

32 位 微控制器

HC32M120 系列的通用定时器 TIMERA

适用对象

系列	产品型号
HC32M120	HC32M120J6TB
	HC32M120F6TB

目 录

1	摘要	3
2	TIMERA 简介	3
2.1	主要特性	3
2.2	基本框图	4
3	HC32M120 系列的 TIMERA	5
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	正交编码	8
3.3.1	相位差计数.....	8
3.3.2	方向计数	9
3.4	特殊功能	10
3.4.1	缓存功能	10
3.5	寄存器介绍.....	11
4	样例代码	12
4.1	代码介绍	12
4.2	工作流程	14
4.3	代码运行	15
5	总结	16
6	版本信息 & 联系方式	17

1 摘要

本篇应用笔记主要介绍 HC32M120 系列通用定时器（TIMERA）模块，并简要说明通过 TIMERA 的正交编码计数功能如何实现 2 相正交编码计数。

2 TIMERA 简介

通用定时器（TIMERA）是一个具有 16 位计数宽度、2 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，比较基准值寄存器支持缓存功能，支持 2 相正交编码计数。

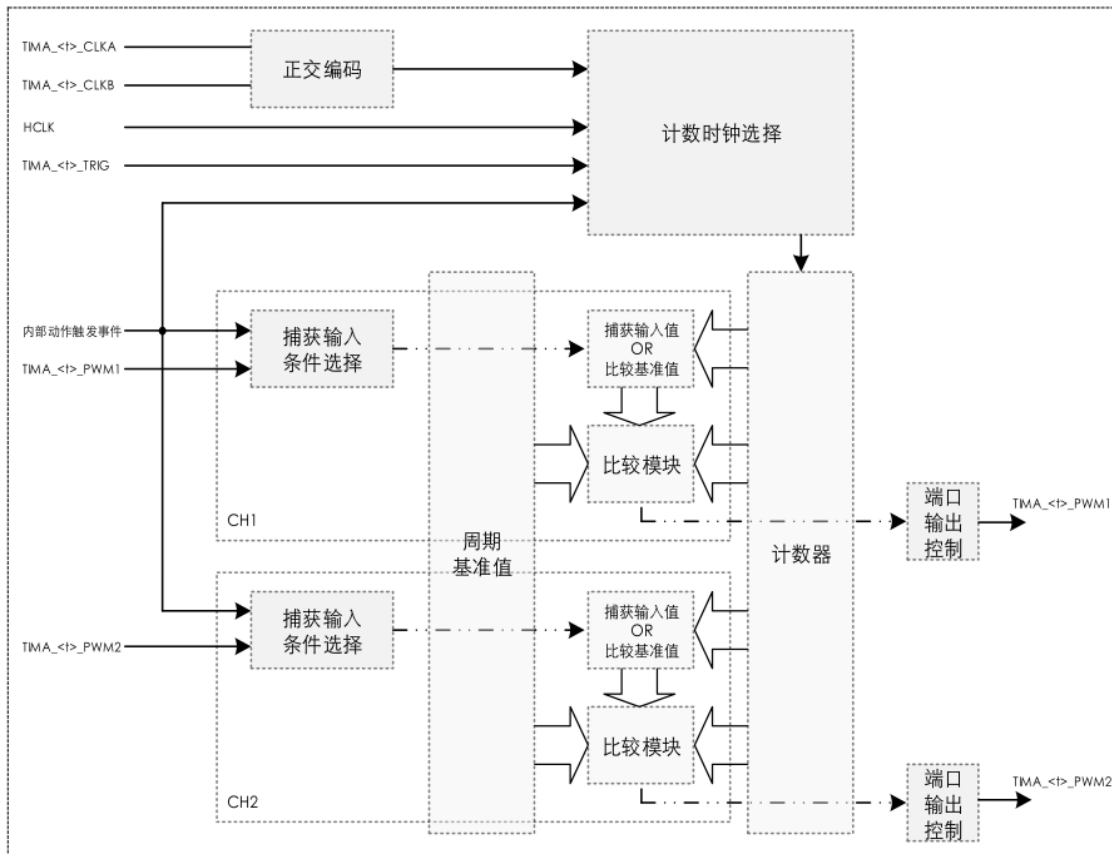
本系列产品搭载 1 个单元 TIMERA，可实现 2 路 PWM 输出。

2.1 主要特性

- 波形模式：锯齿波、三角波
- 递加、递减计数方向
- 基准值缓存功能
- 正交编码计数
- 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

