装载功能, 读写操作仅对预装载寄存器操作, TIM1_CCR1的预装载值在更新事件到来时被加载至当前寄存器中。 注1: 一旦LOCK级别设为3(TIM1_BKR寄存器中的LOCK位)并且CC1S=00(该通道配置成输出) 则该位不能被修改。 注2: 为了操作正确, 在PWM模式下必须使能预装载功能。但在单脉冲模式下(TIM1_CR1寄存器的OPM=1), 它不是必须的。							
2	T1OC1FE: 输出比较1 快速使能 该位用于加快CC输出对触发输入事件的响应。 0: 根据计数器与CCR1的值, CC1正常操作, 即使触发器是打开的。当触发器的输入有一个有效沿时, 激活CC1输出的最小延时为5个时钟周期。 1: 输入到触发器的有效沿的作用就象发生了一次比较匹配。因此, OC被设置为比较电平而与比较结果无关。采样触发器的有效沿和CC1输出间的延时被缩短为3个时钟周期。 OCFE只在通道被配置成PWM1或PWM2模式时起作用。							
1:0	T1CC1S[1:0]: 捕获/比较1 选择。 这2位定义通道的方向(输入/输出), 及输入脚的选择:							

	<p>00: CC1通道被配置为输出;</p> <p>01: CC1通道被配置为输入, IC1映射在TI1FP1上;</p> <p>10: CC1通道被配置为输入, IC1映射在TI2FP1上;</p> <p>11: CC1通道被配置为输入, IC1映射在TRC上。此模式仅工作在内部触发器输入被选中时(由TIM1_SMCR寄存器的TS位选择)。</p> <p>注: CC1S仅在通道关闭时(TIM1_CCER1寄存器的CC1E=0)才是可写的。</p>
--	---

Name	TI1C1F[3:0]				TI1C1PSC[1:0]		TI1CC1S[1:0]	
Type	RW	RW	RW	RW	RW	RW	RW	RW
7:4	<p>TI1C1F[3:0]: 输入捕获1滤波器</p> <p>这几位定义了TI1输入的采样频率及数字滤波器长度。数字滤波器由一个事件计数器组成, 只有发生了N个事件后输出的跳变才被认为有效。</p> <p>0000: 无滤波器, $f_{\text{SAMPLING}}=f_{\text{MASTER}}$ 1000: 采样频率$f_{\text{SAMPLING}}=f_{\text{MASTER}}/8$, $N=6$</p> <p>0001: 采样频率$f_{\text{SAMPLING}}=f_{\text{MASTER}}$, $N=2$ 1001: 采样频率$f_{\text{SAMPLING}}=f_{\text{MASTER}}/8$, $N=8$</p> <p>0010: 采样频率$f_{\text{SAMPLING}}=f_{\text{MASTER}}$, $N=4$ 1010: 采样频率$f_{\text{SAMPLING}}=f_{\text{MASTER}}/16$, $N=5$</p> <p>0011: 采样频率$f_{\text{SAMPLING}}=f_{\text{MASTER}}$, $N=8$ 1011: 采样频率$f_{\text{SAMPLING}}=f_{\text{MASTER}}/16$, $N=6$</p> <p>0100: 采样频率$f_{\text{SAMPLING}}=f_{\text{MASTER}}/2$, $N=6$ 1100: 采样频率$f_{\text{SAMPLING}}=f_{\text{MASTER}}/16$, $N=8$</p> <p>0101: 采样频率$f_{\text{SAMPLING}}=f_{\text{MASTER}}/2$, $N=8$ 1101: 采样频率$f_{\text{SAMPLING}}=f_{\text{MASTER}}/32$, $N=5$</p> <p>0110: 采样频率$f_{\text{SAMPLING}}=f_{\text{MASTER}}/4$, $N=6$ 1110: 采样频率$f_{\text{SAMPLING}}=f_{\text{MASTER}}/32$, $N=6$</p> <p>0111: 采样频率$f_{\text{SAMPLING}}=f_{\text{MASTER}}/4$, $N=8$ 1111: 采样频率$f_{\text{SAMPLING}}=f_{\text{MASTER}}/32$, $N=8$</p> <p>注: 即使对于带互补输出的通道, 该位域也是非预装载的, 并且不会考虑CCPC (TIM1_CR2 寄存器) 的值。</p>							
3:2	<p>TI1C1PSC[1:0]: 输入/捕获1预分频器</p> <p>这2位定义了CC1输入(IC1)的预分频系数。</p> <p>一旦CC1E=0(TIM1_CCER寄存器中), 则预分频器复位。</p> <p>00: 无预分频器, 捕获输入上检测到的每一个边沿都触发一次捕获;</p> <p>01: 每2个事件触发一次捕获;</p> <p>10: 每4个事件触发一次捕获;</p> <p>11: 每8个事件触发一次捕获。</p>							
1:0	<p>TI1CC1S[1:0]: 捕获/比较1 选择。</p> <p>这2位定义通道的方向(输入/输出), 及输入脚的选择:</p> <p>00: CC1通道被配置为输出;</p> <p>01: CC1通道被配置为输入, IC1映射在TI1FP1上;</p> <p>10: CC1通道被配置为输入, IC1映射在TI2FP1上;</p> <p>11: CC1通道被配置为输入, IC1映射在TRC上。此模式仅工作在内部触发器输入被选中时(由TIM1_SMCR寄存器的TS位选择)。</p> <p>注: CC1S仅在通道关闭时(TIM1_CCER1寄存器的CC1E=0)才是可写的。</p>							

1.4.8. TIM1CCMR2, 地址: 0x21A

Bit	7	6	5	4	3	2	1	0
Name	TI0C2CE	TI0C2M[2:0]			TI0C2PE	TI0C2FE	TI1CC2S[1:0]	
Reset	0	0	0	0	0	0	0	0
Type	RW	RW	RW	RW	RW	RW	RW	RW
7	TI0C2CE: 输出比较2清零使能							

6:4	T1OC2M[2:0] : 输出比较2模式
3	T1OC2PE : 输出比较2预装载使能
2	T1OC1FE : 输出比较2快速使能
1:0	T1CC2S[1:0] : 捕获/比较2选择。 该位定义通道的方向(输入/输出), 及输入脚的选择: 00: CC2通道被配置为输出; 01: CC2通道被配置为输入, IC2映射在TI2FP2上; 10: CC2通道被配置为输入, IC2映射在TI1FP2上; 11: 预留 注: CC2S仅在通道关闭时(TIM1_CCER1寄存器的CC2E=0, CC2NE=0且已被更新)才是可写的。

Name	T1IC2F[3:0]				T1IC2PSC[1:0]		T1CC2S[1:0]	
Type	RW	RW	RW	RW	RW	RW	RW	RW
7:4	T1IC2F[3:0] : 输入捕获2滤波器							
3:2	T1IC2PSC[1:0] : 输入/捕获2预分频器							
1:0	T1CC2S[1:0] : 捕获/比较2选择。 这2位定义通道的方向(输入/输出), 及输入脚的选择: 00: CC2通道被配置为输出; 01: CC2通道被配置为输入, IC2映射在TI2FP2上; 10: CC2通道被配置为输入, IC2映射在TI1FP2上; 11: CC2通道被配置为输入, IC2映射在TRC上。此模式仅工作在内部触发器输入被选中时(由TIM1_SMCR寄存器的TS位选择)。 注: CC2S仅在通道关闭时(TIM1_CCER1寄存器的CC2E=0, CC2NE=0且已被更新)才是可写的。							

1.4.9. TIM1CCMR3, 地址: 0x21B

Bit	7	6	5	4	3	2	1	0
Type	T1OC3CE	T1OC3M[2:0]			T1OC3PE	T1OC3FE	T1CC3S[1:0]	
Reset	0	0	0	0	0	0	0	0
Type	RW	RW	RW	RW	RW	RW	RW	RW
7	T1OC3CE : 输出比较3清零使能 该位用于使能使用TIM1_TRIG引脚上的外部事件来清通道3的输出信号(OC3REF), 参考17.5.9 在外部事件发生时清除OCREF信号 0: OC3REF 不受ETRF输入(来自TIM1_TRIG引脚)的影响; 1: 一旦检测到ETRF输入高电平, OC3REF=0。							
6:4	T1OC3M[2:0] : 输出比较3模式							
3	T1OC3PE : 输出比较3预装载使能							
2	T1OC3FE : 输出比较3快速使能							
1:0	T1CC3S[1:0] : 捕获/比较3选择。 该位定义通道的方向(输入/输出), 及输入脚的选择: 00: CC3通道被配置为输出; 01: CC3通道被配置为输入, IC3映射在TI3FP3上;							

	10: CC3通道被配置为输入, IC3映射在TI4FP3上; 11: 预留 注: CC3S仅在通道关闭时(TIM1_CCER2寄存器的CC3E=0, CC3NE=0且已被更新)才是可写的。
--	---

Name	T1IC3F[3:0]				T1IC3PSC[1:0]		T1CC3S[1:0]	
Type	RW	RW	RW	RW	RW	RW	RW	RW
7:4	T1IC3F[3:0]: 输入捕获3滤波器							
3:2	T1IC3PSC[1:0]: 输入/捕获3预分频器							
1:0	T1CC3S[1:0]: 捕获/比较3选择。 这2位定义通道的方向(输入/输出), 及输入脚的选择: 00: CC3通道被配置为输出; 01: CC3通道被配置为输入, IC3映射在TI3FP3上; 10: CC3通道被配置为输入, IC3映射在TI4FP3上; 11: 预留 注: CC3S仅在通道关闭时(TIM1_CCER2寄存器的CC3E=0, CC3NE=0且已被更新)才是可写的。							

