

## 32 位微控制器

## HC32L15 系列的复合定时器

### 适用对象

系列	产品型号
HC32L15	HC32L150KATA
	HC32L150JATA
	HC32L150FAUA
	HC32L156KATA
	HC32L156JATA

## 目 录

<b>1</b>	<b>摘要 .....</b>	<b>3</b>
<b>2</b>	<b>复合定时器简介 .....</b>	<b>3</b>
<b>3</b>	<b>HC32L15 系列的复合定时器模块.....</b>	<b>5</b>
	3.1 简介.....	5
	3.2 说明.....	5
	3.2.1 复合定时器 I/O 选择功能介绍.....	5
	3.2.2 定时器功能介绍.....	7
	3.2.3 寄存器介绍.....	15
	3.2.4 工作流程介绍.....	16
<b>4</b>	<b>样例代码 .....</b>	<b>20</b>
	4.1 代码介绍.....	20
	4.2 代码运行.....	23
<b>5</b>	<b>总结 .....</b>	<b>25</b>
<b>6</b>	<b>版本信息 &amp; 联系方式 .....</b>	<b>26</b>

## 1 摘要

本篇应用笔记主要介绍 HC32L15 系列的复合定时器工作模式，并简要说明通过复合定时器如何产生 PWM 方波。

## 2 复合定时器简介

### 什么是复合定时器？

该定时器有如下复合五种工作模式（如下），通过设置定时器控制寄存器的 FUNC2, 1, 0 位，选择定时器功能。可根据不同需求，配置复合定时器，产生 PWM/PPG 方波驱动 LED 或电机；捕获输入信号，测量外部输入脉冲宽度和周期。

FUNC2, FUNC1, FUNC0 位的设定	功能
0b000	复位模式
0b001	16 位 PWM 定时器
0b010	16 位 PPG 定时器
0b011	16/32 位重载定时器
0b100	16/32 位 PWC 定时器

### 工作模式说明

- 复位模式

复位模式是指复合定时器的初始复位后的状态 (各寄存器都是初始值)。

- 16 位 PWM（Pulse-Width Modulation）定时器

16 位 PWM 定时器从触发启动时周期设定的值开始递减计数。当前计数值与占空比设定寄存器的值匹配或下溢时，输出取反。

- 16 位 PPG（Program Pulse Generator）定时器

16 位 PPG 定时器从触发启动时“L”脉宽设定重载寄存器的值开始递减计数，当前计数值下溢时，输出电平取反跳变；继续从“H”脉宽设定重载寄存器的值开始递减计数，再发生下溢时，输出电平取反跳变。

- 16/32 位重载定时器

16 位重载定时器从触发启动时周期设定的值开始递减计数。16 位递减计数器发生下溢时，输出电平取反跳变。

- 16/32 位 PWC（Pulse Width Capture）定时器

PWC 定时器通过设定输入信号的测量起始沿启动 16/32 位递增计数器，和测量结束沿停止计数器。这期间的计数值以脉宽形式保存到数据缓冲寄存器。

## 3 HC32L15 系列的复合定时器模块

### 3.1 简介

华大 HC32L15 系列单片机支持 8 路通道复合定时器，主要包含两个功能：复合定时器 I/O 选择功能和定时器功能。

### 3.2 说明

#### 3.2.1 复合定时器 I/O 选择功能介绍

复合定时器 I/O 选择功能通过设定 I/O 模式决定复合定时器的信号(外部时钟/外部启动触发/波形)的输出或输入方法。每 2 路通道可以选择以下 9 种 I/O 模式之一，具有多路通道软件同时启动的功能，最大支持 8 路通道软件同时启动。

1) I/O 模式 0: 16 位定时器标准模式

复合定时器在每路通道单独工作。

2) I/O 模式 1: 定时器全模式

复合定时器偶数通道的信号单独分配到外部引脚的工作模式。

3) I/O 模式 2: 外部触发共享模式

本模式可同时输入外部启动触发信号到复合定时器的 2 个通道。使用此模式，可同时启动复合定时器的该 2 路通道。

4) I/O 模式 3: 其他通道触发共享模式

将另一个通道的外部信号用作外部启动触发进行启动的模式。通道 0 和通道 1 不可设定为此模式。

5) I/O 模式 4: 定时器启动/停止模式

使用偶数通道控制奇数通道的启动/停止的模式。奇数通道在偶数通道的输出信号的上升沿启动、下降沿停止。

6) I/O 模式 5: 软件同步启动模式

使用软件同时启动多个通道的模式。

7) I/O 模式 6: 软件启动定时器启动/停止模式

本模式使用偶数通道控制奇数通道的启动/停止。使用软件启动偶数通道。奇数通道在偶数通道的输出信号的上升沿启动，下降沿停止。

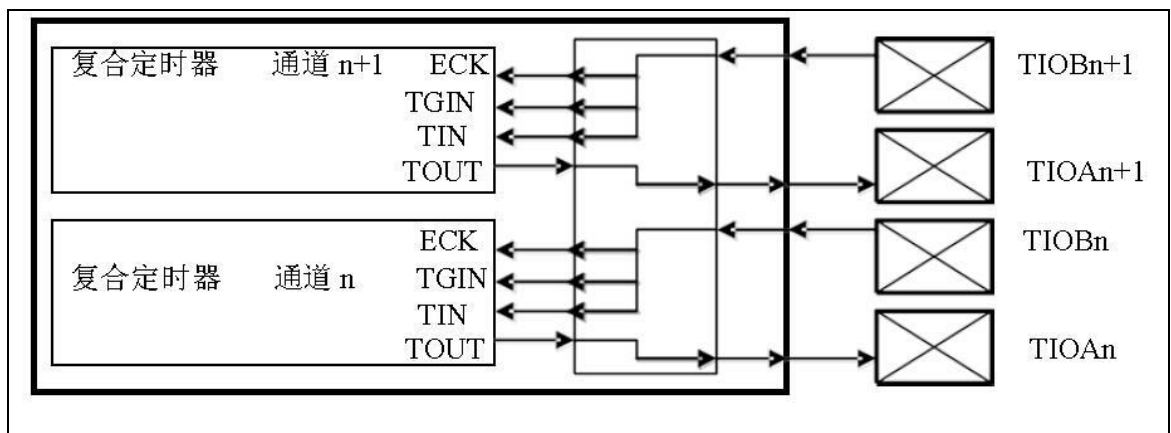
#### 8) I/O 模式 7：定时器启动模式

使用偶数通道控制奇数通道的启动的模式。奇数通道在偶数通道的输出信号的上升沿启动。

#### 9) I/O 模式 8：其他通道触发共享定时器启动/停止模式

将一个通道的外部信号用作另一个通道的外部启动触发进行启动的模式。通道 0 和 通道 1 不可设定为此模式。

根据不同的 IO 引脚配置，可以有 9 种 IO 模式，此处只介绍 IO Mode 0，如下图所示，其它模式请参见“HC32L15 用户手册”。



本模式所用的外部引脚的连接对象和输入输出信号如下表所示：

外部引脚	I/O	连接对象(内部信号)	输入输出信号
TIOA	输出	TOUT	输出复合定时器的波形
TIOB	输入	ECK/TGIN/TIN	输入的信号作为以下之一使用 • 外部时钟(ECK信号) • 外部启动触发(TGIN信号) • 测定波形(TIN 信号)

## 3.2.2 定时器功能介绍

复合定时器（Composite Timer, CT）可以被配置为 4 种模式：PWM 定时器、PPG 定时器、重载定时器、PWC 定时器。下面分别介绍这 4 种功能模式。

### 3.2.2.1 PWM 定时器

设定 CT 工作模式为 PWM 定时器功能，PWM 只能被配置为 16 位模式。CTn 和 CTn+1 可以分别输出两路 PWM 波形。PWM 功能定时器，TIOA 和 TIOB 引脚所使用的功能如下表所示：