TIMERA 的基本框图如下图所示，图中“<t>”为单元编号，即“<t>”为1；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
TIMA_<t>_PWMn	输入或输出	捕获输入事件端口或 PWM 输出端口（n=1~2）
TIMA_<t>_CLKA	输入	正交编码计数事件输入端口
TIMA_<t>_CLKB		
TIMA_<t>_TRIG	输入	硬件触发启动、停止、清零事件输入端口

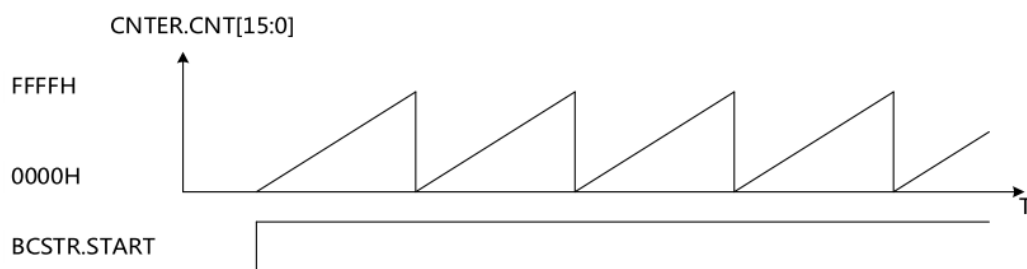
3 HC32M120 系列的 TIMERA

3.1 波形模式

TIMERA 有 2 种基本计数波形模式，锯齿波模式和三角波模式。

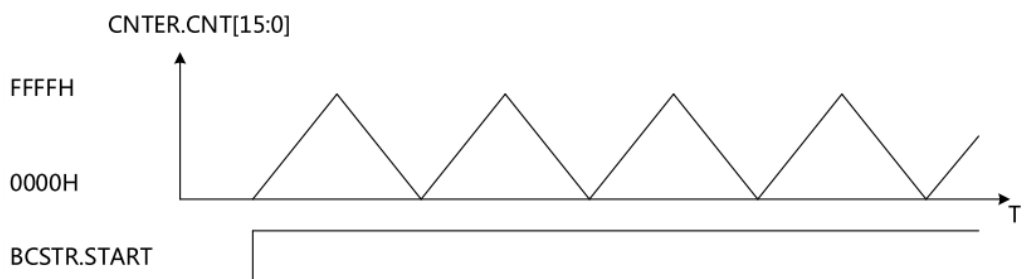
3.1.1 锯齿波模式

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 TIMERA 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。