1.4.10. TIM1CCMR4, 地址: 0x21C

Bit	7	6	5	4	3	2	1	0
Name	T1OC4CE	T1OC4M[2:0]			T1OC4PE	T1OC4FE	T1CC4S[1:0]	
Reset	0	0	0	0	0	0	0	0
Type	RW	RW	RW	RW	RW	RW	RW	RW
7	T1OC4CE: 输出比较4清零使能 该位用于使能使用TIM1_TRIG引脚上的外部事件来清通道4的输出信号(OC4REF), 参考17.5.9 在外部事件发生时清除OCREF信号 0: OC4REF 不受ETRF输入(来自TIM1_TRIG引脚)的影响; 1: 一旦检测到ETRF输入高电平, OC4REF=0。							
6:4	T1OC4M[2:0]: 输出比较4模式							
3	T1OC4PE: 输出比较4预装载使能							
2	T1OC4FE: 输出比较4快速使能							
1:0	T1CC4S[1:0]: 捕获/比较4选择。 该位定义通道的方向(输入/输出), 及输入脚的选择: 00: CC4通道被配置为输出; 01: CC4通道被配置为输入, IC4映射在TI3FP4上; 10: CC4通道被配置为输入, IC4映射在TI4FP4上; 11: 预留 注: CC4S仅在通道关闭时(TIM1_CCER2寄存器的CC4E=0, CC4NE=0且已被更新)才是可写的。							

Name	T1IC4F[3:0]				T1IC4PSC[1:0]		T1CC4S[1:0]	
Type	RW	RW	RW	RW	RW	RW	RW	RW
7:4	T1IC4F[3:0]: 输入捕获4滤波器							

3:2	T1IC4PSC[1:0] : 输入/捕获4预分频器
1:0	T1CC4S[1:0] : 捕获/比较4选择。 这2位定义通道的方向(输入/输出), 及输入脚的选择: 00: CC4通道被配置为输出; 01: CC4通道被配置为输入, IC4映射在TI3FP4上; 10: CC4通道被配置为输入, IC4映射在TI4FP4上; 11: 预留 注: CC4S仅在通道关闭时(TIM1_CCER2寄存器的CC4E=0, CC4NE=0且已被更新)才是可写的。

1.4.11. TIM1CCER1, 地址: 0x21D

Bit	7	6	5	4	3	2	1	0
Name	T1CC2NP	T1CC2NE	T1CC2P	T1CC2E	T1CC1NP	T1CC1NE	T1CC1P	T1CC1E
Reset	0	0	0	0	0	0	0	0
Type	RW	RW	RW	RW	RW	RW	RW	RW
7	T1CC2NP : 输入捕获/比较2互补输出极性。参考CC1NP的描述。							
6	T1CC2NE : 输入捕获/比较2互补输出使能。参考CC1NE的描述。							
5	T1CC2P : 输入捕获/比较2输出极性。参考CC1P的描述。							
4	T1CC2E : 输入捕获/比较2输出使能。参考CC1E的描述。							
3	T1CC1NP : 输入捕获/比较1互补输出极性 0: OC1N高电平有效; 1: OC1N低电平有效。 注1: 一旦LOCK级别(TIM1_BKR寄存器中的LCCK位)设为3或2且CC1S=00(通道配置为输出) 则该位不能被修改。 注2: 对于有互补输出的通道, 该位是预装载的。如果CCPC=1 (TIM1_CR2寄存器), 只有在COM事件发生时, CC1NP位才从预装载位中取新值。							
2	T1CC1NE : 输入捕获/比较1互补输出使能 0: 关闭— OC1N禁止输出, 因此OC1N的输出电平依赖于MOE、OSSI、OSSR、OIS1、OIS1N和CC1E位的值。 1: 开启— OC1N信号输出到对应的输出引脚, 其输出电平依赖于MOE、OSSI、OSSR、OIS1、OIS1N和CC1E位的值。 注: 对于有互补输出的通道, 该位是预装载的。如果CCPC=1(TIM1_CR2寄存器), 只有在COM事件发生时, CC1NE位才从预装载位中取新值。							
1	T1CC1P : 输入捕获/比较1输出极性CC1通道配置为输出: 0: OC1高电平有效; 1: OC1低电平有效。 CC1通道配置为触发(参考图61): 0: 触发发生在TI1F的高电平或上升沿; 1: 触发发生在TI1F的低电平或下降沿。 CC1通道配置为输入(参考图61): 0: 捕捉发生在TI1F的高电平或上升沿; 1: 捕捉发生在TI1F的低电平或下降沿。 注1: 一旦LOCK级别(TIM1_BKR寄存器中的LCCK位)设为3或2, 则该位不能被修改。 注2: 对于有互补输出的通道, 该位是预装载的。如果CCPC=1 (TIM1_CR2寄存器), 只有在							

	COM事件发生时，CC1P位才从预装载位中取新值。
0	<p>T1CC1E: 输入捕获/比较1输出使能</p> <p>CC1通道配置为输出:</p> <p>0: 关闭— OC1禁止输出, 因此OC1的输出电平依赖于MOE、OSSI、OSSR、OIS1、OIS1N和CC1NE位的值。</p> <p>1: 开启— OC1信号输出到对应的输出引脚, 其输出电平依赖于MOE、OSSI、OSSR、OIS1、OIS1N和CC1NE位的值。</p> <p>CC1通道配置为输入:</p> <p>该位决定了计数器的值是否能捕获入TIM1_CCR1寄存器。</p> <p>0: 捕获禁止;</p> <p>0: 捕获使能。</p> <p>注: 对于有互补输出的通道, 该位是预装载的。如果CCPC=1(TIM1_CR2寄存器), 只有在COM事件发生时, CC1E位才从预装载位中取新值。</p>

1.4.12. TIM1CCER2, 地址: 0x21E

Bit	7	6	5	4	3	2	1	0
Name	reserved	reserved	T1CC4P	T1CC4E	T1CC3NP	T1CC3NE	T1CC3P	T1CC3E
Reset	—	—	0	0	0	0	0	0
Type	RO-0	RO-0	RW	RW	RW	RW	RW	RW
5	T1CC4P: 输入捕获/比较4输出极性。参考CC1P的描述。							
4	T1CC4E: 输入捕获/比较4输出使能。参考CC1E 的描述。							
3	T1CC3NP: 输入捕获/比较3互补输出极性。参考CC1NP的描述。							
2	T1CC3NE: 输入捕获/比较3互补输出使能。参考CC1NE的描述。							
1	T1CC3P: 输入捕获/比较3输出极性。参考CC1P的描述。							
0	T1CC3E: 输入捕获/比较3输出使能。参考CC1E 的描述。							

1.4.13. TIM1CNTRH, 地址: 0x28C

Bit	7	6	5	4	3	2	1	0
Name	T1CNT[15:8]							
Reset	0	0	0	0	0	0	0	0
Type	RW	RW	RW	RW	RW	RW	RW	RW
7:0	T1CNT[15:8]: 计数器的高8位值							

1.4.14. TIM1CNTRL, 地址: 0x28D

Bit	7	6	5	4	3	2	1	0
Name	T1CNT[7:0]							
Reset	0	0	0	0	0	0	0	0

Type	RW	RW	RW	RW	RW	RW	RW	RW
7:0	T1CNT[7:0]: 计数器的低8位值							

1.4.15. TIM1PSCRH, 地址: 0x28E

Bit	7	6	5	4	3	2	1	0
Name	T1PSC[15:8]							
Reset	0	0	0	0	0	0	0	0
Type	RW	RW	RW	RW	RW	RW	RW	RW
7:0	T1PSC[15:7]: 预分频器的高8位值 预分频器用于对CK_PSC进行分频。 计数器的时钟频率(f_{CK_CNT})等于 $f_{CK_PSC}/(PSCR[15:0]+1)$ 。PSCR 包含了当更新事件产生时装入当前预分频器寄存器的值(更新事件包括计数器被TIM_EGR的UG位清0或被工作在复位模式的从控制器清0)。这意味着为了使新的值起作用,必须产生一个更新事件。							

1.4.16. TIM1PSCRL, 地址: 0x28F

Bit	7	6	5	4	3	2	1	0
Name	T1PSC[7:0]							
Reset	0	0	0	0	0	0	0	0
Type	RW	RW	RW	RW	RW	RW	RW	RW
7:0	T1PSC[7:0]: 预分频器的低8位值 预分频器用于对CK_PSC进行分频。 计数器的时钟频率(f_{CK_CNT})等于 $f_{CK_PSC}/(PSCR[15:0]+1)$ 。 PSCR 包含了当更新事件产生时装入当前预分频器寄存器的值(更新事件包括计数器被TIM_EGR的UG位清0或被工作在复位模式的从控制器清0)。即: 为了使新的值起作用,必须产生一个更新事件。							

1.4.17. TIM1ARRH, 地址: 0x290

Bit	7	6	5	4	3	2	1	0
Name	T1ARR[15:8]							
Reset	1	1	1	1	1	1	1	1
Type	RW	RW	RW	RW	RW	RW	RW	RW
7:0	T1ARR[15:8]: 自动重载的高8位值 ARR包含了将要装载入实际的自动重载寄存器的值。 当自动重载的值为空时,计数器不工作。							

