

## 32 位微控制器

# HC32M120 系列的通用定时器 TIMERB

适用对象

系列	产品型号
<b>HC32M120</b>	HC32M120J6TB
	HC32M120F6TB

# 目 录

<b>1</b>	<b>摘要 .....</b>	<b>3</b>
<b>2</b>	<b>TIMERB 简介 .....</b>	<b>3</b>
2.1	主要特性 .....	3
2.2	基本框图 .....	4
<b>3</b>	<b>HC32M120 系列的 TIMERB .....</b>	<b>5</b>
3.1	波形模式 .....	5
3.1.1	锯齿波模式.....	5
3.1.2	三角波模式.....	5
3.2	基本功能 .....	6
3.2.1	基本计数 .....	6
3.2.2	捕获输入 .....	6
3.2.3	比较输出 .....	7
3.3	寄存器说明.....	8
<b>4</b>	<b>样例代码 .....</b>	<b>9</b>
4.1	代码介绍 .....	9
4.2	工作流程 .....	13
4.3	代码运行 .....	14
<b>5</b>	<b>总结 .....</b>	<b>16</b>
<b>6</b>	<b>版本信息 &amp; 联系方式 .....</b>	<b>17</b>

## 1 摘要

本篇应用笔记主要介绍 HC32M120 系列通用定时器（TIMERB）模块，并简要说明如何使用 TIMERB 输出比较。

## 2 TIMERB 简介

通用定时器（TIMERB）是一个具有 16 位计数宽度、1 路 PWM 输出的定时器，该定时器支持三角波和锯齿波两种波形模式，可生成各种 PWM 波形，支持计数器同步启动；支持单元间级联实现 32 位计数；支持脉宽测量和周期测量。

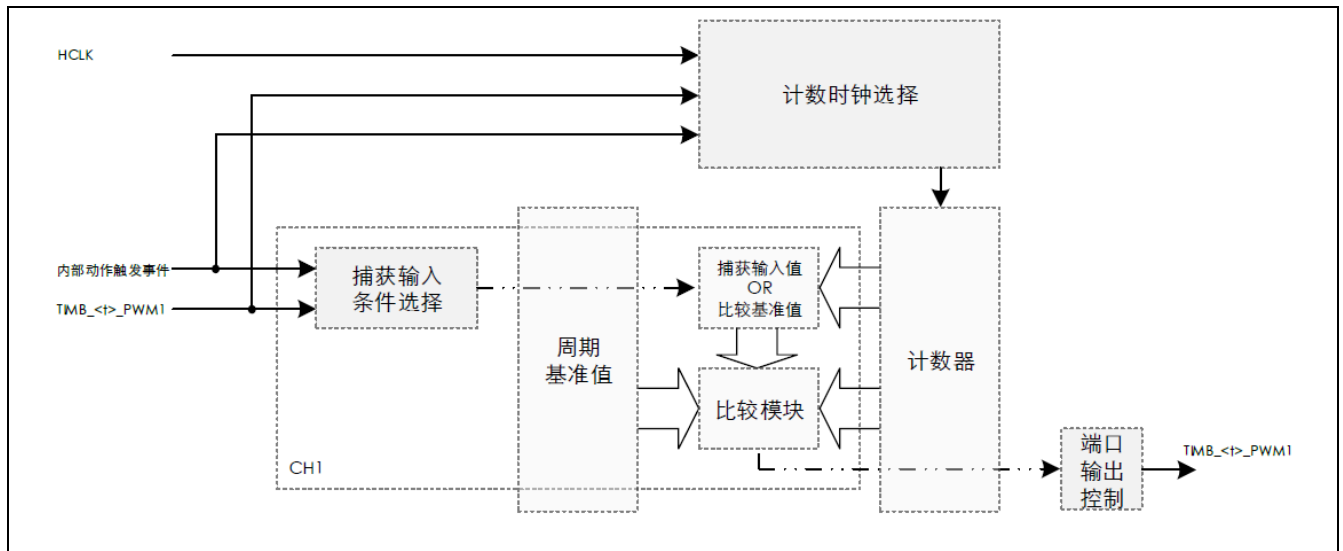
本系列产品搭载 4 个单元 TIMERB。

### 2.1 主要特性

- 波形模式：锯齿波、三角波
- 递加、递减计数方向
- 同步启动计数器
- 32 位级联计数
- 脉宽测量
- 周期测量
- 1 路 PWM 输出
- 比较匹配事件输出
- 比较匹配中断
- 周期匹配中断