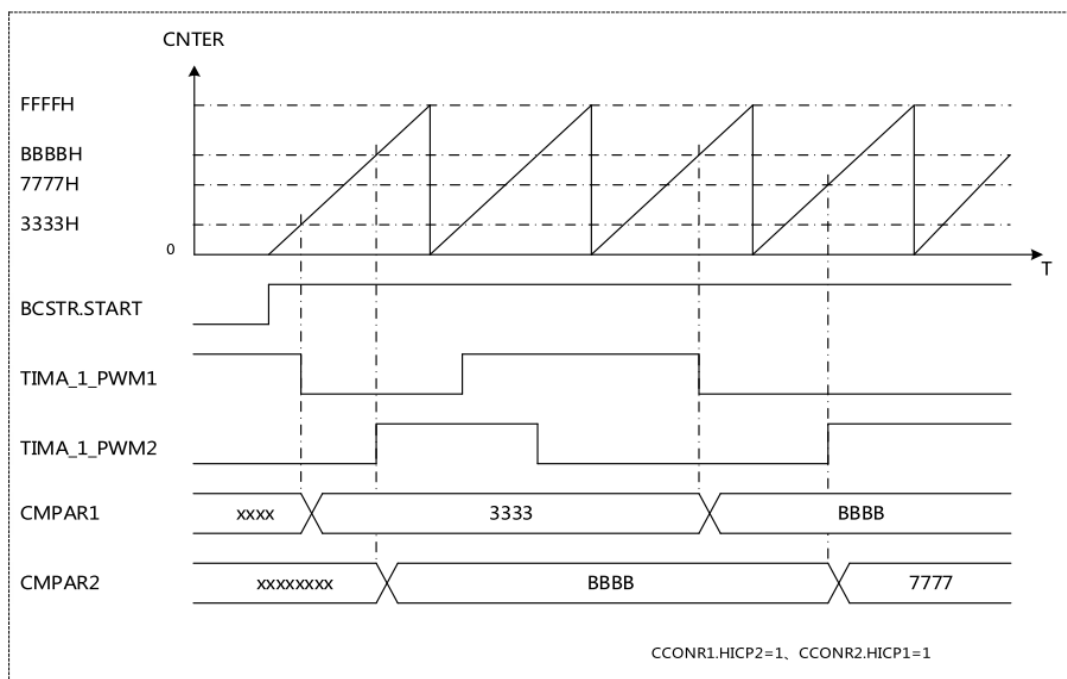
锯齿波模式递增计数至上溢点、锯齿波模式递减计数至下溢点、三角波模式计数至谷点或峰点，控制状态寄存器（BCSTR）的 OVFF 或 UDFE 位会被置为 1。

3.2.2 捕获输入

每个 TIMERA 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（CCONRn）的 CCONR.CAPMD 位为 1，捕获输入功能变为有效，每个单元的 TIMA_<t>_PWMn（捕获输入功能时）端口输入都有数字滤波功能。

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（CMPARn）中（n=1~2），捕获输入条件可以选择内部捕获动作触发事件（通过 HTSSR 寄存器选择）、

TIMA_<t>_PWMn 端口输入等，具体的条件选择可通过捕获控制寄存器（CCONRn）的 HICP 位来设定（n=1~2），下图为单元 2 的捕获输入动作示例：