1.4.18. TIM1ARRL, 地址: 0x291

Bit	7	6	5	4	3	2	1	0
Name	T1ARR[7:0]							
Reset	1	1	1	1	1	1	1	1
Type	RW	RW	RW	RW	RW	RW	RW	RW
7:0	T1ARR[7: 0]: 自动重载的低8位值 ARR包含了将要装载入实际的自动重载寄存器的值。 当自动重载的值为空时, 计数器不工作。							

1.4.19. TIM1RCR, 地址: 0x292

Bit	7	6	5	4	3	2	1	0
Name	T1REP[7:0]							
Reset	0	0	0	0	0	0	0	0
Type	RW	RW	RW	RW	RW	RW	RW	RW
7:0	T1REP[7:0]: 重复计数器的值 开启了预装载功能后, 这些位允许用户设置比较寄存器的更新速率(即周期性地从预装载寄存器传输到当前寄存器); 如果允许产生更新中断, 则会同时影响产生更新中断的速率。 每次向下计数器REP_CNT达到0, 会产生一个更新事件并且计数器REP_CNT重新从REP值开始计数。由于REP_CNT只有在周期更新事件U_RC发生时才重载REP值, 因此对TIM1_RCR寄存器写入的新值只在下次周期更新事件发生时才起作用。 这意味着在PWM模式中, (REP+1)对应着: --- 在边沿对齐模式下, PWM周期的数目; --- 在中心对称模式下, PWM半周期的数目;							

1.4.20. TIM1CCR1H, 地址: 0x293

Bit	7	6	5	4	3	2	1	0
Name	T1CCR1[15:8]							
Reset	0	0	0	0	0	0	0	0
Type	RW	RW	RW	RW	RW	RW	RW	RW
7:0	T1CCR1[15:8]: 捕获/比较1的高8位值 若CC1通道配置为输出(TIM1_CCMR1的CC1S位): CCR1包含了装入当前捕获/比较1寄存器的值(预装载值)。 如果在TIM1_CCMR1寄存器(OC1PE位)中未选择预装载功能, 写入的数值会立即传输至当前寄存器中。否则只有当更新事件发生时, 此预装载值才传输至当前捕获/比较1寄存器中。 当前捕获/比较寄存器的值同计数器TIM1_CNT的值相比较, 并在OC1端口上产生输出信号。若CC1通道配置为输入: CCR1包含了上一次输入捕获1事件(IC1)发生时的计数器值(此时该寄存器为只读)。							

1.4.21. TIM1CCR1L, 地址: 0x294

Bit	7	6	5	4	3	2	1	0
Name	T1CCR1[7:0]							
Reset	RW	RW	RW	RW	RW	RW	RW	RW
7:0	T1CCR1[7:0]: 捕获/比较1的低8位值							

1.4.22. TIM1CCR2H, 地址: 0x295

Bit	7	6	5	4	3	2	1	0
Name	T1CCR2[15:8]							
Reset	0	0	0	0	0	0	0	0
Type	RW	RW	RW	RW	RW	RW	RW	RW
7:0	T1CCR2[15:8]: 捕获/比较2的高8位值 若CC2通道配置为输出(TIM1_CCMR2的CC2S位): CCR2包含了装入当前捕获/比较2寄存器的值(预装载值)。 如果在TIM1_CCMR2寄存器(OC2PE位)中未选择预装载功能, 写入的数值会立即传输至当前寄存器中。否则只有当更新事件发生时, 此预装载值才传输至当前捕获/比较1寄存器中。 当前捕获/比较寄存器的值同计数器TIM1_CNT的值相比较, 并在OC2端口上产生输出信号。若CC2通道配置为输入: CCR2包含了由上一次输入捕获2事件(IC2)传输的计数器值(此时该寄存器为只读)。							

1.4.23. TIM1CCR2L, 地址: 0x296

Bit	7	6	5	4	3	2	1	0
Name	T1CCR2[7:0]							
Reset	0	0	0	0	0	0	0	0
Type	RW	RW	RW	RW	RW	RW	RW	RW
7:0	T1CCR2[7:0]: 捕获/比较1的低8位值							

1.4.24. TIM1CCR3H, 地址: 0x297

Bit	7	6	5	4	3	2	1	0
Name	T1CCR3[15:8]							
Reset	0	0	0	0	0	0	0	0

Type	RW	RW	RW	RW	RW	RW	RW	RW
7:0	T1CCR3[15:8]: 捕获/比较3的高8位值 若CC3通道配置为输出(TIM1_CCMR3的CC3S位): CCR3包含了装入当前捕获/比较3寄存器的值(预装载值)。 如果在TIM1_CCMR3寄存器(OC3PE位)中未选择预装载功能, 写入的数值会立即传输至当前寄存器中。否则只有当更新事件发生时, 此预装载值才传输至当前捕获/比较1寄存器中。 当前捕获/比较寄存器的值同计数器TIM1_CNT的值相比较, 并在OC3端口上产生输出信号。若CC3通道配置为输入: CCR3包含了由上一次输入捕获3事件(IC3)传输的计数器值(此时该寄存器为只读)。							

1.4.25. TIM1CCR3L, 地址: 0x298

Bit	7	6	5	4	3	2	1	0
Name	T1CCR3[7:0]							
Reset	0	0	0	0	0	0	0	0
Type	RW	RW	RW	RW	RW	RW	RW	RW
7:0	T1CCR3[7:0]: 捕获/比较3的低8位值							

1.4.26. TIM1CCR4H, 地址: 0x299

Bit	7	6	5	4	3	2	1	0
Name	T1CCR4[15:8]							
Reset	0	0	0	0	0	0	0	0
Type	RW	RW	RW	RW	RW	RW	RW	RW
7:0	T1CCR4[15:8]: 捕获/比较4的高8位值 若CC4通道配置为输出(TIM1_CCMR4的CC4S位): CCR4包含了装入当前捕获/比较4寄存器的值(预装载值)。 如果在TIM1_CCMR4寄存器(OC4PE位)中未选择预装载功能, 写入的数值会立即传输至当前寄存器中。否则只有当更新事件发生时, 此预装载值才传输至当前捕获/比较1寄存器中。 当前捕获/比较寄存器的值同计数器TIM1_CNT的值相比较, 并在OC4端口上产生输出信号。若CC4通道配置为输入: CCR4包含了由上一次输入捕获4事件(IC4)传输的计数器值(此时该寄存器为只读)。							

1.4.27. TIM1CCR4L, 地址: 0x29A

Bit	7	6	5	4	3	2	1	0
Name	T1CCR4[7:0]							
Reset	0	0	0	0	0	0	0	0
Type	RW	RW	RW	RW	RW	RW	RW	RW

7:0	T1CCR4[7:0]: 捕获/比较3的低8位值
-----	--------------------------

1.4.28. TIM1BRK, 地址: 0x29B

Bit	7	6	5	4	3	2	1	0
Name	T1MOE	T1AOE	T1BKP	T1BKE	T1OSSR	T1OSSI	T1LOCK[1:0]	
Reset	0	0	0	0	0	0	0	0
Type	RW	RW	RW	RW	RW	RW	RW	RW
7	T1MOE: 主输出使能 一旦刹车输入有效, 该位被硬件异步清0。根据AOE位的设置值, 该位可以由软件置1或被自动置1。它仅对配置为输出的通道有效。 0: 禁止OC和OCN输出或强制为空闲状态; 1: 如果设置了相应的使能位(TIM1_CCERX寄存器的CCIE位), 则使能OC和OCN输出。							
6	T1AOE: 自动输出使能 0: MOE只能被软件置1; 1: MOE能被软件置1或在下一个更新事件被自动置1(如果刹车输入无效)。 注: 一旦LOCK级别(TIM1_BKR寄存器中的LOCK位)设为1, 则该位不能被修改。							
5	T1BKP: 刹车输入极性 (只对故障源TIM1_BKIN有效) 0: 刹车输入低电平有效; 1: 刹车输入高电平有效。 注: 一旦LOCK级别(TIM1_BKR寄存器中的LOCK位)设为1, 则该位不能被修改。							
4	T1BKE: 刹车功能使能 0: 禁止刹车输入(BRK); 1: 开启刹车输入(BRK)。 注: 一旦LOCK级别(TIM1_BKR寄存器中的LOCK位)设为1, 则该位不能被修改。							
3	T1OSSR: 运行模式下“关闭状态”选择 该位用于当MOE=1且通道为互补输出时。 参考OC/OCN使能的详细说明(参见17.7.13)。 0: 当定时器不工作时, 禁止OC/OCN输出(OC/OCN使能输出信号=0); 1: 当定时器不工作时, 一旦CciE=1或CciNE=1, 首先开启OC/OCN并输出无效电平, 然后置OC/OCN使能输出信号=1。 注: 一旦LOCK级别(TIM1_BKR寄存器中的LOCK位)设为2, 则该位不能被修改。							
2	T1OSSI: 空闲模式下“关闭状态”选择 该位用于当MOE=0且通道设为输出时。 参考OC/OCN使能的详细说明(参见17.7.13)。 0: 当定时器不工作时, 禁止OC/OCN输出(OC/OCN使能输出信号=0); 1: 当定时器不工作时, 一旦CciE=1或CciNE=1, OC/OCN首先输出其空闲电平, 然后OC/OCN使能输出信号=1。 注: 一旦LOCK级别(TIM1_BKR寄存器中的LOCK位)设为2, 则该位不能被修改。							
1:0	T1LOCK[1:0]: 锁定设置 该位为防止软件错误而提供写保护。 00: 锁定关闭, 寄存器无写保护; 01: 锁定级别1, 不能写入TIM1_BKR寄存器的BKE、BKP、AOE位和TIM1_OISR寄存器的							