CT 功能引脚	方向	内部信号	功能描述
TIOA	输出	TOUT	输出 PWM 的波形
TIOB	输入	ECK/TGIN	使用下列功能之一 <ul style="list-style-type: none"> <li>• 外部时钟输入（ECK）</li> <li>• 外部触发输入（TGIN）</li> </ul>

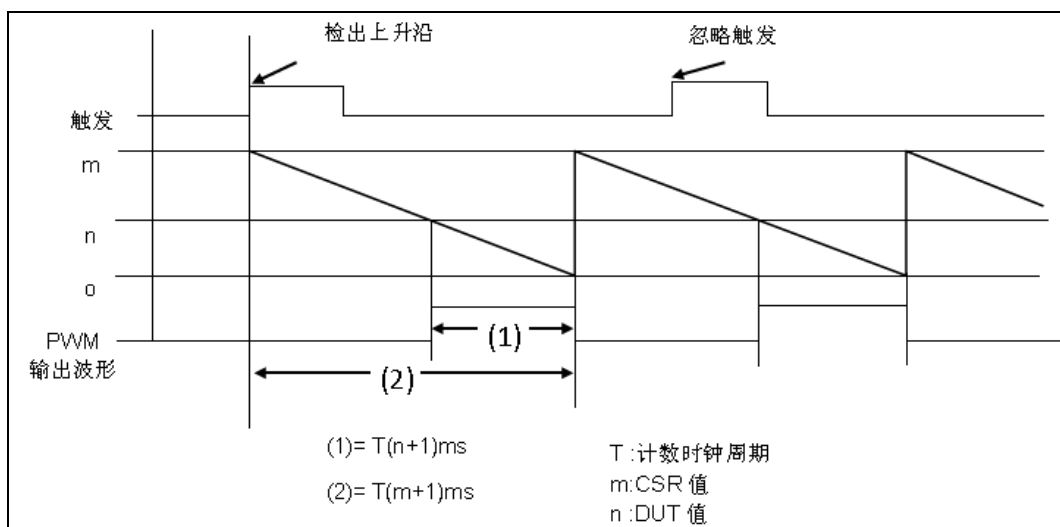
PWM 定时器有连续操作和单次操作两种工作模式。下面分别介绍这两种模式。

#### 连续操作

当 PWM 启动后，PWM 周期设定寄存器值加载到 PWM 定时器寄存器，然后递减计数。计数值和 PWM 占空比设定寄存器值相匹配时，产生占空比匹配中断并且 PWM 输出波形翻转。

当 PWM 计数值为 0 时，下一个计数时钟，PWM 计数值溢出，发生计数值溢出中断，然后 PWM 定时器寄存器被重新加载。

下图为禁止重启情况，所以在 PWM 启动后，PWM 的触发被忽视，如果使能重启的情况，在检测到 PWM 启动后，PWM 会重新启动，它的计数值将被重新加载。

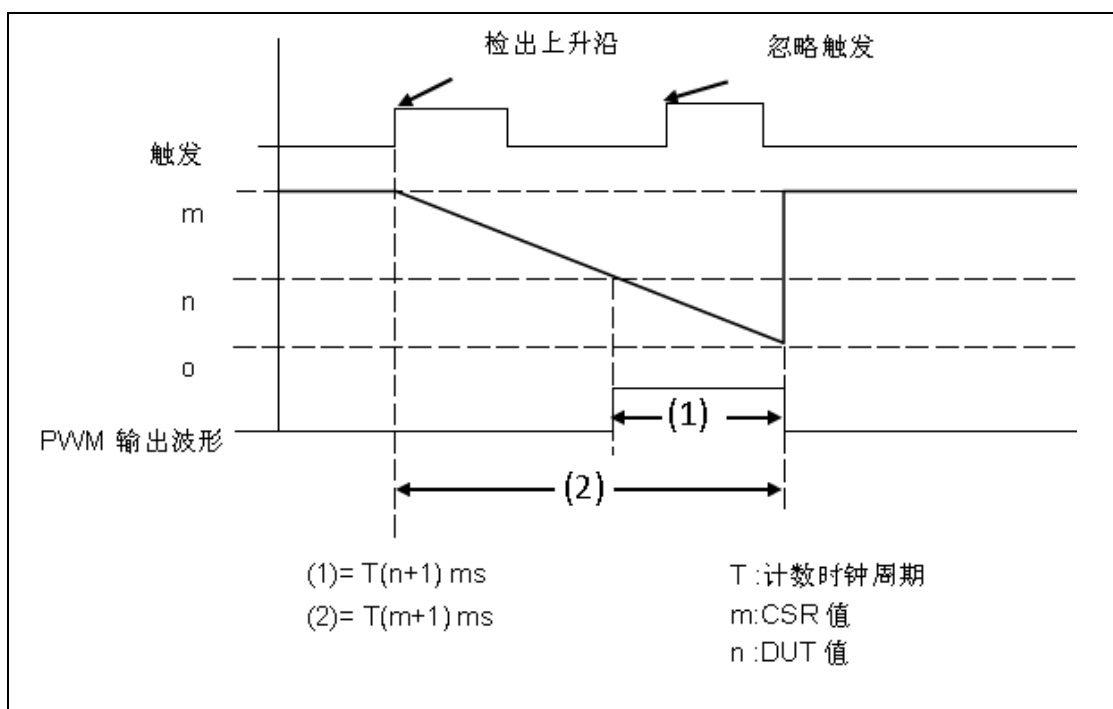


## 单次操作

当 PWM 启动后，PWM 周期设定寄存器值加载到 PWM 定时器寄存器，然后递减计数。计数值和 PWM 占空比设定寄存器值相匹配时，产生占空比匹配中断并且 PWM 输出波形翻转。

当 PWM 计数值为 0 时，下一个计数时钟，PWM 计数值溢出，发生计数值溢出中断，然后 PWM 定时器停止计数，PWM 波形停止输出。

下图为禁止重启情况，所以在 PWM 启动后，PWM 的触发被忽视，如果使能重启的情况，在检测到 PWM 启动后，PWM 会重新启动，它的计数值将被重新加载。





### 3.2.2.2 PPG 定时器

设定 CT 工作模式为 PPG 定时器功能，PPG 只能被配置为 16 位模式。CTn 和 CTn+1 可以分别输出两路 PPG 波形。PPG 功能定时器，TIOA 和 TIOB 引脚所使用的功能如下表所示：

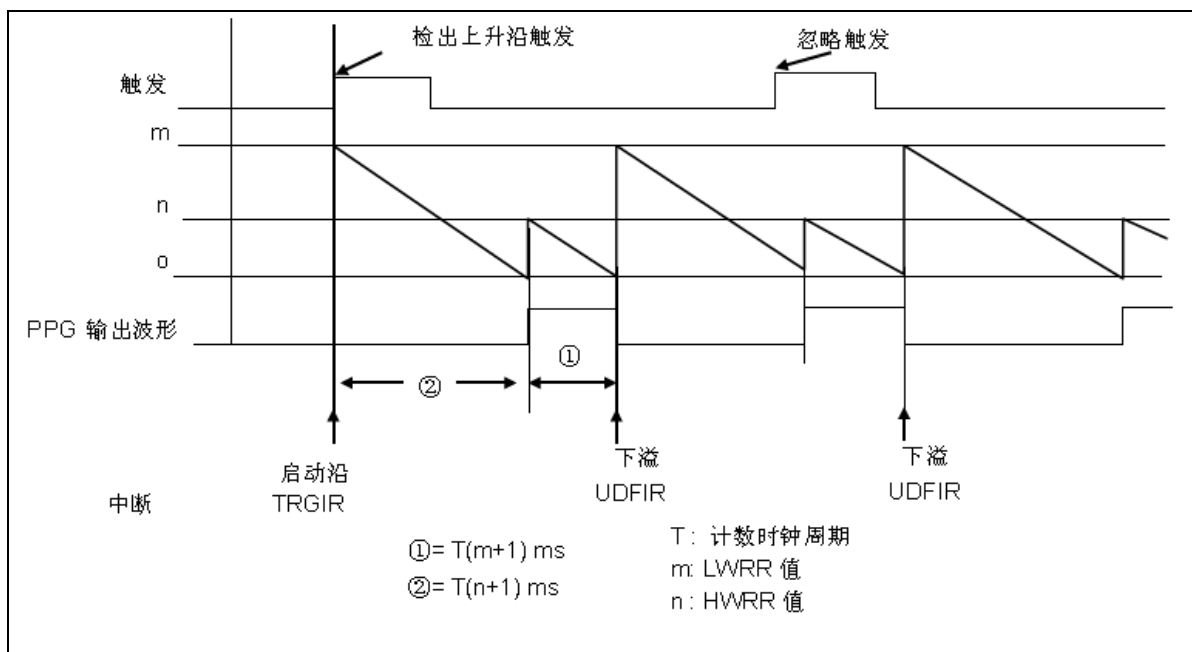
CT 功能引脚	方向	内部信号	功能描述
TIOA	输出	TOUT	输出 PWM 的波形
TIOB	输入	ECK/TGIN	使用下列功能之一 <ul style="list-style-type: none"> <li>• 外部时钟输入（ECK）</li> <li>• 外部触发输入（TGIN）</li> </ul>