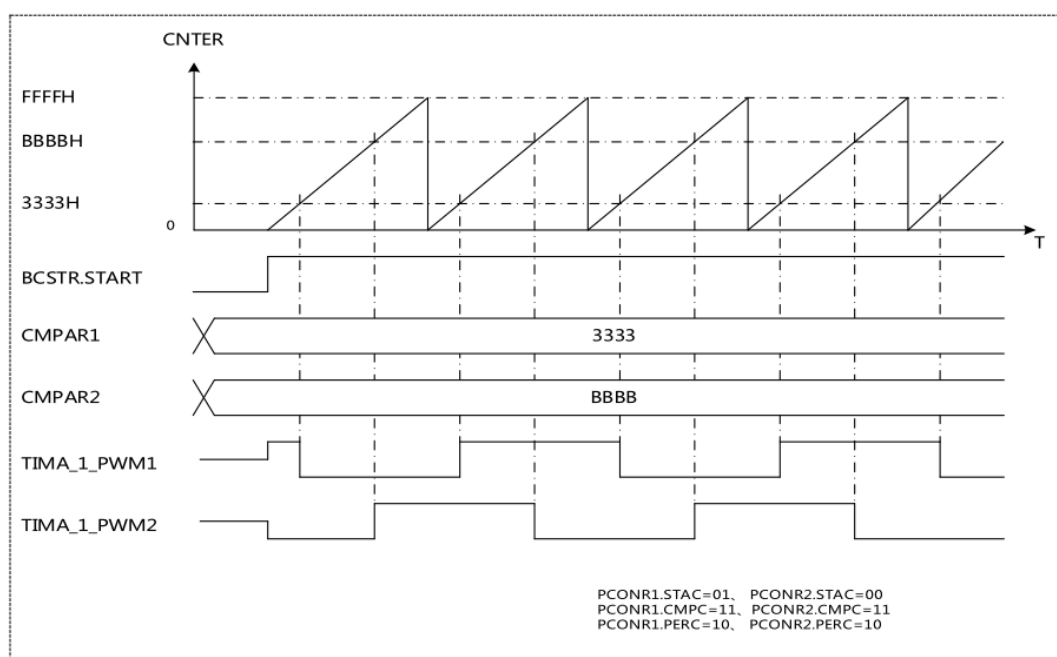


3.2.3 比较输出

每个 TIMERA 单元内部均含有 2 个通道的比较输出 (TIMA_<t>_PWMn)，可在计数值与比较基准值比较匹配时输出指定的电平，CMPARn 寄存器分别对应 TIMA_<t>_PWMn 输出端口的计数比较基准值；当定时器的计数值和 CMPARn 相等时，TIMA_<t>_PWMn 端口输出指定的电平 (n=1~2)。