	<p>OISI位;</p> <p>10: 锁定级别2, 不能写入锁定级别1中的各位, 也不能写入CC极性位(一旦相关通道通过CCIS位设为输出, CC极性位是TIM1_CCERX寄存器的CCIP位)以及OSSR/OSSI位;</p> <p>11: 锁定级别3, 不能写入锁定级别2中的各位, 也不能写入CC控制位(一旦相关通道通过CCIS位设为输出, CC控制位是TIM1_CCMRx寄存器的OCIM/OCIBE位);</p> <p>注: 在系统复位后, 只能写一次LOCK位, 一旦写入TIM1_BDR寄存器, 则其内容保持不变直至复位。</p>
--	--

1.4.29. TIM1DTR, 地址: 0x29C

Bit	7	6	5	4	3	2	1	0
Name	T1DTR[7:0]							
Reset	0	0	0	0	0	0	0	0
Type	RW	RW	RW	RW	RW	RW	RW	RW
7:0	<p>T1UTG[7:0]: 死区发生器设置</p> <p>这些位定义了插入互补输出之间的死区持续时间。假设DT表示其持续时间, t_{CK_PSC}为TIM1的时钟脉冲:</p> <p>DTG[7:5]=0xx => DT=DTG[7:0]x tdtg, 其中: t_{dtg}=t_{CK_PSC}. (f1)</p> <p>DTG[7:5]=10x => DT=(64+DTG[5:0])x tdtg, 其中: t_{dtg}=t_{CK_PSC}. (f2)</p> <p>DTG[7:5]=110 => DT=(32+DTG[4:0])x tdtg, 其中: tdtg=8x t_{CK_PSC}. (f3)</p> <p>DTG[7:5]=111 => DT=(32+DTG[4:0])x tdtg, 其中: tdtg=16x t_{CK_PSC}. (f4)</p> <p>举例:</p> <p>如果t_{CK_PSC}=125 ns (8 MHz), 可能的死区时间为:</p> <p>DTG[7:0]=0 到 7Fh, 0 到 15875 ns, 步长时间为 125 ns (参考 f1), DTG[7:0]=80h 到 BFh, 16 μs 到 31750 ns, 步长时间为250 ns (参考f2), DTG[7:0]=C0h 到 DFh, 32 μs 到 63 μs, 步长时间为 1μs (参考 f3), DTG[7:0]=E0h 到 FFh, 64 μs 到 126 μs, 步长时间为2 μs (参考f4),</p> <p>注: 一旦LOCK级别(TIM1_BKR寄存器中的LOCK位)设为1、2或3, 则不能修改这些位。</p>							

1.4.30. TIM1OISR, 地址: 0x29D

Bit	7	6	5	4	3	2	1	0
Name	reserved	T1OIS4	T1OIS3N	T1OIS3	T1OIS2N	T1OIS2	T1OIS1N	T1OIS1
Reset	—	0	0	0	0	0	0	0
Type	RO-0	RW	RW	RW	RW	RW	RW	RW
6	T1OIS4: 输出空闲状态4(OC4输出)。参见OIS1位。							
5	T1OIS3N: 输出空闲状态3(OC3N输出)。参见OIS1N位。							
4	T1OIS3: 输出空闲状态3(OC3输出)。参见OIS1位。							
3	T1OIS2N: 输出空闲状态2(OC2N输出)。参见OIS1N位。							
2	T1OIS2: 输出空闲状态2(OC2输出)。参见OIS1位。							
1	<p>T1OIS1N: 输出空闲状态1(OC1N输出)。</p> <p>0: 当MOE=0时, 则在一个死区时间后, OC1N=0;</p> <p>1: 当MOE=0时, 则在一个死区时间后, OC1N=1。</p>							

	注：已经设置了LOCK(TIM1_BKR寄存器)级别1、2或3后，该位不能被修改。
0	T1OIS1 ：输出空闲状态1(OC1输出)。 0：当MOE=0时，如果OC1N使能，则在一个死区后，OC1=0； 1：当MOE=0时，如果OC1N使能，则在一个死区后，OC1=1。 注：已经设置了LOCK(TIM1_BKR寄存器)级别1、2或3后，该位不能被修改。

1.4.31. LEBCON 寄存器，地址 0x41C

Bit	7	6	5	4	3	2	1	0
Name	LEBEN	LEBCH[1:0]		reserved	EDGS	BKS[2:0]		
Reset	0	0	0	—	0	0	0	0
Type	RW	RW	RW	RO-0	RW	RW	RW	RW

Bit	Name	Function
7	LEBEN	前沿消隐使能位（仅当 ADGO=0 时可进行切换，否则 ADC 工作异常） 1 = 使能 0 = 禁止
6:5	LEBCH[1:0]	前沿消隐通道选择 00 = TIM1_CH1 01 = TIM1_CH2 10 = TIM1_CH3 11 = TIM1_CH4
4	N/A	保留位，读 0
3	EDGS	PWM 消隐沿选择 0 = PWM 上升沿 1 = PWM 下降沿
2	BKS[2:0]	BKS[2:0]，TIM1 的故障源使能，高有效 BKS2：选择 ADC 阈值比较 BKS1：选择 LVD 检测 BKS0：选择 BKIN 管脚

2 应用范例

//**

/* 文件名: TEST_62F08x_TIM1_CAPTURE.c

* 功能： FT62F08x_TIM1_CAPTURE 功能演示

* IC： FT62F088 LQFP32

* 内部： 16M

* empno: 500

* 说明： 例程通过 TIM1 的捕获通道 1 测量 TIM4 输出到 RB3 的波形的周期（4K）

* TIM1 计数器捕获的数据放在 testdata 数组里，一共 20 个。

* 程序只用到一个数据(testdata[1])来计算周期。

```

*
* 参考原理图 TEST_62F08x_sch.pdf
*/
//*****
#include "SYSCFG.h"
//*****
//*****宏定义*****
#define unchar    unsigned char
#define uint      unsigned int
#define ulong     unsigned long

#define DemoPortOut RB3
#define DemoPortOut1 RB4

//volatile unchar mydata; //全局查看变量定义
volatile uint testdata[20];
volatile unchar y=0;
volatile unchar dataH;
volatile unchar dataL;
volatile uint period=0;
/*-----
* 函数名: 中断
* 功能:
* 输入: 无
* 输出: 无
-----*/
void interrupt ISR(void)
{
    //定时器 4 的中断处理*****
    if(T4UIE && T4UIF)
    {
        T4UIF = 1; //写 1 清零标志位
        DemoPortOut = ~DemoPortOut; //翻转电平
    }
    if(T1CC1IE && T1CC1IF) //捕获中断
    {
        T1CC1IF=1;

        T1CC1OF=0;
        DemoPortOut1 = ~DemoPortOut1; //捕获中断调试查看
        dataH = TIM1CCR1H;
        dataL = TIM1CCR1L;
        testdata[y++]=(uint)(dataH<<8 | dataL);
    }
}

```

```

NOP();

if(y>=20)
{
    y=0;

    //计算周期(us)
    period = testdata[1]/32;    //周期约等于 250us
}
}
}
}
/*-----
* 函数名: POWER_INITIAL
* 功能:   上电系统初始化
* 输入:   无
* 输出:   无
-----*/
void POWER_INITIAL (void)
{

    OSCCON = 0B01110001;    //WDT 32KHZ IRCF=111=16MHZ
                           //Bit0=1,系统时钟为内部振荡器
                           //Bit0=0,时钟源由 FOSC<2: 0>决定即编译选项时选择

    INTCON = 0;            //暂禁止所有中断

    PORTA = 0B00000000;
    TRISA = 0B11111111;    //PA 输入输出 0-输出 1-输入
    PORTB = 0B00000000;
    TRISB = 0B11100111;    //PB 输入输出 0-输出 1-输入
    PORTC = 0B00000000;
    TRISC = 0B11111111;    //PC 输入输出 0-输出 1-输入
    PORTD = 0B00000000;
    TRISD = 0B11111111;    //PD 输入输出 0-输出 1-输入

    WPUA = 0B00000000;    //PA 端口上拉控制 1-开上拉 0-关上拉
    WPUB = 0B00000000;    //PB 端口上拉控制 1-开上拉 0-关上拉
    WPUC = 0B00000000;    //PC 端口上拉控制 1-开上拉 0-关上拉
    WPUD = 0B00000000;    //PD 端口上拉控制 1-开上拉 0-关上拉

    WPDA = 0B00000000;    //PA 端口上拉控制 1-开下拉 0-关下拉
    WPDB = 0B00000000;    //PB 端口上拉控制 1-开下拉 0-关下拉
    WPDC = 0B00000000;    //PC 端口上拉控制 1-开下拉 0-关下拉
    WPDD = 0B00000000;    //PD 端口上拉控制 1-开下拉 0-关下拉