## 2.2 基本框图

TIMERB 的基本框图如下图所示，图中”<t>”为单元编号，即”<t>”为 1~4；本文后续提到“<t>”时均指单元编号。



框图中的输入输出端口如下表：

端口名	方向	功能
TIMB_<m>_PWM1 (m=1~3)	输入或输出	1) 捕获输入事件端口或 PWM 输出端口 2) 硬件触发启动、停止、清零、计数事件输入端口
TIMB_<n>_PWM1 (n=4)	输入或输出	1) 捕获输入事件端口或 PWM 输出端口 2) 硬件触发启动、停止、清零、计数事件输入端口 3) 与 USART（通道 1）联动实现 LIN 功能

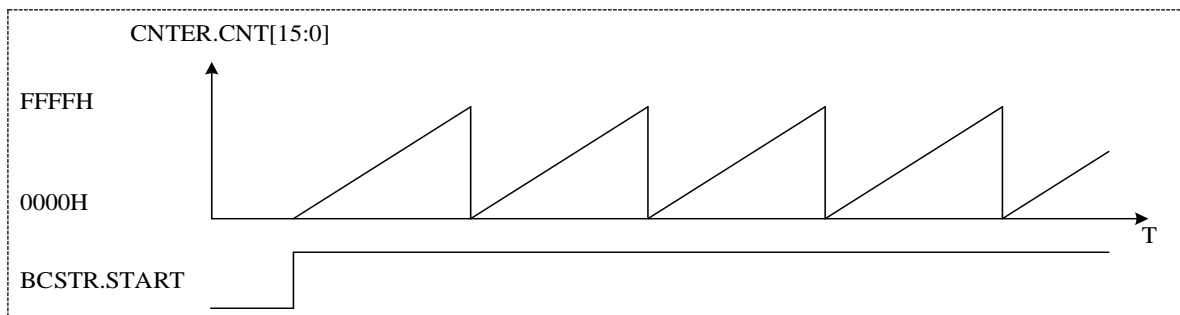
## 3 HC32M120 系列的 TIMERB

### 3.1 波形模式

TIMERB 有 2 种基本计数波形模式，锯齿波模式和三角波模式。

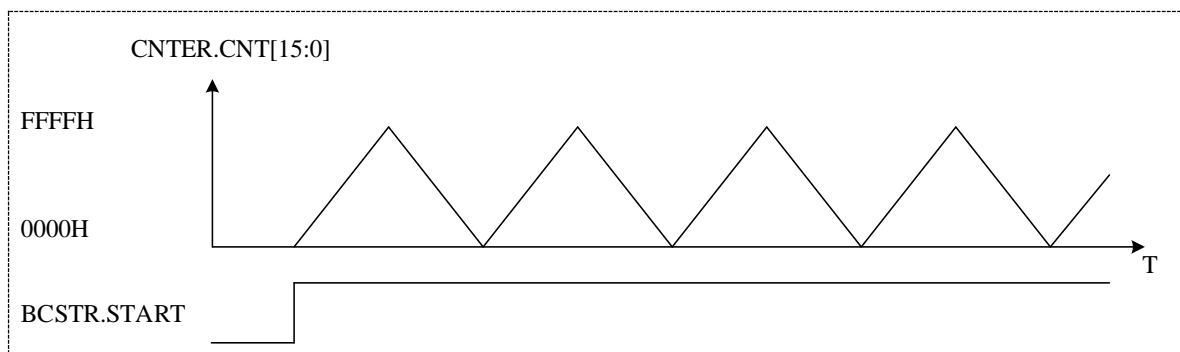
#### 3.1.1 锯齿波模式

计数器设置为锯齿波模式，基本波形如下图：