TIMERA 内部的 2 路输出 TIMA_<t>_PWMn，每路输出都可以通过端口控制寄存器

(PCONRn) 的相关控制位实现不同的输出波形 (n=1~2)，下图为锯齿波模式下，PWM 输出波形示例：



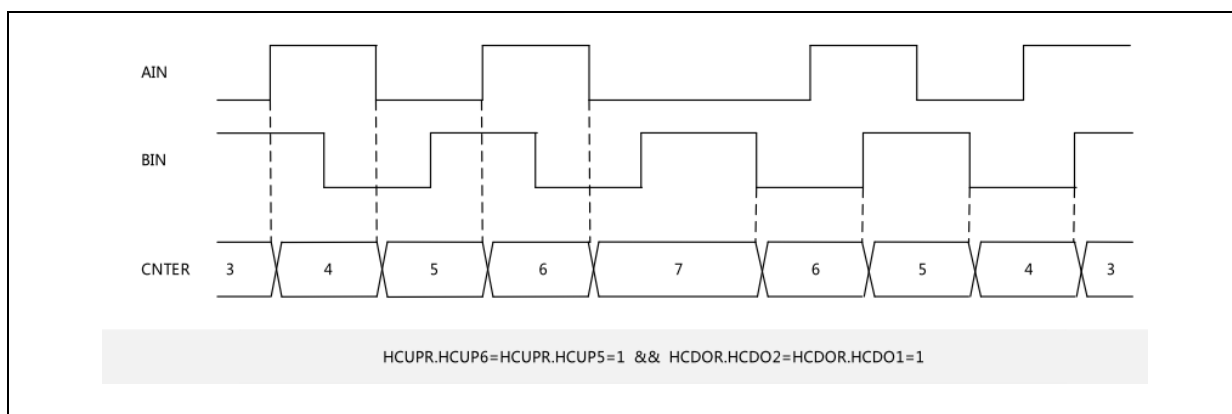
3.3 正交编码

将 TIMA_<t>_CLKA 输入看作 AIN 输入、TIMA_<t>_CLKB 输入看作 BIN 输入，TIMERA 就可以实现二路输入的正交编码计数。

每个单元的 AIN、BIN 输入可以实现位置计数模式，位置计数模式根据 AIN、BIN 的输入实现基本计数功能、相位差计数功能和方向计数功能。

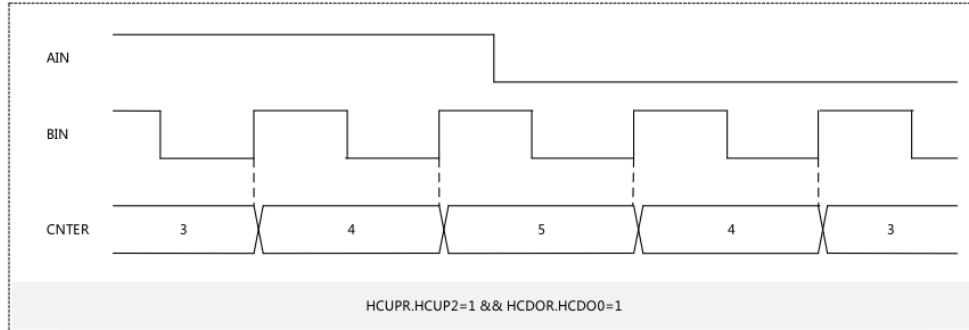
3.3.1 相位差计数

相位差计数是指根据 AIN 和 BIN 的相位关系进行计数。根据设定的不同，可以实现 1 倍计数、2 倍计数、4 倍计数等，如下图所示为相位差 2 倍计数：



3.3.2 方向计数

方向计数是指将 AIN 的输入状态设定为方向控制，将 BIN 的输入作为时钟计数，如下图所示：



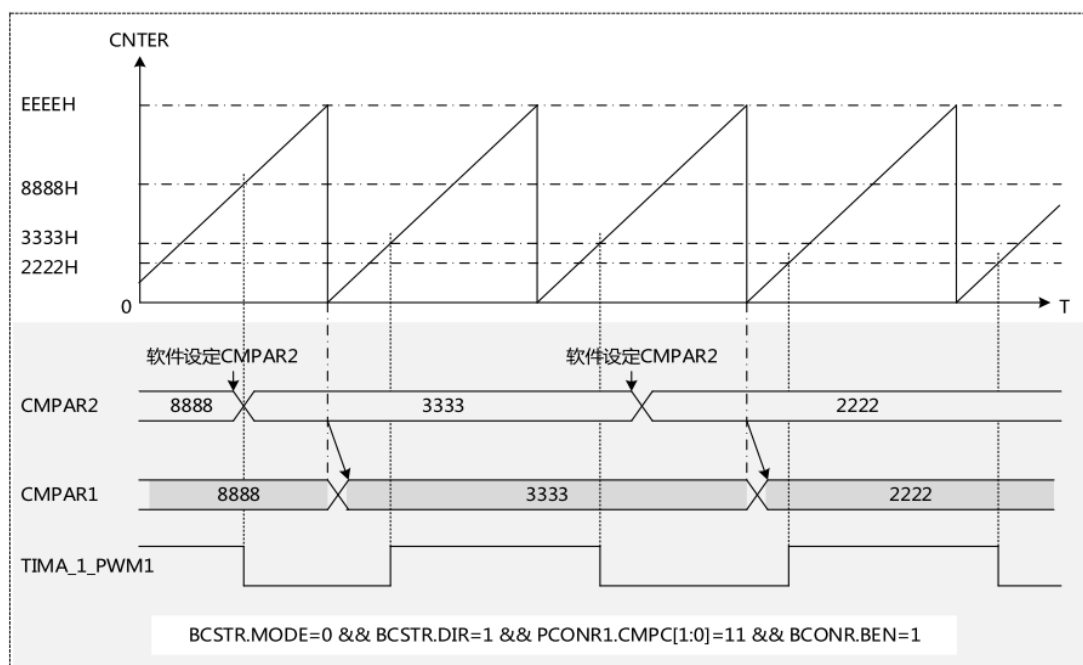
3.4 特殊功能

3.4.1 缓存功能

TIMERA 的共计 2 个比较基准寄存器 (CMPARn) 可以成对实现缓存功能 (n=1~2)，即 CMPAR2 作为 CMPAR1 的缓存基准值，缓存控制寄存器 (BCONR) 实现对缓存功能的控制。