```

```

PSRC0 = 0B11111111;    //PORTA,PORTB 源电流设置最大
//BIT7~BIT6:PORTB[7:4]源电流能力控制,BIT5~BIT4:PORTB[3:0]源电流能力控制
//BIT3~BIT2:PORTA[7:4]源电流能力控制,BIT1~BIT0:PORTA[3:0]源电流能力控制

PSRC1 = 0B11111111;    //PORTC,PORTD 源电流设置最大
//BIT7~BIT6:PORTD[7:4]源电流能力控制,BIT5~BIT4:PORTD[3:0]源电流能力控制
//BIT3~BIT2:PORTC[7:4]源电流能力控制,BIT1~BIT0:PORTC[3:0]源电流能力控制

PSINK0 = 0B11111111;    //PORTA 灌电流设置最大 0:最小, 1:最大
PSINK1 = 0B11111111;    //PORTB 灌电流设置最大 0:最小, 1:最大
PSINK2 = 0B11111111;    //PORTC 灌电流设置最大 0:最小, 1:最大
PSINK3 = 0B11111111;    //PORTD 灌电流设置最大 0:最小, 1:最大

ANSELA = 0B00000000;    //全为数字管脚

}
/*-----
* 函数名称: DelayUs
* 功能:    短延时函数 --16M-2T--大概快 1%左右.
* 输入参数: Time 延时时间长度 延时时长 Time Us
* 返回参数: 无
-----*/
void DelayUs(unsigned char Time)
{
    unsigned char a;
    for(a=0;a<Time;a++)
    {
        NOP();
    }
}
/*-----
* 函数名称: DelayMs
* 功能:    短延时函数
* 输入参数: Time 延时时间长度 延时时长 Time ms
* 返回参数: 无
-----*/
void DelayMs(unsigned char Time)
{
    unsigned char a,b;
    for(a=0;a<Time;a++)
    {
        for(b=0;b<5;b++)

```



```

        {
            DelayUs(197); //快 1%
        }
    }
}
/*-----
* 函数名称: DelayS
* 功能:    短延时函数
* 输入参数: Time 延时时间长度 延时时长 Time S
* 返回参数: 无
-----*/
void DelayS(unsigned char Time)
{
    unsigned char a,b;
    for(a=0;a<Time;a++)
    {
        for(b=0;b<10;b++)
        {
            DelayMs(100);
        }
    }
}
/*-----
* 函数名称: Time4Initial
* 功能:
* 输入参数:
* 返回参数: 无
-----*/
void Time4Initial(void)
{
    PCKEN |= 0B00001000;    //TIME4 模块时钟使能

    TIM4CR1 = 0B00000001;
    //BIT7: 0: TIM1_ARR 寄存器没有缓冲, 它可以被直接写入; 1: TIM1_ARR 寄存器由预装
    载缓冲器缓冲。
    //BIT6:保留
    //BIT5~BIT4:timer4 时钟选择位。
        //00: 系统时钟/主时钟
        //01: 内部快时钟 HIRC
        //10: LP 时钟, 只有当 FOSC 选择 LP 模式时才有意义
        //11: XT 时钟, 只有当 FOSC 选择 XT 模式时才有意义

    //BIT3:

```

```
//      0: 在发生更新事件时, 计数器不停止;
//      1: 在发生下一次更新事件(清除 CEN 位)时, 计数器停止。

//BIT2:
//      0: 如果 UDIS 允许产生更新事件, 则下述任一事件产生一个更新中断:
//          //寄存器被更新(计数器上溢/下溢)
//          //软件设置 UG 位
//          //时钟/触发控制器产生的更新
//      1: 如果 UDIS 允许产生更新事件, 则只有当下列事件发生时才产生更新中断,
并 UIF 置 1:
//          //寄存器被更新(计数器上溢/下溢)

//BIT1:
//      0: 一旦下列事件发生, 产生更新(UEV)事件:
//          //计数器溢出/下溢
//          //产生软件更新事件
//          //时钟/触发模式控制器产生的硬件复位被缓存的寄存器被装入它们的预装
载值。
//      1: 不产生更新事件, 影子寄存器(ARR、PSC、CCRx)保持它们的值。如果设置
了 UG 位或时钟/触发控制器发出了一个硬件复位, 则计数器和预分频器被重新初始化。

// BIT0: 0: 禁止计数器; 1: 使能计数器。

TIM4IER=0B00000001;
//BIT0: 0: 禁止更新中断; 1: 允许更新中断。

TIM4ISR=0B00000000;
//BIT0:当产生更新事件时该位由硬件置 1。它由软件写 1 清 0
//0: 无更新事件产生;
//1: 更新事件等待响应。当寄存器被更新时该位由硬件置 1:
//若 TIM4_CR1 寄存器的 UDIS=0, 当计数器上溢或下溢时;
//若 TIM4_CR1 寄存器的 UDIS=0、URS=0, 当设置 TIM4_EGR 寄存器的 UG 位
软件对计数器
//CNT 重新初始化时;
//若 TIM4_CR1 寄存器的 UDIS=0、URS=0, 当计数器 CNT 被触发事件重新初始
化时。

TIM4EGR =0B00000000;
//BIT0:该位由软件置 1, 由硬件自动清 0。
//0: 无动作;
//1: 重新初始化计数器, 并产生一个更新事件。注意预分频器的计数器也被清 0(但是预分
频系数不变)。若在中心对称模式下或 DIR=0(向上计数)则计数器被清 0; 若 DIR=1(向下计数)
```

则计数器取 TIM1_ARR 的值。

```
TIM4CNTR=0; //TIM4 8 位计数器
```

```
TIM4PSCR=0B00000100;
```

```
//预分频器对输入的 CK_PSC 时钟进行分频。
```

```
//计数器的时钟频率 fCK_CNT 等于 fCK_PSC/2(PSC[2:0])。PSC[7:3]由硬件清 0。
```

//PSCR 包含了当更新事件产生时装入当前预分频器寄存器的值(包括由于清除 TIMx_EGR 寄存器的 UG 位产生的计数器清除事件)。这意味着如要新的预分频值生效，必须产生更新事件或者 CEN=0。

```
TIM4ARR    =124;
```

```
//ARR 包含了将要装载入实际的自动重装载寄存器的值。
```

```
//当自动重装载的值为空时，计数器不工作。
```

```
INTCON |= 0B11000000;    //开总中断和外设中断
```

```
}
```

```
/*-----
```

```
* 函数名: Time1Initial
```

```
* 功能:   上电系统初始化
```

```
* 输入:   无
```

```
* 输出:   无
```

```
-----*/
```

```
void Time1_Capture_Initial(void)
```

```
{
```

```
PCKEN |=0B00000010;    //使能 timer1 时钟模块
```

```
CKOCON=0B00100000;
```

```
TCKSRC=0B00000011;    //TIM1 时钟为 HIRC 的 2 倍频
```

```
//BIT7 低频内振模式: 1 = 256K 振荡频率模式,0 = 32K 振荡频率模式
```

```
//BIT6~BIT4TIM2 时钟源选择位
```

```
    //值 时钟源
```

```
    //0  系统时钟/主时钟
```

```
    //1  HIRC
```

```
    //2  XT 时钟/外部时钟
```

```
    //3  HIRC 的 2 倍频
```

```
    //4  XT 时钟/外部时钟的 2 倍频
```

```
    //5  LIRC
```

```
    //6  LP 时钟/外部时钟
```

```
    //7  LP 时钟/外部时钟的 2 倍频
```

```
//BIT3:保留位
```

```
//BIT2~BIT1:TIM1 时钟源选择位
```

```
    //值 时钟源
```

```
    //0  系统时钟/主时钟
```

//1 HIRC
//2 XT 时钟/外部时钟
//3 HIRC 的 2 倍频
//4 XT 时钟/外部时钟的 2 倍频
//5 LIRC
//6 LP 时钟/外部时钟
//7 LP 时钟/外部时钟的 2 倍频

TIM1CR1 = 0B00000110; //预载允许, 边沿对齐向上计数器, 计数器使能

//BIT7:自动预装载允许位

//0: TIM1_ARR 寄存器没有缓冲, 它可以被直接写入;

//1: TIM1_ARR 寄存器由预装载缓冲器缓冲。

//BIT6~BIT5:选择对齐模式

//00: 边沿对齐模式。计数器依据方向位(DIR)向上或向下计数。

//01: 中央对齐模式 1。计数器交替地向上和向下计数。配置为输出的通道 (TIM1_CCMRx 寄存器中 CciS=00)的输出比较中断标志位, 只在计数器向下计数时被置 1。

//10:中央对齐模式 2。计数器交替地向上和向下计数。配置为输出的通道 (TIM1_CCMRx 寄存器中 CciS=00)的输出比较中断标志位, 只在计数器向上计数时被置 1。

//11: 中央对齐模式 3。计数器交替地向上和向下计数。配置为输出的通道 (TIM1_CCMRx 寄存器中 CciS=00)的输出比较中断标志位, 在计数器向上和向下计数时均被置 1。

//BIT4:方向

//0: 计数器向上计数;

//1: 计数器向下计数。

//BIT3:单脉冲模式

//0: 在发生更新事件时, 计数器不停止;

//1: 在发生下一次更新事件(清除 CEN 位)时, 计数器停止。

//BIT2:更新请求源

//0: 如果 UDIS 允许产生更新事件, 则下述任一事件产生一个更新中断:

//寄存器被更新(计数器上溢/下溢)

//软件设置 UG 位

//时钟/触发控制器产生的更新

//1: 如果 UDIS 允许产生更新事件, 则只有当下列事件发生时才产生更新中断, 并 UIF 置 1:

//寄存器被更新(计数器上溢/下溢)

//BIT1: 禁止更新

//0: 一旦下列事件发生, 产生更新(UEV)事件:

//计数器溢出/下溢

//产生软件更新事件

//时钟/触发模式控制器产生的硬件复位被缓存的寄存器被装入它们的预装载值。

//1: 不产生更新事件, 影子寄存器(ARR、PSC、CCRx)保持它们的值。如果设置了 UG 位或时钟/触发控制器发出了一个硬件复位, 则计数器和预分频器被重新初始化。

//BIT0 允许计数器
//0: 禁止计数器;
//1: 使能计数器。

TIM1CR2=0B01000000;

//BIT7: TI1 选择

//0: CC1 输入管脚连到 TI1(数字滤波器的输入);
//1: CC1、CC2 和 CC3 管脚经异或后连到 TI1。

//BIT6~BIT4: 主模式选择,该位用于选择在主模式下送到 ADC 或其它从定时器的同步信息(TRGO)。可能的组合如下:

//000: 复位 - TIM1_EGR 寄存器的 UG 位被用于作为触发输出(TRGO)。如果触发输入(时钟/触发控制器配置为复位模式)产生复位,则 TRGO 上的信号相对实际的复位会有一个延迟。

//001: 使能 - 计数器使能信号被用于作为触发输出(TRGO)。其用于启动多个定时器或 ADC,以便控制在一段时间内使能从定时器或 ADC。计数器使能信号是通过 CEN 控制位和门控模式下的触发输入信号的逻辑或产生。除非选择了主/从模式(见 TIM1_SMCR 寄存器中 MSM 位的描述),当计数器使能信号受控于触发输入时,TRGO 上会有一个延迟。

//010: 更新 - 更新事件被选为触发输入(TRGO)。

//011: 比较脉冲(MATCH1) - 一旦发生一次捕获或一次比较成功,当 CC1IF 标志被置 1 时(即使它已经为高),触发输出送出一个正脉冲(TRGO)。

//100: 比较 - OC1REF 信号被用于作为触发输出(TRGO)。

//101: 比较 - OC2REF 信号被用于作为触发输出(TRGO)。

//110: 比较 - OC3REF 信号被用于作为触发输出(TRGO)。

//111: 比较 - OC4REF 信号被用于作为触发输出(TRGO)。

//BIT3:D 保留

//BIT2: 捕获/比较控制位的更新控制选择

//0: 当捕获/比较的控制位为预装载时(CCPC=1),只有在 COMG 位置 1 的时候这些控制位才被更新;
1: 当捕获/比较的控制位为预装载时(CCPC=1),只有在 COMG 位置 1 或 TRGI 发生上升沿的时候这些控制位才被更新;

//注: 该位只对拥有互补输出的通道有效。

//BIT1:保留,被硬件设为 0。

//BIT0: 捕获/比较预装载控制位

//0: CCIE, CCINE, CCiP, CCiNP 位(TIM1_CCERx 寄存器)和 OCIM 位(TIM1_CCMRx 寄存器)

//不是预装载的;

//1: CCIE, CCINE, CCiP, CCiNP 和 OCIM 位是预装载的;设置该位后,它们只在设置了

//COMG 位(TIM1_EGR 寄存器)后被更新。

//注: 该位只对具有互补输出的通道起作用。

```
//TIM1SMCR =0B01110000;  
//TIM1SMCR|=0B00000101;  
TIM1SMCR =0B01010000;  
//TIM1SMCR|=0B00000110;  
TIM1SMCR|=0B00000100;
```

//BIT7: 主/从模式

//0: 无作用;

//1: 触发输入(TRGI)上的事件被延迟了, 以允许定时器 1 与它的从定时器间的完美同步(通过 TRGO)。

//BIT6~BIT4: 触发选择,这 3 位选择用于选择同步计数器的触发输入。

//000: 内部触发 ITR0 连接到 TIM6 TRGO (此设计没有 TIM6, 所以固定接 0)

//001: 保留

//010: 内部触发 ITR2 连接到 TIM5 TRGO(此设计没有 TIM5, 所以固定接 0)

//011: 保留

//100: TI1 的边沿检测器(TI1F_ED)

//101: 滤波后的定时器输入 1(TI1FP1)

//110: 滤波后的定时器输入 2(TI2FP2)

//111: 外部触发输入(ETRF)

//注: 这些位只能在未用到(如 SMS=000)时被改变, 以避免在改变时产生错误的边沿检测。

//BIT3:保留

//BIT2~BIT0: 时钟/触发/从模式选择,当选择了外部信号, 触发信号(TRGI)的有效边沿与选中的外部输入极性相关(见输入控制寄存器和控制寄存器的说明)

//000: 时钟/触发控制器禁止 - 如果 CEN=1, 则预分频器直接由内部时钟驱动。

//001: 编码器模式 1 - 根据 TI1FP1 的电平, 计数器在 TI2FP2 的边沿向上/下计数。

//010: 编码器模式 2 - 根据 TI2FP2 的电平, 计数器在 TI1FP1 的边沿向上/下计数。

//011: 编码器模式 3 - 根据另一个输入的电平, 计数器在 TI1FP1 和 TI2FP2 的边沿向上/下计数。

//100: 复位模式 - 在选中的触发输入(TRGI)的上升沿时重新初始化计数器, 并且产生一个更新寄存器的信号。

//101: 门控模式 - 当触发输入(TRGI)为高时, 计数器的时钟开启。一旦触发输入变为低, 则计数器停止(但不复位)。计数器的启动和停止都是受控的。

//110: 触发模式 - 计数器在触发输入 TRGI 的上升沿启动(但不复位), 只有计数器的启动是受控的。

//111: 外部时钟模式 1 - 选中的触发输入(TRGI)的上升沿驱动计数器。

//注: 如果 TI1F_ED 被选为触发输入(TS=100)时, 不要使用门控模式。这是因为 TI1F_ED 在每次

//TI1F 变化时只是输出一个脉冲, 然而门控模式是要检查触发输入的电平。

TIM1ETR = 0B00000000;

//BIT7: 外部触发极性,该位决定是 ETR 还是 ETR 用于触发操作。

//0: ETR 不反相,即高电平或上升沿有效;

//1: ETR 反相,即低电平或下降沿有效。

//BIT6: 外部时钟使能,该位用于使能外部时钟模式 2。

//0: 禁止外部时钟模式 2;

//1: 使能外部时钟模式 2,计数器的时钟为 ETRF 的有效沿。

//注 1: ECE 位置 1 的效果与选择把 TRGI 连接到 ETRF 的外部时钟模式 1 相同 (TIM1_SMCR 寄存器中, SMS=111, TS=111)。

//注 2: 外部时钟模式 2 可与下列模式同时使用: 触发标准模式; 触发复位模式; 触发门控模式。但是,此时 TRGI 决不能与 ETRF 相连(TIM1_SMCR 寄存器中,TS 不能为 111)。

//注 3: 外部时钟模式 1 与外部时钟模式 2 同时使能,外部时钟输入为 ETRF。

//BIT5~BIT4: 外部触发预分频器,外部触发信号 EPRP 的频率最大不能超过 fMASTER/4。可用预分频器来降低 ETRP 的频率,当

//EPRP 的频率很高时,它非常有用:

//00: 预分频器关闭;

//01: EPRP 的频率/2;

//02: EPRP 的频率/4;

//03: EPRP 的频率/8。

//BIT3~BIT0: 外部触发滤波器选择,该位域定义了 ETRP 的采样频率及数字滤波器长度。数字滤波器由一个事件计数器组成,它记录到 N 个事件后会产生一个输出的跳变:

//0000: 无滤波器,以 fMASTER 采样 1000 : 采样频率
fSAMPLING=fMASTER/8, N=6

//0001: 采样频率 fSAMPLING=fMASTER, N=2 1001 : 采样频率
fSAMPLING=fMASTER/8, N=8

//0010: 采样频率 fSAMPLING=fMASTER, N=4 1010 : 采样频率
fSAMPLING=fMASTER/16, N=5

//0011: 采样频率 fSAMPLING=fMASTER, N=8 1011 : 采样频率
fSAMPLING=fMASTER/16, N=6

//0100: 采样频率 fSAMPLING=fMASTER/2, N=6 1100 : 采样频率
fSAMPLING=fMASTER/16, N=8

//0101: 采样频率 fSAMPLING=fMASTER/2, N=8 1101 : 采样频率
fSAMPLING=fMASTER/32, N=5

//0110: 采样频率 fSAMPLING=fMASTER/4, N=6 1110 : 采样频率
fSAMPLING=fMASTER/32, N=6

//0111: 采样频率 fSAMPLING=fMASTER/4, N=8 1111 : 采样频率
fSAMPLING=fMASTER/32, N=8

TIM1IER = 0B00000010;//捕获比较 1 中断使能

//BIT7: 允许刹车中断

//0: 禁止刹车中断;

//1: 允许刹车中断。

//BIT6: 触发中断使能

//0: 禁止触发中断;
//1: 使能触发中断。
//BIT5: 允许 COM 中断
//0: 禁止 COM 中断;
//1: 允许 COM 中断。
//BIT4: 允许捕获/比较 4 中断
//0: 禁止捕获/比较 4 中断;
//1: 允许捕获/比较 4 中断。
//BIT3: 允许捕获/比较 3 中断
//0: 禁止捕获/比较 3 中断;
//1: 允许捕获/比较 3 中断。
//BIT2: 允许捕获/比较 2 中断
//0: 禁止捕获/比较 2 中断;
//1: 允许捕获/比较 2 中断。
//BIT1: 允许捕获/比较 1 中断
//0: 禁止捕获/比较 1 中断;
//1: 允许捕获/比较 1 中断。
//BIT0: 允许更新中断
//0: 禁止更新中断;
//1: 允许更新中断。