PPG 定时器有连续操作和单次操作两种工作模式。下面分别介绍这两种模式。

#### 连续操作

当 PPG 启动后，PPG "L" 脉宽设定重载寄存器值加载到 PPG 定时器寄存器，然后递减计数，当计数器发生溢出时，PPG 输出波形发生翻转并且 PPG "H" 脉宽设定重载寄存器值加载到 PPG 定时器寄存器，然后递减计数，当计数器发生溢出时，循环上述过程。当 PPG 计数值溢出时，会产生 PPG 溢出中断。

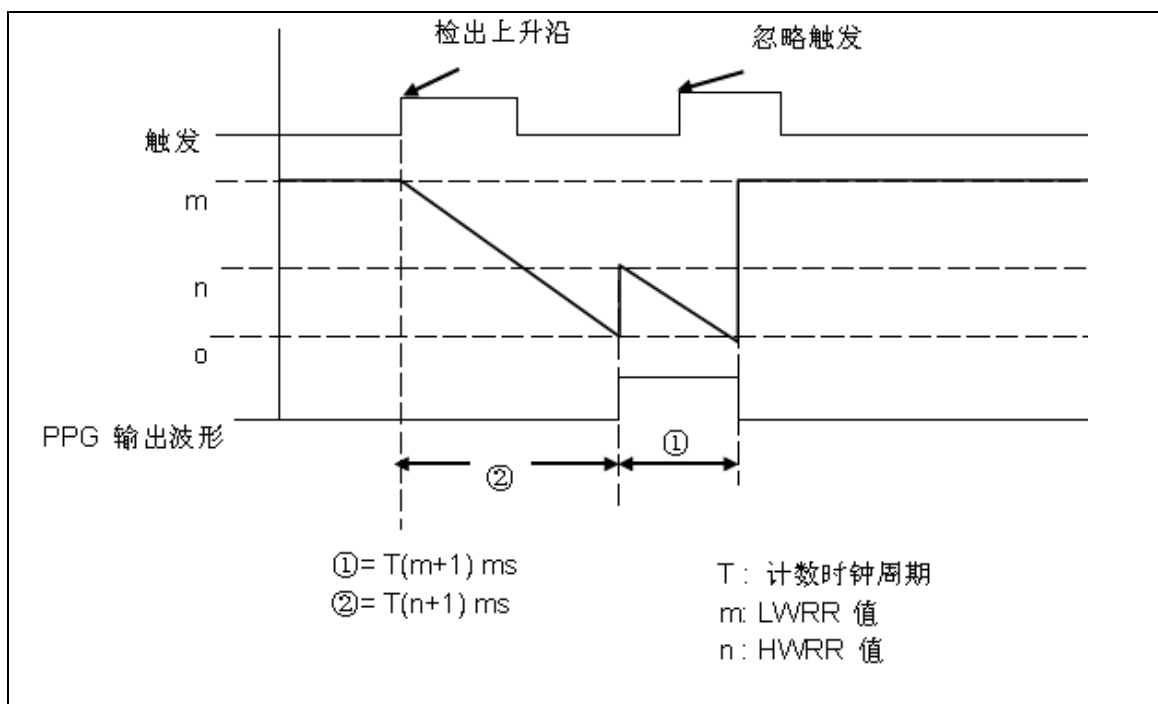
下图为禁止重启情况，所以在 PPG 启动后，PPG 的触发被忽视，如果使能重启的情况，在检测到 PPG 触发后，PPG 会重新启动，它的计数值将被重新加载。



## 单次操作

当 PPG 启动后，PPG "L"脉宽设定重载寄存器值加载到 PPG 定时器寄存器，然后递减计数，当计数器发生溢出时，PPG 输出波形发生翻转并且 PPG "H"脉宽设定重载寄存器值加载到 PPG 定时器寄存器，然后递减计数，当计数器发生溢出时，PPG 停止，PPG 输出波形也停止。当 PPG 计数值溢出时，会产生 PPG 溢出中断。

下图为禁止重启情况，所以在 PPG 启动后，PPG 的触发被忽视，如果使能重启的情况，在检测到 PPG 触发后，PPG 会重新启动，它的计数值将被重新加载。



### 3.2.2.3 重载定时器

设定 CT 工作模式为重载定时器功能，重载定时器可配置为 16 位或 32 位模式。为 32 位模式时，偶数通道的周期寄存器存放周期值低 16 位，奇数通道的周期寄存器存放周期值高 16 位，重载定时器的配置（启动、关闭、中断设置等）是由对偶数通道的操作来决定的。

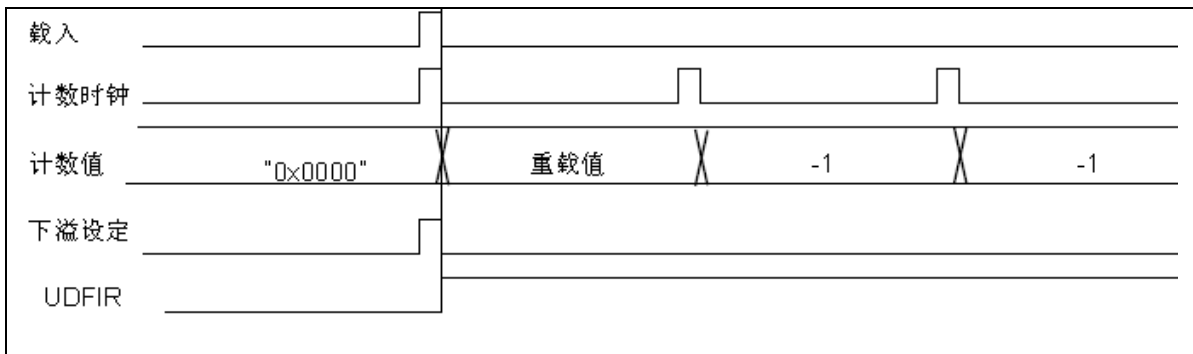
重载定时器功能，TIOA 和 TIOB 引脚所使用的功能如下表所示：

CT 功能引脚	方向	内部信号	功能描述
TIOA	输出	TOUT	输出 PWM 的波形
TIOB	输入	ECK/TGIN	使用下列功能之一 <ul style="list-style-type: none"> <li>• 外部时钟输入（ECK）</li> <li>• 外部触发输入（TGIN）</li> </ul>