当缓存控制寄存器 (BCONR) 的 BEN 位被置位时，缓存功能变为有效，此时计数器计数到特定时间点时就发生一次缓存传送 (CMPAR2->CMPAR1)，该“特定时间点”有几种情况，详见参考手册。

下图为锯齿波模式的缓存传送示意图：



3.5 寄存器介绍

通用定时器 TIMERA 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	备注
TMRA_CNTER	通用计数值寄存器	
TMRA_PERAR	周期基准值寄存器	
TMRA_CMPARn	比较基准值寄存器	n=1~2
TMRA_BCSTR	控制状态寄存器	
TMRA_ICONR	中断控制寄存器	
TMRA_ECONR	事件控制寄存器	
TMRA_FCONR	滤波控制寄存器	
TMRA_STFLR	状态标志寄存器	
TMRA_BCONR	缓存控制寄存器	
TMRA_CCONRn	捕获控制寄存器	n=1~2
TMRA_PCONRn	端口控制寄存器	n=1~2
TMRA_HCONR	硬件触发事件选择寄存器	
TMRA_HCUPR	硬件递加事件选择寄存器	
TMRA_HCDOR	硬件递减事件选择寄存器	
TMRA_HTSSR	内部触发事件选择寄存器	

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到 HC32M120 系列 MCU 的设备驱动库（Device Driver Library，DDL）来体验 TIMERA 的正交编码计数功能。

以下部分主要基于 DDL 的 TIMERA 模块的正交编码计数功能样例 timer_phase_difference 代码，简要介绍 TIMERA 的正交编码计数使用方法：

- 1) 开启 TIMERA 时钟：

```
/* Configuration peripheral clock */  
CLK_FcgPeriphClockCmd(TIMERA_UNIT1_CLOCK, Enable);
```

- 2) 配置 TIMERA 正交编码计数功能使用的 IO：

```
/* Configuration TIMERA capture Port */  
GPIO_SetFunc(TIMERA_UNIT1_CLKA_PORT, TIMERA_UNIT1_CLKA_PIN,  
              TIMERA_UNIT1_CLKA_FUNC);  
GPIO_SetFunc(TIMERA_UNIT1_CLKB_PORT, TIMERA_UNIT1_CLKB_PIN,  
              TIMERA_UNIT1_CLKB_FUNC);
```

- 3) 初始化 TIMERA 单元 1 基本计数和正交编码计数功能：

```
/* Configuration timera 1 unit structure */  
stcTimeraInit.u16CountMode = TIMERA_SAWTOOTH_WAVE;  
stcTimeraInit.u16PeriodVal = TIMERA_UNIT1_PERIOD_VALUE; //50Hz  
stcTimeraInit.u16ClkAFilterState = TIMERA_CLKA_FILTER_ENABLE;  
stcTimeraInit.u16ClkAFilterClkDiv = TIMERA_CLKA_CLKDIV_DIV4;  
stcTimeraInit.u16ClkBFilterState = TIMERA_CLKB_FILTER_ENABLE;  
stcTimeraInit.u16ClkBFilterClkDiv = TIMERA_CLKB_CLKDIV_DIV4;  
stcTimeraInit.u16HwUpCondition = TIMERA_HWUP_CLKB_HIGH_CLKA_RISING;  
stcTimeraInit.u16HwDownCondition =  
TIMERA_HWDOWN_CLKB_LOW_CLKA_RISING;  
TIMERA_Init(TIMERA_UNIT1, &stcTimeraInit);  
TIMERA_IntCmd(TIMERA_UNIT1, TIMERA_INT_OVF | TIMERA_INT_UDF, Enable);
```