#### 3.1.2 三角波模式

计数器设置为三角波模式，基本波形如下图：



## 3.2 基本功能

### 3.2.1 基本计数

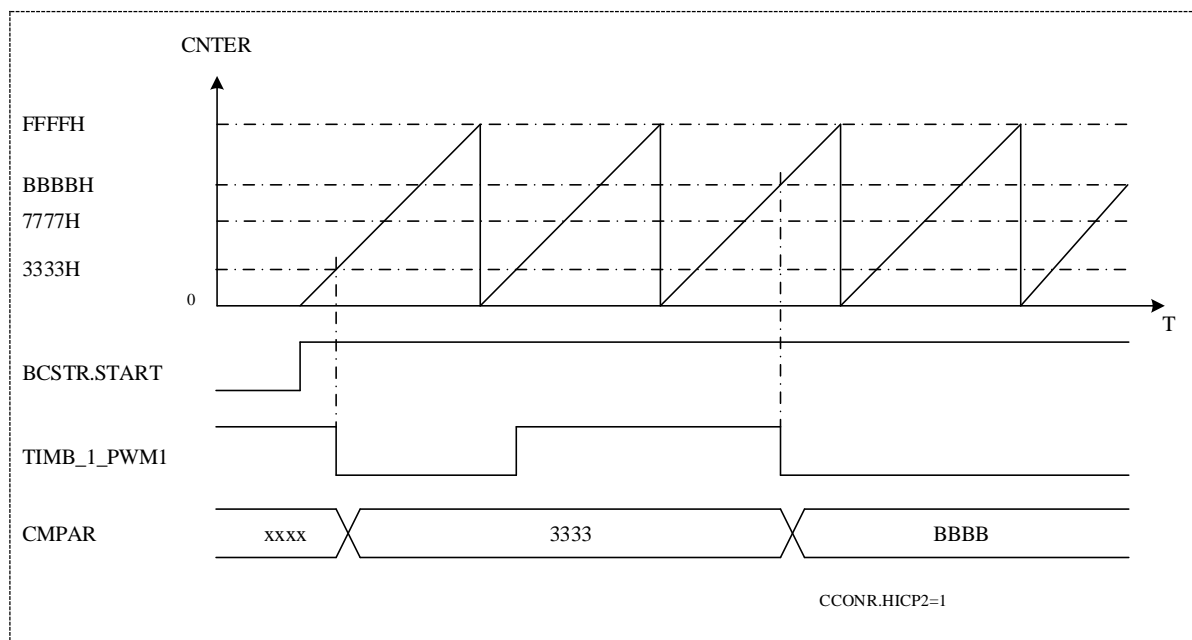
每个 **TIMERB** 单元可以设定一个基准计数值，根据配置的模式可以选择向上计数、向下计数，计数溢出时会可以配置相应的中断触发，每个通道可设定在计数值和基准值相等时产生计数比较匹配事件。

锯齿波模式递增计数至上溢点、锯齿波模式递减计数至下溢点、三角波模式计数至谷点或峰点，控制状态寄存器 (**TMRB\_BCSTR**) 的 **OVFF** 或 **UDFF** 位会被置为 1。

### 3.2.2 捕获输入

各个 **TIMERB** 单元的每个 **PWM** 输出通道都具有捕获输入功能，用于保存捕获到的计数值，设定捕获控制寄存器 (**TMRB\_CCONR**) 的 **CCONR.CAPMD** 位为 1，捕获输入功能变为有效。