TIM1SR1 = 0B00000000; //T1CC1IF

//BIT7: 刹车中断标记,一旦刹车输入有效,由硬件对该位置 1。如果刹车输入无效,则该位可由软件清 0。

//0: 无刹车事件产生;
//1: 刹车输入上检测到有效电平。

//BIT6: 触发器中断标记,当发生触发事件(当从模式控制器处于除门控模式外的其它模式时,在 TRGI 输入端检测到有效边沿,或门控模式下的任一边沿)时由硬件对该位置 1。它由软件清 0。

//0: 无触发器事件产生;
//1: 触发中断等待响应。

//BIT5: COM 中断标记,一旦产生 COM 事件(当捕获/比较控制位: CciE、CciNE、OciM 已被更新)该位由硬件置 1。它由软件清 0。

//0: 无 COM 事件产生;
//1: COM 中断等待响应。

//BIT4: 捕获/比较 4 中断标记
//参考 CC1IF 描述。

//BIT3: 捕获/比较 3 中断标记
//参考 CC1IF 描述。

//BIT2: 捕获/比较 2 中断标记
//参考 CC1IF 描述。

//BIT1: 捕获/比较 1 中断标记 如果通道 CC1 配置为输出模式: 当计数器值与比较值匹配时该位由硬件置 1,但在中心对称模式下除外(参考 TIM1_CR1 寄存器的 CMS 位)。它由

软件清 0。

//0: 无匹配发生;

//1: TIMx_CNT 的值与 TIMx_CCR1 的值匹配。

//注: 在中心对称模式下, 当计数器值为 0 时, 向上计数, 当计数器值为 ARR 时, 向下计数 (它从 0 向上计数到 ARR-1, 再由 ARR 向下计数到 1)。因此, 对所有的 SMS 位值, 这两个值都不置标记。但是, 如果 CCR1>ARR, 则当 CNT 达到 ARR 值时, CC1IF 置 1。

//如果通道 CC1 配置为输入模式: 当捕获事件发生时该位由硬件置 1, 它由软件清 0 或通过读 TIM1_CCR1L 清 0。

//0: 无输入捕获产生;

//1: 计数器值已被捕获(拷贝)至 TIM1_CCR1(在 IC1 上检测到与所选极性相同的边沿)。

//BIT0: 更新中断标记, 当产生更新事件时该位由硬件置 1。它由软件清 0。

//0: 无更新事件产生;

//1: 更新事件等待响应。当寄存器被更新时该位由硬件置 1:

//若 TIM1_CR1 寄存器的 UDIS=0, 当计数器上溢或下溢时;

//若 TIM1_CR1 寄存器的 UDIS=0、URS=0, 当设置 TIM1_EGR 寄存器的 UG 位软件对计数器

//CNT 重新初始化时;

//若 TIM1_CR1 寄存器的 UDIS=0、URS=0, 当计数器 CNT 被触发事件重新初始化时 (参考 0

//从模式控制寄存器 TIM1_SMCR)。

TIM1SR2 = 0B00000000;

TIM1EGR = 0B00000000;

//BIT7: 产生刹车事件, 该位由软件置 1, 用于产生一个刹车事件, 由硬件自动清 0。

//0: 无动作;

//1: 产生一个刹车事件。此时 MOE=0、BIF=1, 若开启对应的中断(BIE=1), 则产生相应的中断。

//BIT6: 产生触发事件, 该位由软件置 1, 用于产生一个触发事件, 由硬件自动清 0。

//0: 无动作;

//1: TIM1_SR 寄存器的 TIF=1, 若开启对应的中断 (TIE=1), 则产生相应的中断。

//BIT5: 捕获/比较事件, 产生控制更新该位由软件置 1, 由硬件自动清 0。

//0: 无动作;

//1: 当 CCPC=1, 允许更新 CCIE、CCINE、CciP, CciNP, OCIM 位。

//注: 该位只对拥有互补输出的通道有效。

//BIT4: 产生捕获/比较 4 事件

//参考 CC1G 描述。

//BIT3: 产生捕获/比较 3 事件

//参考 CC1G 描述。

//BIT2: 产生捕获/比较 2 事件

//参考 CC1G 描述。

//BIT1: 产生捕获/比较 1 事件

//该位由软件置 1, 用于产生一个捕获/比较事件, 由硬件自动清 0。

//0: 无动作;

//1: 在通道 CC1 上产生一个捕获/比较事件: 若通道 CC1 配置为输出:

//设置 CC1IF=1, 若开启对应的中断, 则产生相应的中断。若通道 CC1 配置为输入:

//当前的计数器值被捕获至 TIM1_CCR1 寄存器, 设置 CC1IF=1, 若开启对应的中断, 则产生相应的中断。若 CC1IF 已经为 1, 则设置 CC1OF=1。

//BIT0: 产生更新事件

//该位由软件置 1, 由硬件自动清 0。

//0: 无动作;

//1: 重新初始化计数器, 并产生一个更新事件。注意预分频器的计数器也被清 0(但是预分频系数不变)。若在中心对称模式下或 DIR=0(向上计数)则计数器被清 0; 若 DIR=1(向下计数)则计数器取 TIM1_ARR 的值。

TIM1CCMR1 = 0B00000001; //CC1 通道被配置为输入

//BIT7: 输出比较 1 清零使能

//该位用于使能使用 TIM1_TRIG 引脚上的外部事件来清通道 1 的输出信号 (OC1REF), 参考 17.5.9 在外部事件发生时清除 OC1REF 信号

//0: OC1REF 不受 ETRF 输入 (来自 TIM1_TRIG 引脚) 的影响;

//1: 一旦检测到 ETRF 输入高电平, OC1REF=0。

//BIG6~BIT4: 输出比较 1 模式, 该 3 位定义了输出参考信号 OC1REF 的动作, 而 OC1REF 决定了 OC1 的值。OC1REF 是高电平有效, 而 OC1 的有效电平取决于 CC1P 位。

//000: 冻结。输出比较寄存器 TIM1_CCR1 与计数器 TIM1_CNT 间的比较对 OC1REF 不起作用;

//001: 匹配时设置通道 1 的输出为有效电平。当计数器 TIM1_CNT 的值与捕获/比较寄存器 1(TIM1_CCR1)相同时, 强制 OC1REF 为高。

//010: 匹配时设置通道 1 的输出为无效电平。当计数器 TIM1_CNT 的值与捕获/比较寄存器 1(TIM1_CCR1)相同时, 强制 OC1REF 为低。

//011: 翻转。当 TIM1_CCR1=TIM1_CNT 时, 翻转 OC1REF 的电平。

//100: 强制为无效电平。强制 OC1REF 为低。

//101: 强制为有效电平。强制 OC1REF 为高。

//110: PWM 模式 1— 在向上计数时, 一旦 TIM1_CNT < TIM1_CCR1 时通道 1 为有效电平, 否则为无效电平; 在向下计数时, 一旦 TIM1_CNT > TIM1_CCR1 时通道 1 为无效电平(OC1REF=0), 否则为有效电平(OC1REF=1)。

//111: PWM 模式 2— 在向上计数时, 一旦 TIM1_CNT < TIM1_CCR1 时通道 1 为无效电平, 否则为有效电平; 在向下计数时, 一旦 TIM1_CNT > TIM1_CCR1 时通道 1 为有效电平, 否则为无效电平。

//注 1: 一旦 LOCK 级别设为 3(TIM1_BKR 寄存器中的 LOCK 位)并且 CC1S=00(该通道配置成输出) 则该位不能被修改。

//注 2: 在 PWM 模式 1 或 PWM 模式 2 中, 只有当比较结果改变了或在输出比较模式中从冻结模式切换到 PWM 模式时, OC1REF 电平才改变。(参考 17.5.7 PWM 模式)

//注 3: 在有互补输出的通道上, 这些位是预装载的。如果 TIM1_CR2 寄存器的 CCPC=1, OCM 位只有在 COM 事件发生时, 才从预装载位取新值。

//BIT3: 输出比较 1 预装载使能

//0: 禁止 TIM1_CCR1 寄存器的预装载功能, 可随时写入 TIM1_CCR1 寄存器, 并且新写入的数值立即起作用。

//1: 开启 TIM1_CCR1 寄存器的预装载功能, 读写操作仅对预装载寄存器操作, TIM1_CCR1 的预装载值在更新事件到来时被加载至当前寄存器中。

//注 1: 一旦 LOCK 级别设为 3(TIM1_BKR 寄存器中的 LOCK 位)并且 CC1S=00(该通道配置成输出) 则该位不能被修改。

//注 2: 为了操作正确, 在 PWM 模式下必须使能预装载功能。但在单脉冲模式下(TIM1_CR1 寄存器的 OPM=1), 它不是必须的。

//BIT2: 输出比较 1 快速使能, 该位用于加快 CC 输出对触发输入事件的响应。

//0: 根据计数器与 CCR1 的值, CC1 正常操作, 即使触发器是打开的。当触发器的输入有一个有效沿时, 激活 CC1 输出的最小延时为 5 个时钟周期。

//1: 输入到触发器的有效沿的作用就象发生了一次比较匹配。因此, OC 被设置为比较电平而与比较结果无关。采样触发器的有效沿和 CC1 输出间的延时被缩短为 3 个时钟周期。