重载定时器有连续操作和单次操作两种工作模式。下面分别介绍这两种模式。

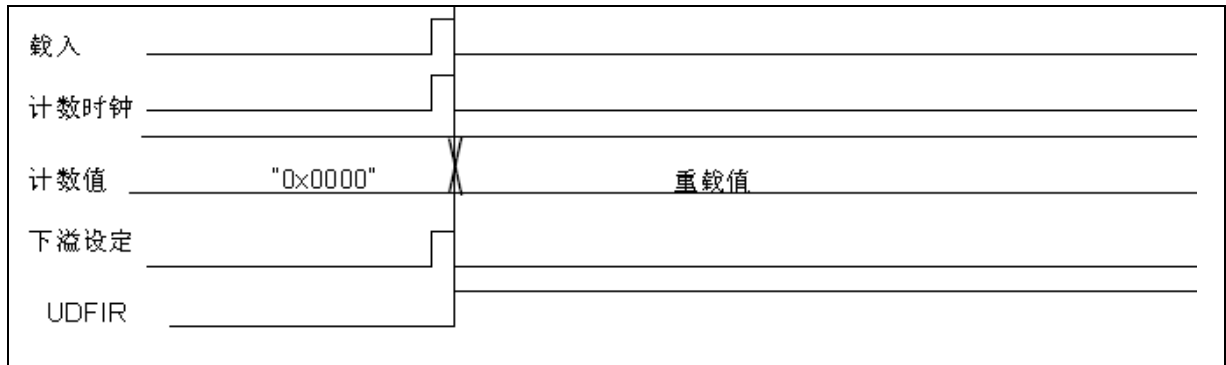
#### 连续操作

当重载定时器启动后，周期寄存器值自动加载到定时器寄存器里，然后递减计数，当计数值减为 0 时，在下个时钟信号时，重载计数器会再次把周期寄存器值加载定时器寄存器，然后计数器重新开始计数，如下图所示：



## 单次操作

当重载定时器启动后，周期寄存器值自动加载到定时器寄存器里，然后递减计数，当计数值减为 0，在下个时钟信号时，重载计数器停止，输出波形也停止，如下图所示：



### 3.2.2.4 PWC 定时器

设定 CT 工作模式为 PWC 定时器，可以用来测量输入波形周期、脉宽，可设置多种检测模式，如检测下降沿到下降沿，下降沿到上升沿等。

PWC 定时器可设置为 16 位或者 32 位模式，当为 32 位模式时，偶数通道的计数寄存器存放周期值低 16 位，奇数通道的计数寄存器存放周期值高 16 位，PWC 定时器的配置（启动、关闭、中断设置等）是由对偶数通道的操作来决定的。

当使用 PWC timer 时，TIOA 和 TIOB 引脚所使用的功能如下表所示：

设定 CT 工作模式为 PPG 定时器功能，PPG 只能被配置为 16 位模式。CTn 和 CTn+1 可以分别输出两路 PPG 波形。PPG 功能定时器，TIOA 和 TIOB 引脚所使用的功能如下表所示：

CT 功能引脚	方向	内部信号	功能描述
TIOA	输出	TOUT	输出 PWM 的波形
TIOB	输入	TIN	只能使用下列功能 •测定波形(TIN 信号)

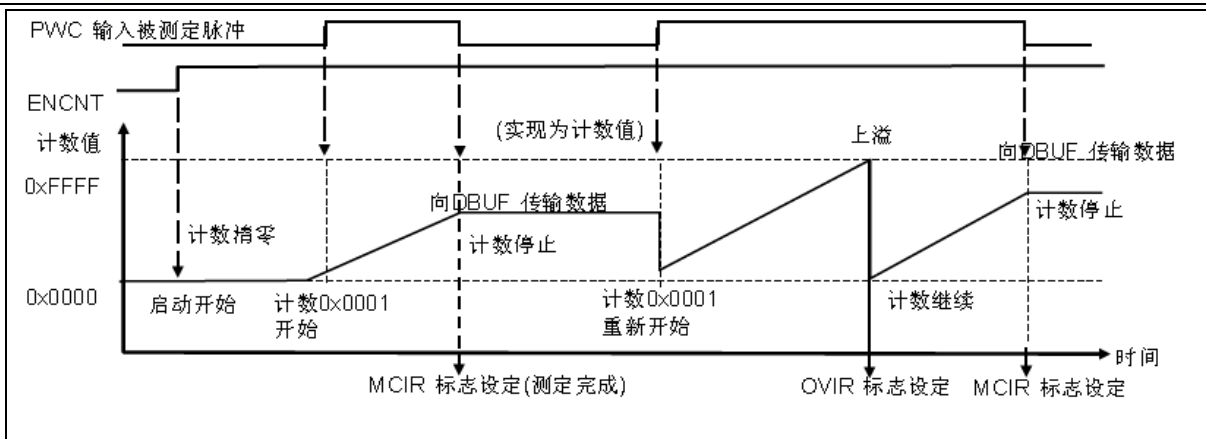
PWC 定时器有连续操作和单次操作两种工作模式。下面分别介绍这两种模式。

#### 连续操作

下图例子采用的是上升沿到下降沿检测模式，当 PWC 检测到第一次上升沿时，PWC 计数值开始加 1，当 PWC 检测到第一次下降沿时，PWC 会把计数值加载到数据缓冲寄存器，并且产生 PWC 测量结束中断。到下一次上升沿被检测到后，PWC 重新开始计数，然后计数值又从 0 开始计数，等到下次下降沿到来时，再把计数值加载到数据缓冲寄存器。

如果前面一次数据缓冲寄存器值未读取，DBUF 寄存器值又被更新，前一次的测定结果会丢失。PWC 的错误标志位会置 1。

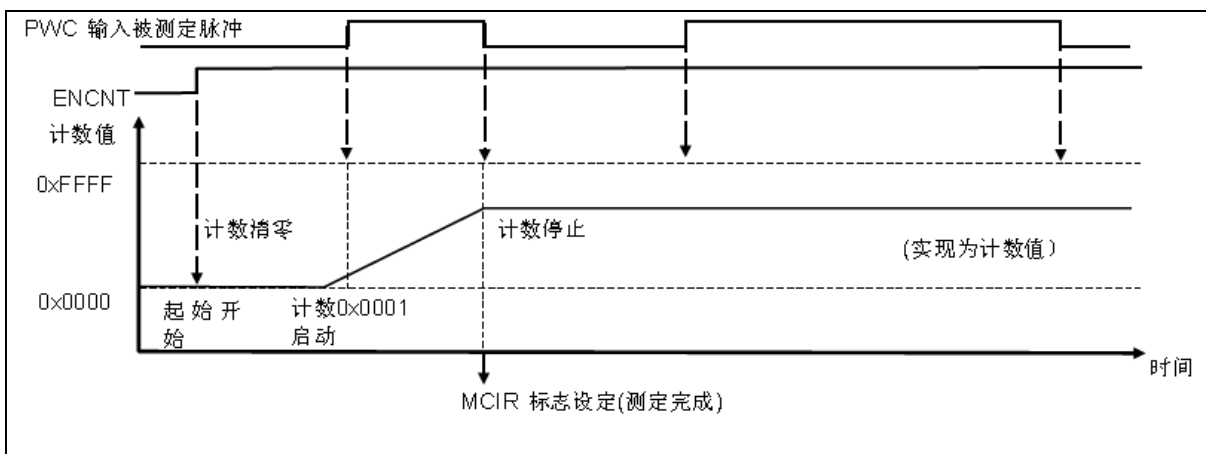
如果在过程中计数值发生溢出，那么 PWC 的上溢中断请求位会置 1，计数值会重新从 0 开始计数。如果 PWC 的上溢中断使能，将会产生上溢中断。



## 单次操作

下图例子采用的是上升沿到下降沿检测模式，当 PWC 检测到第一次上升沿时，PWC 计数值开始加 1，当 PWC 检测到第一次下降沿时，PWC 会把计数值加载到数据缓冲寄存器，并且产生 PWC 测量结束中断。再次输入上升沿时，PWC 不启动计数。