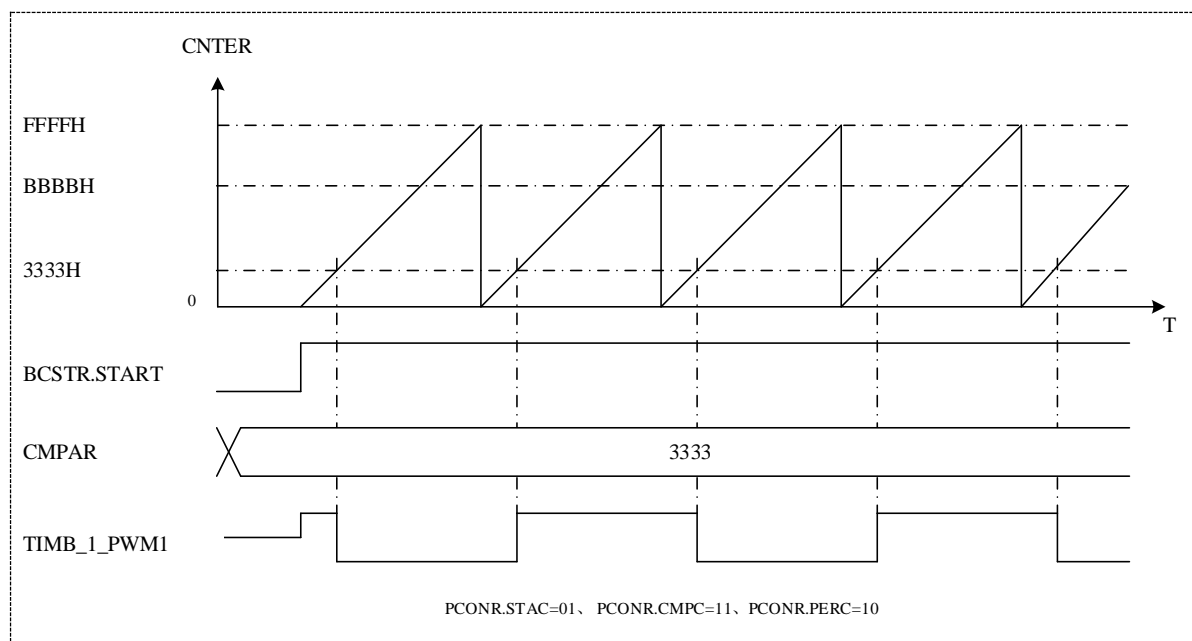
当捕获输入条件有效时，当前的计数值就被保存到相应的寄存器 (**TMRB\_CMPAR**)，捕获输入条件可以选择内部捕获动作触发事件（通过 **TMRB\_HTSSR** 寄存器选择）、**TIMB\_<t>\_PWM1** 端口输入等，具体的条件选择可通过捕获控制寄存器 (**TMRB\_CCONR**) 的 **HICP** 位来设定，下图为单元 1 的捕获输入动作示例：



### 3.2.3 比较输出

各个 **TIMERB** 单元内部均含有 1 个通道的比较输出 (**TIMB\_<t>\_PWM1**)，可在计数值与比较基准值比较匹配时输出指定的电平，**TMRB\_CMPAR** 寄存器分别对应 **TIMB\_<t>\_PWM1** 输出端口的计数比较基准值；当定时器的计数值和 **TMRB\_CMPAR** 相等时，**TIMB\_<t>\_PWM1** 端口输出指定的电平。

**TIMB\_<t>\_PWM1** 端口输出电平，可通过端口控制寄存器 (**TMRB\_PCONRn**) 的相关控制位设定控制，下图为单元 1 PWM 输出波形示例：



### 3.3 寄存器说明

通用定时器 TIMRB 模块的寄存器如下表所示，若需了解具体细节，请参考用户手册：

寄存器简称	寄存器功能	复位值
TMRB_CNTER	通用计数值寄存器	0x0000
TMRB_PERAR	周期基准值寄存器	0xFFFF
TMRB_CMPAR	比较基准值寄存器	0xFFFF
TMRB_BCSTR	控制状态寄存器	0x0002
TMRB_ICONR	中断控制寄存器	0x0000
TMRB_ECONR	事件控制寄存器	0x0000
TMRB_STFLR	状态标志寄存器	0x0000
TMRB_CCONR	捕获控制寄存器	0x0000
TMRB_PCONR	端口控制寄存器	0x0000
TMRB_HCONR	硬件触发事件选择寄存器	0x0000
TMRB_HCUPR	硬件递加事件选择寄存器	0x0000
TMRB_HCDOR	硬件递减事件选择寄存器	0x0000
TMRB_HTSSR	内部触发事件选择寄存器	0x0000007F