//OCFE 只在通道被配置成 PWM1 或 PWM2 模式时起作用。

//BIT1~BIT0: 捕获/比较 1 选择。这 2 位定义通道的方向(输入/输出), 及输入脚的选择:

//00: CC1 通道被配置为输出;

//01: CC1 通道被配置为输入, IC1 映射在 TI1FP1 上;

//10: CC1 通道被配置为输入, IC1 映射在 TI2FP1 上;

//11: CC1 通道被配置为输入, IC1 映射在 TRC 上。此模式仅工作在内部触发器输入被选中时(由

//TIM1_SMCR 寄存器的 TS 位选择)。

//注: CC1S 仅在通道关闭时(TIM1_CCER1 寄存器的 CC1E=0)才是可写的。

//TIM1CCMR2 = 0B00000000;

//TIM1CCMR3 = 0B00000000;

//TIM1CCMR4 = 0B00000000;

TIM1CCER1 = 0B00000001; //上升沿触发, 触发使能

//BIT7: 输入捕获/比较 2 互补输出极性。参考 CC1NP 的描述。

//BIT6: 输入捕获/比较 2 互补输出使能。参考 CC1NE 的描述。

//BIT5: 输入捕获/比较 2 输出极性。参考 CC1P 的描述。

//BIT4: 输入捕获/比较 2 输出使能。参考 CC1E 的描述。

//BIT3: 输入捕获/比较 1 互补输出极性

//0: OC1N 高电平有效;

//1: OC1N 低电平有效。

//注 1: 一旦 LOCK 级别(TIM1_BKR 寄存器中的 LCKK 位)设为 3 或 2 且 CC1S=00(通道配置为输出) 则该位不能被修改。

//注 2: 对于有互补输出的通道, 该位是预装载的。如果 CCPC=1 (TIM1_CR2 寄存器), 只有在

//COM 事件发生时, CC1NP 位才从预装载位中取新值。

//BIT2: 输入捕获/比较 1 互补输出使能

//0: 关闭— OC1N 禁止输出, 因此 OC1N 的输出电平依赖于 MOE、OSSI、OSSR、OIS1、

//OIS1N 和 CC1E 位的值。

//1: 开启— OC1N 信号输出到对应的输出引脚, 其输出电平依赖于 MOE、OSSI、OSSR、

//OIS1、OIS1N 和 CC1E 位的值。

//注: 对于有互补输出的通道, 该位是预装载的。如果 CCPC=1(TIM1_CR2 寄存器), 只有在

//COM 事件发生时, CC1NE 位才从预装载位中取新值。

//BIT1: 输入捕获/比较 1 输出极性 CC1 通道配置为输出:

//0: OC1 高电平有效;

//1: OC1 低电平有效。

//CC1 通道配置为触发(参考图 61):

//0: 触发发生在 TI1F 的高电平或上升沿;

//1: 触发发生在 TI1F 的低电平或下降沿。

//CC1 通道配置为输入(参考图 61):

//0: 捕捉发生在 TI1F 的高电平或上升沿;

//1: 捕捉发生在 TI1F 的低电平或下降沿。

//注 1: 一旦 LOCK 级别(TIM1_BKR 寄存器中的 LCCK 位)设为 3 或 2, 则该位不能被修改。

//注 2: 对于有互补输出的通道, 该位是预装载的。如果 CCPC=1 (TIM1_CR2 寄存器), 只有在

//COM 事件发生时, CC1P 位才从预装载位中取新值。

//BIT0: 输入捕获/比较 1 输出使能

//CC1 通道配置为输出:

//0: 关闭— OC1 禁止输出, 因此 OC1 的输出电平依赖于 MOE、OSSI、OSSR、OIS1、OIS1N 和 CC1NE 位的值。

//1: 开启— OC1 信号输出到对应的输出引脚, 其输出电平依赖于 MOE、OSSI、OSSR、OIS1、OIS1N 和 CC1NE 位的值。
CC1 通道配置为输入:

//该位决定了计数器的值是否能捕获入 TIM1_CCR1 寄存器。

//0: 捕获禁止;

//0: 捕获使能。

//注: 对于有互补输出的通道, 该位是预装载的。如果 CCPC=1(TIM1_CR2 寄存器), 只有在

//COM 事件发生时, CC1E 位才从预装载位中取新值。

TIM1CCER2 = 0B00000000;

```
TIM1CNTRH =0B00000000;//TIM1 计数器
TIM1CNTRL =0B00000000;
```

```
TIM1PSCRH =0B00000000;//计数时钟不分频
TIM1PSCRL =0B00000000;
```

```
//TIM1ARRH =0;
```

```
//TIM1ARRL =0;
```

```
//TIM1ARRH =0x03;          //自动重载, 周期
```

```
//TIM1ARRL =0xe8;
```

```
//TIM1RCR =0B00001111;    //重复计数器的值
```

```
//TIM1CCR1H =0x01;         //PWM 脉宽
```

```
//TIM1CCR1L =0xf4;
```

```
//TIM1CCR2H =0B00000000;
```

```
//TIM1CCR2L =0B00000000;
```

```
//TIM1CCR3H =0B00000000;
```

```
//TIM1CCR3L =0B00000000;
```

```
//TIM1CCR4H =0B00000000;
```

```
//TIM1CCR4L =0B00000000;
```

```
//TIM1BKR =0B11000000;    //输出使能, 禁止刹车
```

```
//TIM1DTR =0B00000111;    //死区发生器
```

//BIT7~BIT0: 死区发生器设置,这些位定义了插入互补输出之间的死区持续时间。假设 DT 表示其持续时间, tCK_PSC 为 TIM1 的时钟脉冲:

//DTG[7:5]=0xx => DT=DTG[7:0]x tdtg, 其中: tdtg=tCK_PSC. (f1)

//DTG[7:5]=10x => DT=(64+DTG[5:0])x tdtg, 其中: tdtg= tCK_PSC. (f2)

//DTG[7:5]=110 => DT=(32+DTG[4:0])x tdtg, 其中: tdtg=8x tCK_PSC. (f3)

//DTG[7:5]=111 => DT=(32+DTG[4:0])x tdtg, 其中: tdtg=16x tCK_PSC. (f4)

```
//TIM1OISR =0B00000000; //空闲输出状态设置
```

```
//BIT1: 输出空闲状态 1(OC1N 输出)。
```

//0: 当 MOE=0 时, 则在一个死区时间后, OC1N=0;

//1: 当 MOE=0 时, 则在一个死区时间后, OC1N=1。

//注: 已经设置了 LOCK(TIM1_BKR 寄存器)级别 1、2 或 3 后, 该位不能被修改。

```
//BIT0: 输出空闲状态 1(OC1 输出)。
```

//0: 当 MOE=0 时, 如果 OC1N 使能, 则在一个死区后, OC1=0;

//1: 当 MOE=0 时, 如果 OC1N 使能, 则在一个死区后, OC1=1。

//注：已经设置了 LOCK(TIM1_BKR 寄存器)级别 1、2 或 3 后，该位不能被修改。

```
//LEBCON =0B00000000; //前沿消隐禁止
//BIT7  前沿消隐使能位（仅当 ADGO=0 时可进行切换，否则 ADC 工作异常）
//1 = 使能
//0 = 禁止

//BIT6~BIT5: 前沿消隐通道选择
//00 = TIM1_CH1
//01 = TIM1_CH2
//10 = TIM1_CH3
//11 = TIM1_CH4

//BIT4:N/A  保留位，读 0
//BIT3:  PWM 消隐沿选择
//0 = PWM 上升沿
//1 = PWM 下降沿

//BIT2~BIT0:TIM1 的故障源使能，高有效
//BKS2: 选择 ADC 阈值比较
//BKS1: 选择 LVD 检测
//BKS0: 选择 BKIN 管脚
```

T1CEN=1;

```
}
/*-----
 * 函数名: main
 * 功能:   主函数
 * 输入:   无
 * 输出:   无
-----*/
void main(void)
{
    POWER_INITIAL();           //系统初始化
    Time1_Capture_Initial();
    Time4Initial();

    while(1)
    {
        NOP();
    }
}
```

Fremont Micro Devices (SZ) Limited

#5-8, 10/F, Changhong Building, Ke-Ji Nan 12 Road, Nanshan District, Shenzhen, Guangdong 518057

Tel: (86 755) 86117811

Fax: (86 755) 86117810

Fremont Micro Devices (Hong Kong) Limited

#16, 16/F, Blk B, Veristrong Industrial Centre, 34-36 Au Pui Wan Street, Fotan, Shatin, Hong Kong

Tel: (852) 27811186

Fax: (852) 27811144

Fremont Micro Devices (USA), Inc.

42982 Osgood Road Fremont, CA 94539

Tel: (1-510) 668-1321

Fax: (1-510) 226-9918

Web Site: <http://www.fremontmicro.com/>

* Information furnished is believed to be accurate and reliable. However, Fremont Micro Devices, Incorporated (BVI) assumes no responsibility for the consequences of use of such information or for any infringement of patents of other rights of third parties which may result from its use. No license is granted by implication or otherwise under any patent rights of Fremont Micro Devices, Incorporated (BVI).

Specifications mentioned in this publication are subject to change without notice. This publication supersedes and replaces all information previously supplied. Fremont Micro Devices, Incorporated (BVI) products are not authorized for use as critical components in life support devices or systems without express written approval of Fremont Micro Devices, Incorporated (BVI). The FMD logo is a registered trademark of Fremont Micro Devices, Incorporated (BVI). All other names are the property of their respective own.