如果在过程中计数值发生溢出，那么 PWC 的上溢中断请求位会置 1，计数值会重新从 0 开始计数。如果 PWC 的上溢中断使能，将会产生上溢中断。



### 3.2.3 寄存器介绍

CT 模块的寄存器如下表所示，若需了解具体细节，请参考 HC32L15 用户手册。

#### 1) 复合定时器 I/O 选择功能的寄存器

寄存器简称	寄存器名称
CTSEL0123	I/O 选择寄存器
CTSEL4567	I/O 选择寄存器
CTSSSR	同时软件启动寄存器

#### 2) PWM 定时器的寄存器

寄存器简称	寄存器名称
TCR	定时器控制寄存器
TMCR2	定时器控制寄存器 2
STA	状态控制寄存器
CSR	PWM 周期设定寄存器
DUT	PWM 占空比设定寄存器
TCNT	定时器寄存器

#### 3) PPG 定时器的寄存器

寄存器简称	寄存器名称
TCR	定时器控制寄存器
TMCR2	定时器控制寄存器 2
STA	状态控制寄存器
LWRR	"L"脉宽设定重载寄存器
HWRR	"H"脉宽设定重载寄存器
TCNT	定时器寄存器

#### 4) 重载定时器的寄存器

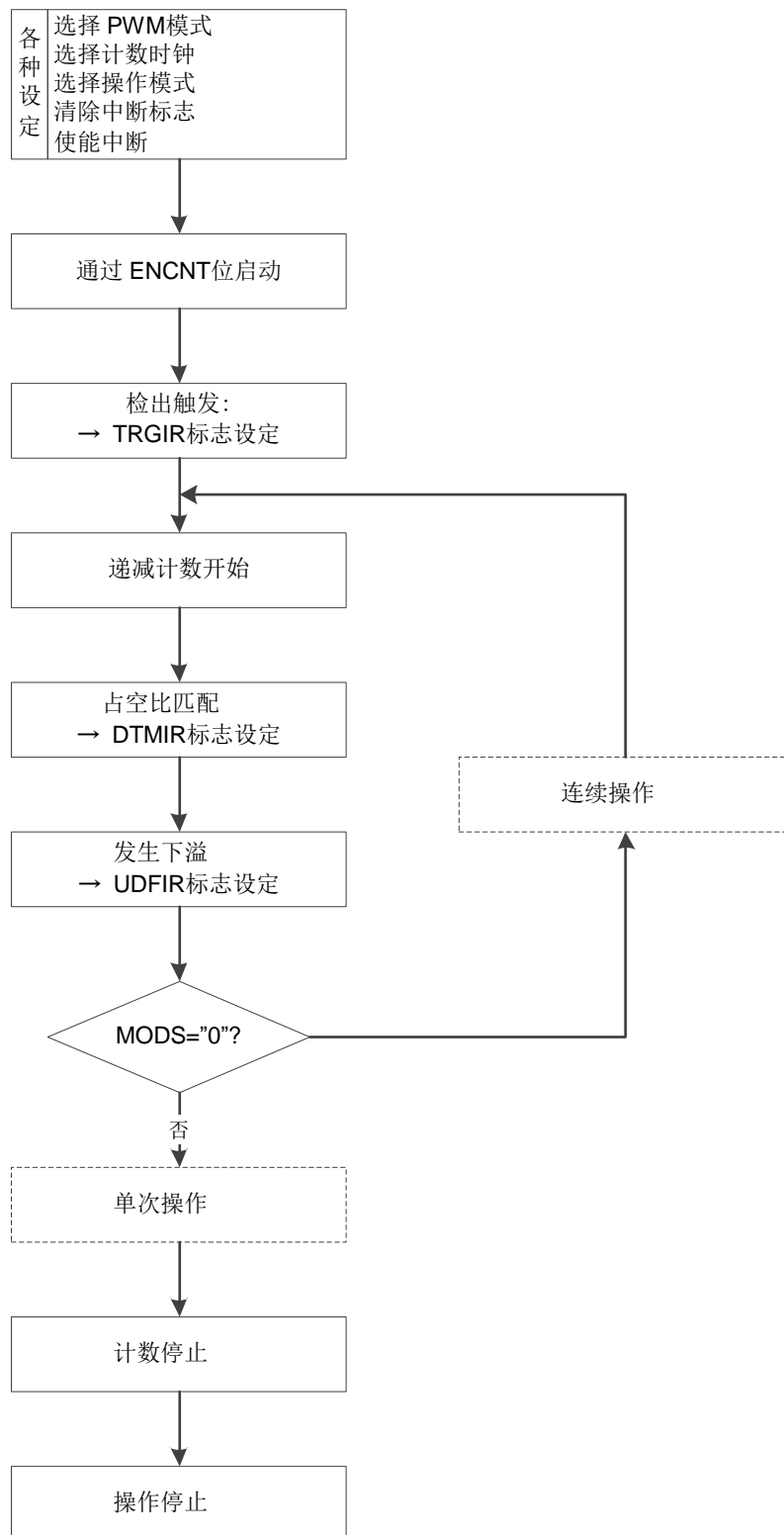
寄存器简称	寄存器名称
TCR	定时器控制寄存器
TMCR2	定时器控制寄存器 2
STA	状态控制寄存器
CSR	周期设定寄存器
TCNT	定时器寄存器