## 4 样例代码

### 4.1 代码介绍

用户可以根据上述的介绍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到 HC32M120 系列 MCU 的设备驱动库（Device Driver Library, DDL）来体验 TIMERB 的正交编码计数功能。

以下部分主要基于 DDL 的 TIMERB 模块的输入捕获样例 timerb\_input\_capture 代码，简要介绍 TIMERB 的输入捕获和输出比较功能使用方法：

#### 1) 配置系统时钟

```
/* Configure system clock. */  
SystemClockConfig();
```

#### 2) 配置 LED:

```
/* Configure RGB LED. */  
LedConfig();
```

#### 3) 配置外部中断:

```
/* External interrupt Ch.0 initialize */  
GPIO_StructInit(&stcGpioInit);  
stcGpioInit.u16ExInt = PIN_EXINT_ON;  
GPIO_Init(EXINT_PORT, EXINT_PIN, &stcGpioInit);  
  
/* EXINT Channel 0 (SW2) configure */  
EXINT_StructInit(&stcExIntInit);  
stcExIntInit.u16ExIntCh = EXINT_CH;  
stcExIntInit.u8ExIntFE = EXINT_FILTER_ON;  
stcExIntInit.u8ExIntFClk = EXINT_FCLK_HCLK_DIV8;  
stcExIntInit.u8ExIntLvl = EXINT_TRIGGER_FALLING;  
EXINT_Init(&stcExIntInit);  
  
/* Configure NVIC */  
NVIC_ClearPendingIRQ(EXINT_IRQn);  
NVIC_SetPriority(EXINT_IRQn, DDL_IRQ_PRIORITY_03);  
NVIC_EnableIRQ(EXINT_IRQn);
```

4) 初始化 TIMERB 单元 3 输出比较功能:

```
static void TimerbOcConfig(void)
{
    stc_timerb_oc_init_t stcTimerbOcInit = {
        .u16CompareVal = 0u,
        .u16PortOutputState = TIMERB_OC_PORT_OUTPUT_ENABLE,
        .u16StartCntOutput = TIMERB_OC_STARTCNT_OUTPUT_HIGH,
        .u16StopCntOutput = TIMERB_OC_STOPCNT_OUTPUT_HIGH,
        .u16CompareMatchOutput = TIMERB_OC_CMPMATCH_OUTPUT_INVERTED,
        .u16PeriodMatchOutput = TIMERB_OC_PERIODMATCH_OUTPUT_INVERTED,
    };

    /* Configure TIM_<t>_PWM1. */
    GPIO_SetFunc(TIMERB_OC_UNIT_PWM1_PORT, TIMERB_OC_UNIT_PWM1_PIN,
TIMERB_OC_UNIT_PWM1_GPIO_FUNC);

    /* Initialize TIMERB unit. */
    m_stcTimerbInit.u16PeriodVal = (uint16_t)TIMERB_OC_UNIT_PERIOD_VALUE;
    TIMERB_Init(TIMERB_OC_UNIT, &m_stcTimerbInit);

    /* Initialize TIMERB output compare function . */
    stcTimerbOcInit.u16CompareVal = (uint16_t)TIMERB_OC_UNIT_COMPARE_VALUE;
    TIMERB_OC_Init(TIMERB_OC_UNIT, &stcTimerbOcInit);
}
```

5) 配置 TIMERB 端口功能:

```
/* Configure TIMERB PWM port. */
GPIO_SetFunc(TIMERB_IC_ODD_UNIT_PWM1_PORT, TIMERB_IC_ODD_UNIT_PWM1_PIN,
TIMERB_IC_ODD_UNIT_PWM1_GPIO_FUNC);
GPIO_SetFunc(TIMERB_IC_EVEN_UNIT_PWM1_PORT, TIMERB_IC_EVEN_UNIT_PWM1_PIN,
TIMERB_IC_EVEN_UNIT_PWM1_GPIO_FUNC);
```

## 6) 初始化 TIMERB 单元 1 和单元 2 输入捕获功能:

```

/* Initialize TIMERB odd unit. */
m_stcTimerbInit.u16PeriodVal = (uint16_t)TIMERB_IC_ODD_UNIT_PERIOD_VALUE;
TIMERB_Init(TIMERB_IC_ODD_UNIT, &m_stcTimerbInit);

/* Initialize TIMERB even unit input capture function. */
TIMERB_IC_Init(TIMERB_IC_ODD_UNIT, &stcTimerbIcInit);
TIMERB_IntCmd(TIMERB_IC_ODD_UNIT, TIMERB_IT_CMP, Enable);

/* Register IRQ handler && configure NVIC. */
stcIrqRegiConf.enIRQn = TIMERB_IC_ODD_UNIT_CMP_IRQn;
stcIrqRegiConf.enIntSrc = TIMERB_IC_ODD_UNIT_CMP_INT;
stcIrqRegiConf.pfnCallback = &TimerbIcOddUnitCmpIrqCallback;
INTC_IrqRegistration(&stcIrqRegiConf);
NVIC_ClearPendingIRQ(stcIrqRegiConf.enIRQn);
NVIC_SetPriority(stcIrqRegiConf.enIRQn, DDL_IRQ_PRIORITY_03);
NVIC_EnableIRQ(stcIrqRegiConf.enIRQn);

/* Initialize TIMERB even unit. */
m_stcTimerbInit.u16PeriodVal = (uint16_t)TIMERB_IC_EVEN_UNIT_PERIOD_VALUE;
TIMERB_Init(TIMERB_IC_EVEN_UNIT, &m_stcTimerbInit);

/* Initialize TIMERB even unit input capture function . */
TIMERB_IC_Init(TIMERB_IC_EVEN_UNIT, &stcTimerbIcInit);
TIMERB_IntCmd(TIMERB_IC_EVEN_UNIT, TIMERB_IT_CMP, Enable);

/* Register IRQ handler && configure NVIC. */
stcIrqRegiConf.enIRQn = TIMERB_IC_EVEN_UNIT_CMP_IRQn;
stcIrqRegiConf.enIntSrc = TIMERB_IC_EVEN_UNIT_CMP_INT;
stcIrqRegiConf.pfnCallback = &TimerbIcEvenUnitCmpIrqCallback;
INTC_IrqRegistration(&stcIrqRegiConf);
NVIC_ClearPendingIRQ(stcIrqRegiConf.enIRQn);
NVIC_SetPriority(stcIrqRegiConf.enIRQn, DDL_IRQ_PRIORITY_03);
NVIC_EnableIRQ(stcIrqRegiConf.enIRQn);

```

## 7) 设置 TIMERB 触发条件:

```

/* Set trigger source. */
TIMERB_SetTriggerSrc(EVT_PORT_EIRQ0);

/* Start TIMERB start count condition. */
TIMERB_SetHwStartCondition(TIMERB_OC_UNIT, TIMERB_HWSTART_TIMB_EVT);
TIMERB_SetHwStartCondition(TIMERB_IC_ODD_UNIT, TIMERB_HWSTART_TIMB_EVT);
TIMERB_SetHwStartCondition(TIMERB_IC_EVEN_UNIT, TIMERB_HWSTART_TIMB_EVT);