#### 5) PWC 定时器的寄存器

寄存器简称	寄存器名称
TCR	定时器控制寄存器
TMCR2	定时器控制寄存器 2
STA	状态控制寄存器
DBUF	数据缓冲寄存器

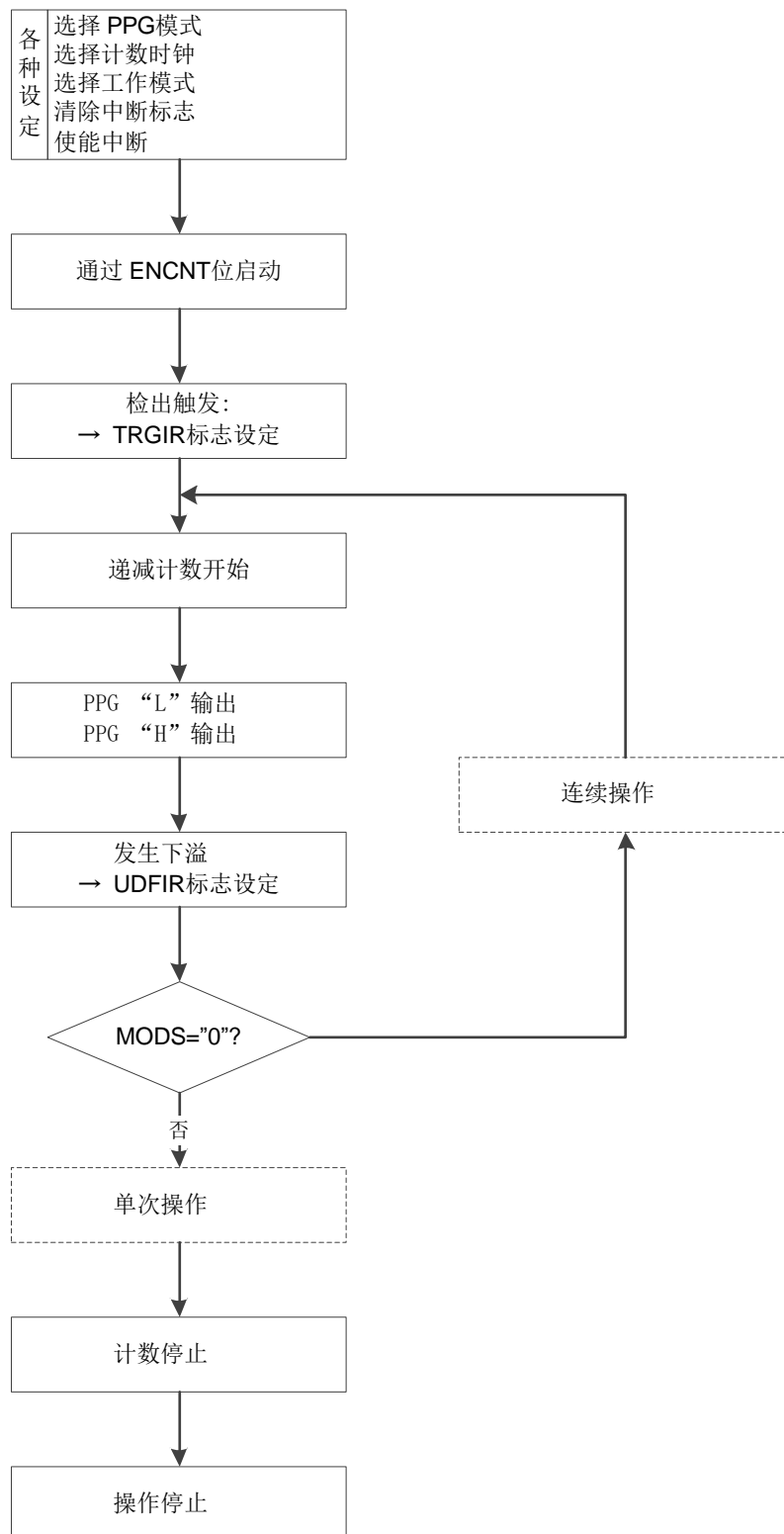
### 3.2.4 工作流程介绍

PWM 定时器操作流程，如下图所示：

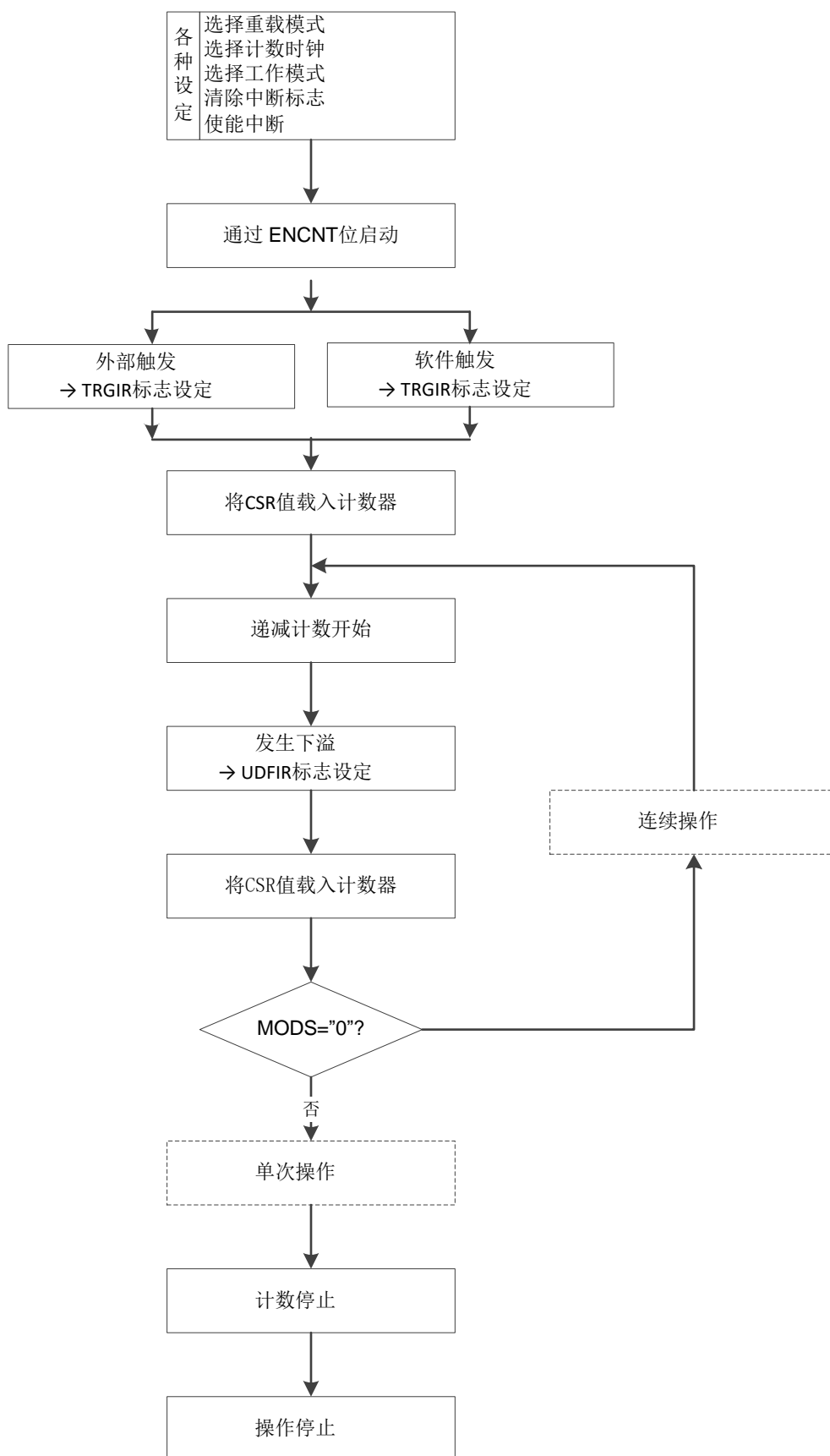




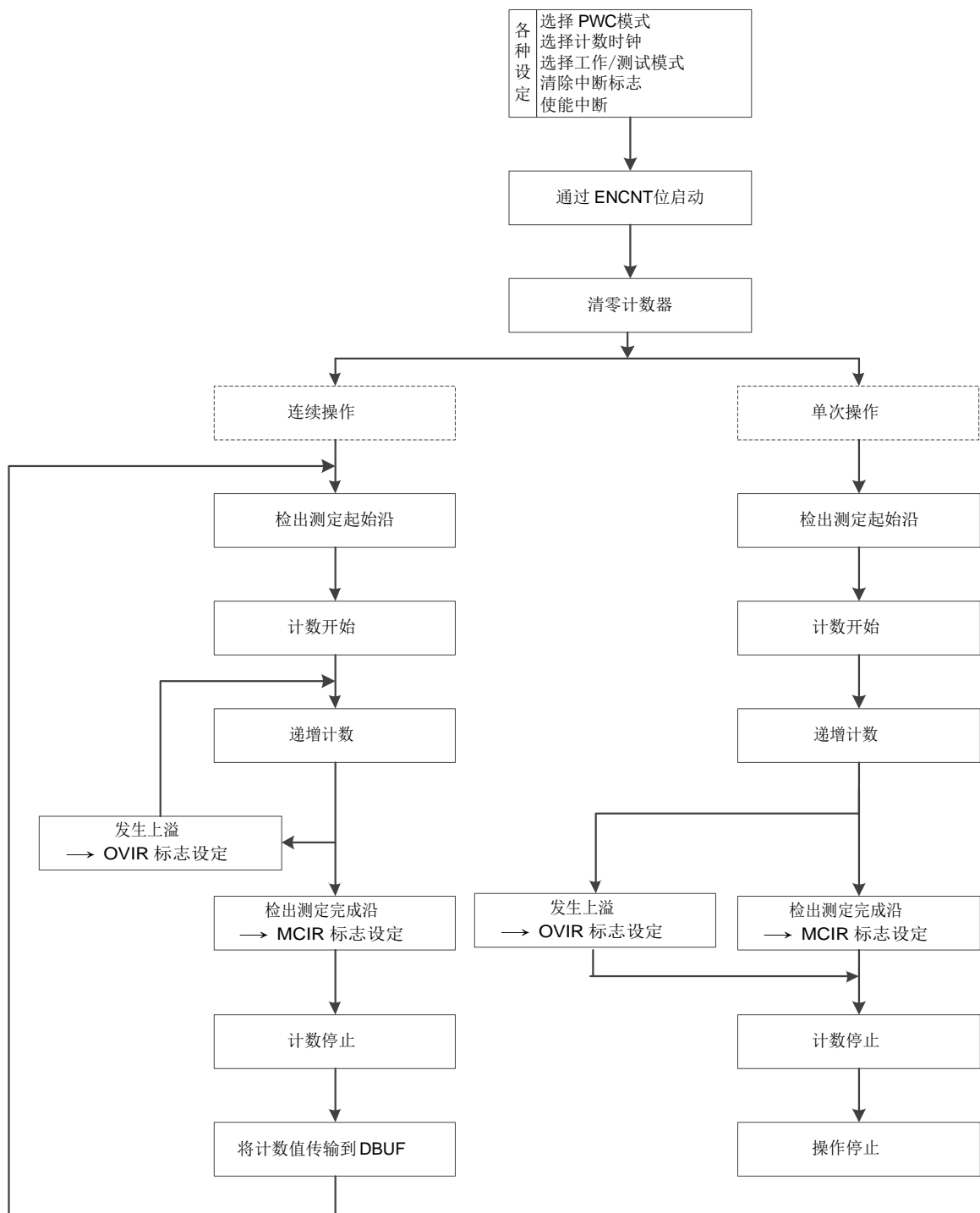
PPG 定时器操作流程，如下图所示：



重载定时器操作流程，如下图所示：



PWC 定时器操作流程，如下图所示：



## 4 样例代码

### 4.1 代码介绍

用户可以根据上述的工作流程编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 HC32L15 系列 MCU 的设备驱动库（Device Driver Library, DDL）来体验复合定时器多种功能模式。