```

8) TIMERB 捕获功能中断处理函数：读取比较值，并写入缓存。

```
/**
 * @brief TIMERB odd unit compare match IRQ callback.
 * @param None
 * @retval None
 */
static void TimerbIcOddUnitCmpIrqCallback(void)
{
    static stc_buffer_t m_stcOddUnitCmpBuf = {
        .u16Index = 0u,
        .u16Capacity = TIMERB_IC_CMPBUF_SIZE,
    };

    BufferWrite(&m_stcOddUnitCmpBuf, TIMERB_GetCompare(TIMERB_IC_ODD_UNIT));
    TIMERB_ClearFlag(TIMERB_IC_ODD_UNIT, TIMERB_FLAG_CMP);
}

/**
 * @brief TIMERB even unit compare match IRQ callback
 * @param None
 * @retval None
 */
static void TimerbIcEvenUnitCmpIrqCallback(void)
{
    static stc_buffer_t m_stcEvenUnitCmpBuf= {
        .u16Index = 0u,
        .u16Capacity = TIMERB_IC_CMPBUF_SIZE,
    };

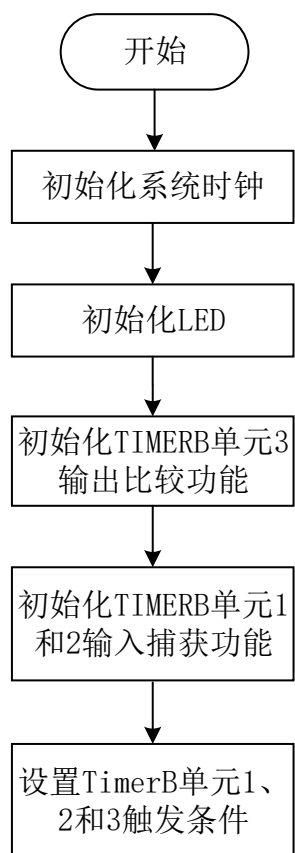
    BufferWrite(&m_stcEvenUnitCmpBuf, TIMERB_GetCompare(TIMERB_IC_EVEN_UNIT));
    TIMERB_ClearFlag(TIMERB_IC_EVEN_UNIT, TIMERB_FLAG_CMP);
}
```

9) 外部中断的中断处理函数：点亮红色 LED 灯，提示触发条件产生。

```
/**
 * @brief External interrupt Ch.0 ISR
 * @param None
 * @retval None
 */
void EXINT00_Handler(void)
{
    if (Set == EXINT_GetExIntSrc(EXINT_CH00))
    {
        EXINT_ClrSrc(EXINT_CH00);
        LED_G_ON();
    }
}
```