以下部分主要基于 DDL 的外设 CT 通道 0 简要介绍 PWM 定时器使用方法：

- 1) 设置复合定时器 I/O 引脚。

```
/* Set Composite Timer IO port */
/* P14/INT12_2/SEG5/SIN6_1/TIOB2_1/TIOA0_0 */
Gpio_SetFunc_TIOA0_0(0u);
```

- 2) 设置复合定时器 I/O 模式。

```
/* Set Composite Timer I/O mode */
Ct_ConfigIOMode(CT_CH, CT_IO_MODE);
```

- 3) 初始化 PWM 定时器：

```
/* Initialize Pointer to interrupt request structure */
stcPwmConfig.pstcPwmIrqEn = &stcIrqEn;
stcPwmConfig.pstcPwmIrqCb = &stcIrqCb;
/* Initialize PWM timer */
stcPwmConfig.enPres = PwmPresNone; /* PWM clock = 5MHz @ PCLK = 20MHz */
stcPwmConfig.enMode = CT_PWM_MODE;
stcPwmConfig.enExtTrig = PwmExtTrigDisable;
stcPwmConfig.enOutputMask = PwmOutputNormal;
stcPwmConfig.enOutputPolarity = PwmPolarityLow;
stcPwmConfig.enRestartEn = PwmRestartEnable;
stcPwmConfig.pstcPwmIrqEn->bPwmTrigIrq = 1;
stcPwmConfig.pstcPwmIrqEn->bPwmDutyMatchIrq = 1;
stcPwmConfig.pstcPwmIrqEn->bPwmUnderflowIrq = 1;
stcPwmConfig.pstcPwmIrqCb->pfnPwmTrigIrqCb = PwmTrigIrqHandler;
stcPwmConfig.pstcPwmIrqCb->pfnPwmDutyMatchIrqCb = PwmDutyMatchIrqHandler;
stcPwmConfig.pstcPwmIrqCb->pfnPwmUnderflowIrqCb = PwmUnderflowIrqHandler;
stcPwmConfig.bTouchNvic = TRUE;
Ct_Pwm_Init(CT_CH, &stcPwmConfig);
```

- 4) 设置 PWM 定时器周期和占空值：

```
/* Set cycle and duty value */
Ct_Pwm_WriteCycleVal(CT_CH, CYCLE_VAL); /* m为CSR值，Cycle=(1+m)*PWM clock */
Ct_Pwm_WriteDutyVal(CT_CH, DUTY_VAL); /* n为DUT值，Duty=(1+m)*PWM clock */;
```

5) 使能计数:

```
Ct_Pwm_EnableCount(CT_CH); /* Enable count operatoin */
```

### 6) 触发 PWM 定时器:

```
Ct_Pwm_EnableSwTrig(CT_CH);/* Start triggered by software */
```

7) 中断方式计数: 触发中断请求、占空比匹配中断请求和下溢中断请求位, PWM 中断处理函数

```

/**
*****
** \brief PWM trigger interrupt handler
*****
*/
static void PwmTrigIrqHandler(void)
{
    m_u32CntIntTrg++;
}

/**
*****
** \brief PWM underflow interrupt handler
*****
*/
static void PwmUnderflowIrqHandler(void)
{
    m_u32CntIntUnder++;
}

/**
*****
** \brief PWM duty match interrupt handler
*****
*/
static void PwmDutyMatchIrqHandler(void)
{
    m_u32CntIntDuty++;
}

```

8) 查询方式计数：触发中断请求、占空比匹配中断请求和下溢中断请求位

```
do
{
    if (Ct_Pwm_GetIrqFlag(CT_CH, PwmDutyMatchIrq) == TRUE)
    {
        Ct_Pwm_ClrIrqFlag(CT_CH, PwmDutyMatchIrq);
        m_u32CntIntDuty++;
    }
    else if (Ct_Pwm_GetIrqFlag(CT_CH, PwmUnderflowIrq) == TRUE)
    {
        Ct_Pwm_ClrIrqFlag(CT_CH, PwmUnderflowIrq);
        m_u32CntIntUnder++;
    }
    else if (Ct_Pwm_GetIrqFlag(CT_CH, PwmTrigIrq) == TRUE)
    {
        Ct_Pwm_ClrIrqFlag(CT_CH, PwmTrigIrq);
        m_u32CntIntTrg++;
    }
} while(m_u32CntIntUnder < 2500UL);
```

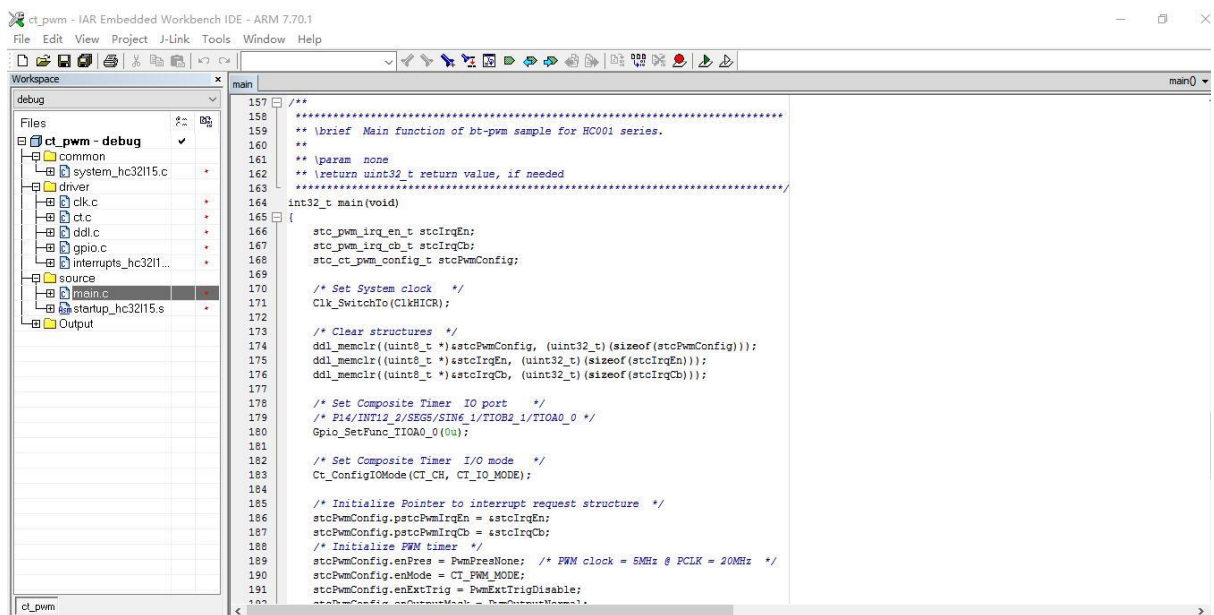
## 4.2 代码运行


用户可以通过华大半导体的网站下载到 HC32L15 的 DDL 的样例代码 (ct\_pwm)，并配合学习板 (比如‘EV-HC32L15-64L-MCU’) 运行相关代码学习使用 CT。