## 4.2 工作流程

样例代码中 **TIMERB** 操作流程如下图所示：

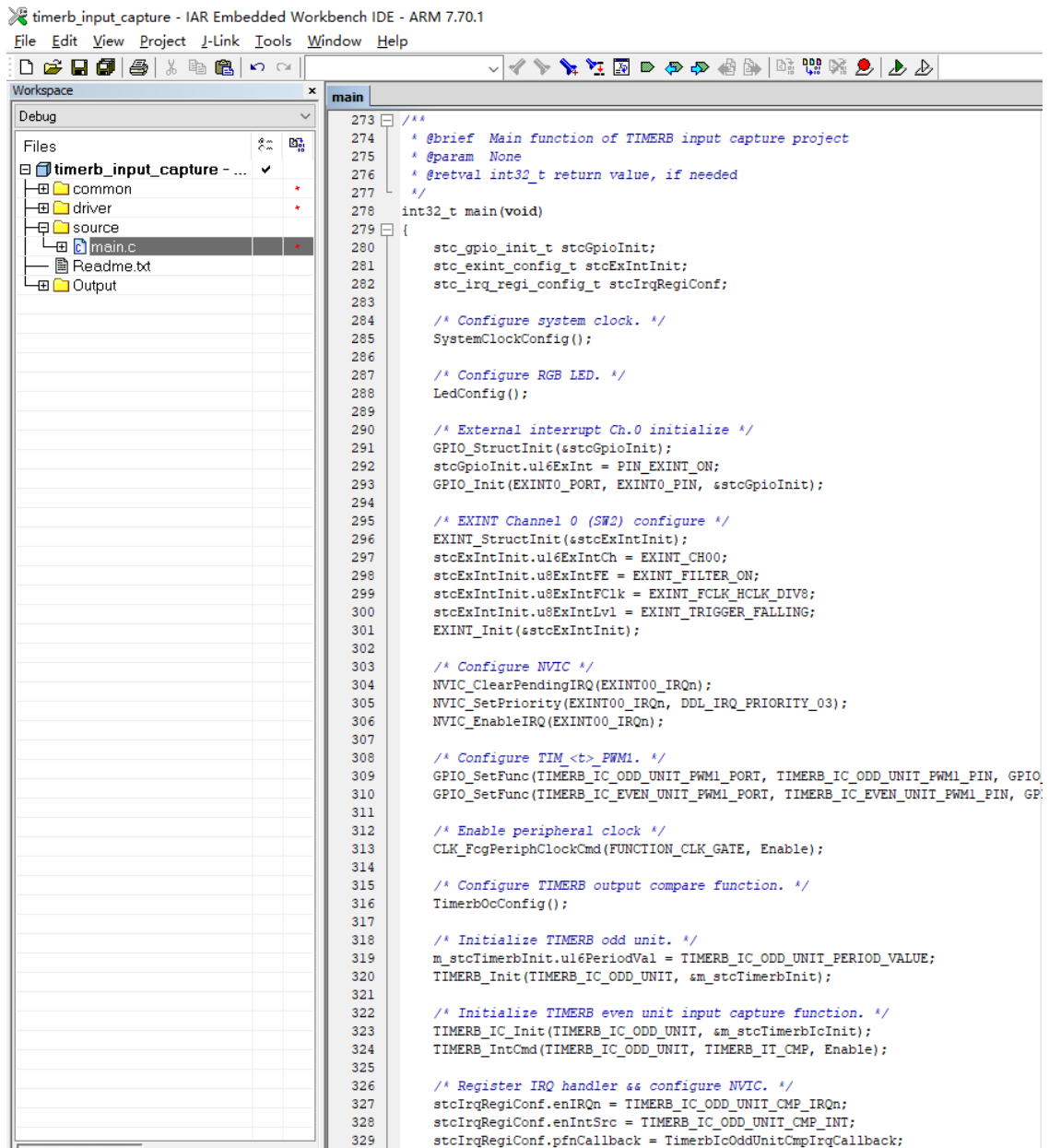




## 4.3 代码运行

用户可以通过华大半导体的网站下载到 DDL 的样例代码（`timerb_input_capture`），并配合评估用板（‘`STK_HC32M120_LQFP48_050_V11`’）运行相关代码学习使用 **TIMERB** 模块。

以下部分主要介绍如何在 ‘`STK_HC32M120_LQFP48_050_V11`’ 评估板上，通过 **IAR EWARM** 编译、运行 `timerb_input_capture` 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 ‘`timerb_input_capture`’ 评估板。
- 从华大半导体网站下载 **HC32M120 DDL** 代码。
- 下载并运行 `timerb\timerb_input_capture\` 中的项目文件：
  - 1) 将 `P15(TIMB_1_PWM1)`、`P16(TIMB_2_PWM1)`、`P62(TIMB_3_PWM1)` 相连
  - 2) 打开 `timerb_input_capture\` 项目，并打开 ‘`main.c`’ 如下视图：



- 3) 点击  重新编译整个项目；
- 4) 点击  将代码下载到评估板上，全速运行；
- 5) 按下 SW2 按键；
- 6) 观察测试板，绿色 LED 灯点亮，表示产生外部中断事件。
- 7) 通过 IDE 的 Live Watch 功能，查看捕获输入缓存变量 m\_stcEvenUnitCmpBuf、m\_stcOddUnitCmpBuf 为 TIMERB 单元 3 TMRB\_CMPAR 值加 1。

## 5 总结

以上章节简要介绍 HC32M120 系列的 TIMERB 寄存器、功能模式。演示了如何操作 TIMERB 输入捕获样例代码，在开发中用户可以根据自己的实际需要使用 TIMERB 模块。



## 6 版本信息 & 联系方式

日期	版本	修改记录
2019/12/10	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: 上海市浦东新区中科路 1867 号 A 座 10 层

邮编: 201203

---