以下部分主要介绍如何在‘EV-HC32L15-64L-MCU’学习板上运行 ct\_pwm 样例代码并观察结果：

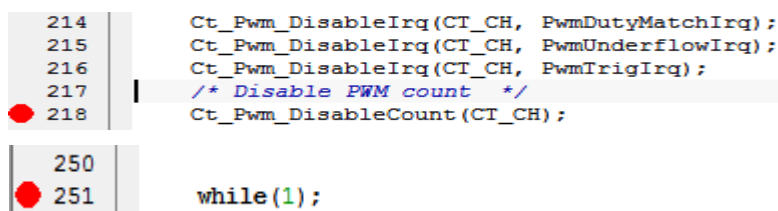
- 确认安装正确的 IAR EWARM v7.7 工具 (请从 IAR 官方网站下载相应的安装包，并参考用户手册进行安装)。
- 获取‘EV-HC32L15-64L-MCU’学习板。
- 从华大半导体网站下载 HC32L15 DDL 代码。
- 下载并运行 ct\_pwm\中的项目文件：

1) 打开 ct\_pwm\项目，并打开‘main.c’如下视图：



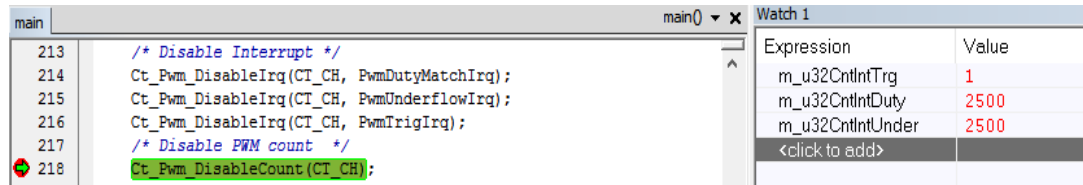
2) TIOA0\_0 引脚和示波器相连，点击  重新编译整个项目。

3) 点击  将代码下载到学习板上，第 218、251 行设置断点，全速运行。



- 4) 停在断点 218 行时，查看 m\_u32CntIntUnder、m\_u32CntIntDuty、m\_u32CntIntTrg，观察示器波形。

m\_u32CntIntUnder、m\_u32CntIntDuty、m\_u32CntIntTrg 如下图所示：



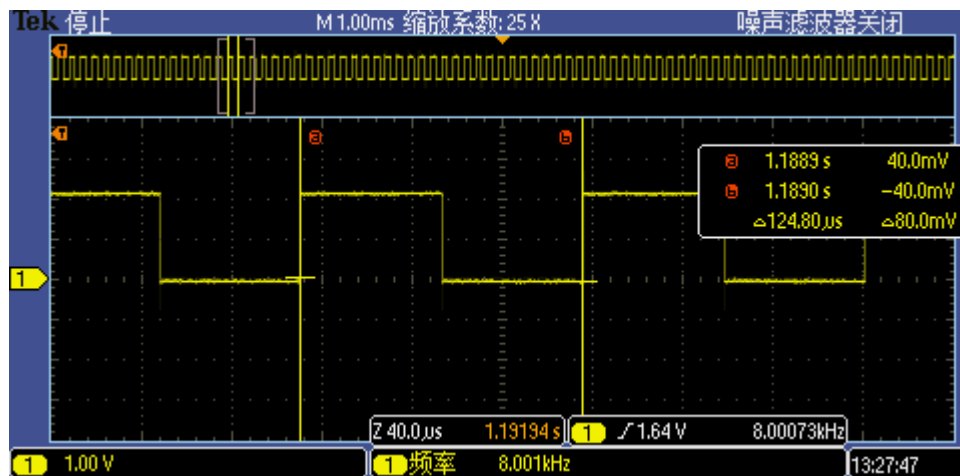
测试板外部高速晶振为 8MHz，HCLK、APB 总线时钟和 PWM 分频都为 1 分频，所以 PWM 时钟为 8MHz。

周期设定寄存器和 PWM 占空比设定寄存器值分别为 999 和 499，

理论周期时间： $T(m+1) = (1/8000000) * (999 + 1) = 125\mu s$ ;

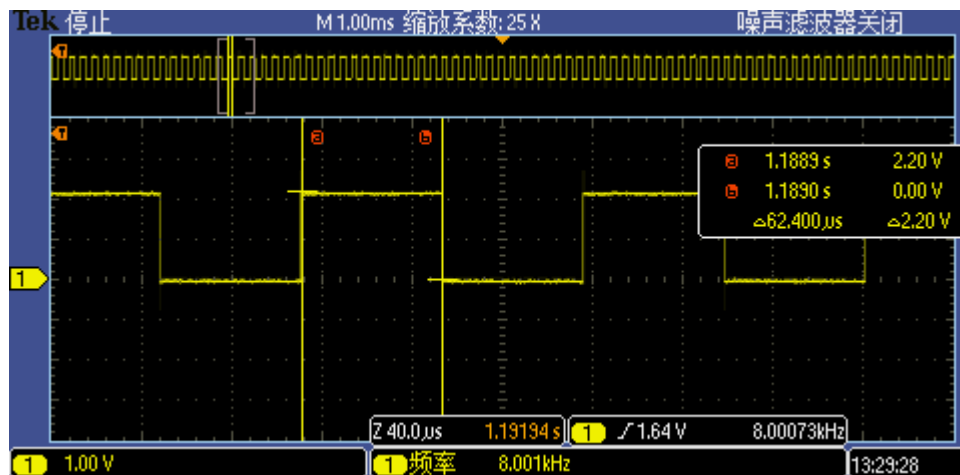
理论占空时间： $T(n+1) = (1/8000000) * (499 + 1) = 62.5\mu s$  (T:计数时钟周期、m:CSR 值、n:DUT 值)

实际测试周期时间，如下图所示：



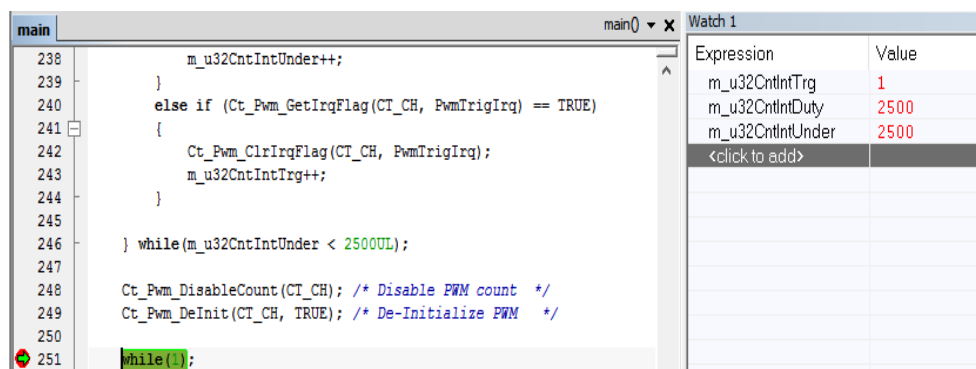


实际测试占空时间，如下图所示：



5) 继续运行。

6) 停在断点 251，查看 m\_u32CntIntUnder、m\_u32CntIntDuty、m\_u32CntIntTrg。



## 5 总结

以上章节简要介绍了复合定时器模块，详细说明了 HC32L15 系列的复合定时器寄存器和功能模式，演示了如何使用相关的复合定时器样例代码操作定时器。在开发中用户可以根据自己的实际需要使用复合定时器模块。

## 6 版本信息 & 联系方式

日期	版本	修改记录
2018/8/9	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](mailto:mcu@hdsc.com.cn)

网址: [www.hdsc.com.cn](http://www.hdsc.com.cn)

通信地址: 上海市张江高科园区碧波路 572 弄 39 号

邮编: 201203

---