4) 初始化 TIMERA 单元 1 计数上溢和下溢中断功能:

```
/* Configuration timera 1 unit overflow interrupt */
stcIrqRegiConf.enIntSrc = TIMERA_UNIT1_OVF_INTn;
stcIrqRegiConf.enIRQn = TIMERA_UNIT1_OVF_IRQn;
stcIrqRegiConf.pfnCallback = TimeraUnit1Overflow_IrqCallback;
INTC_IrqRegistration(&stcIrqRegiConf);
NVIC_ClearPendingIRQ(stcIrqRegiConf.enIRQn);
NVIC_SetPriority(stcIrqRegiConf.enIRQn, DDL_IRQ_PRIORITY_DEFAULT);
NVIC_EnableIRQ(stcIrqRegiConf.enIRQn);

/* Configuration timera 1 unit underflow interrupt */
stcIrqRegiConf.enIntSrc = TIMERA_UNIT1_UDF_INTn;
stcIrqRegiConf.enIRQn = TIMERA_UNIT1_UDF_IRQn;
stcIrqRegiConf.pfnCallback = TimeraUnit1Underflow_IrqCallback;
INTC_IrqRegistration(&stcIrqRegiConf);
NVIC_ClearPendingIRQ(stcIrqRegiConf.enIRQn);
NVIC_SetPriority(stcIrqRegiConf.enIRQn, DDL_IRQ_PRIORITY_DEFAULT);
NVIC_EnableIRQ(stcIrqRegiConf.enIRQn);
```

5) 启动 TIMERA 单元 1 计数:

```
/* Start TIMERA counter */
TIMERA_Enable(TIMERA_UNIT1);
```

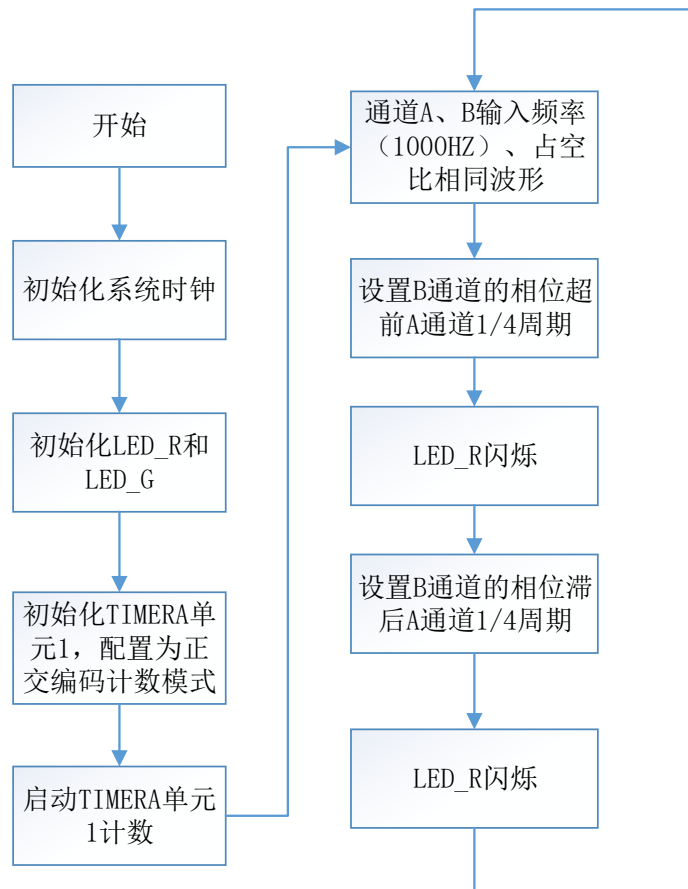
6) 设置 TimerA 时钟源为正交编码端口 CLKA 和 CLKB 输入, 设置 P75 为端口 CLKA、P74 为端口 CLKB, 配置端口 CLKB 为高电平时, 端口 CLKA 采样到上升沿触发递加计数, 配置端口 CLKB 为低电平时, 端口 CLKA 采样到上升沿触发递减计数, 递加计数溢出则 LED_R 切换状态, 递减计数溢出则 LED_G 切换状态。

```
/**
 * @brief TIMERA 1 unit overflow interrupt callback function.
 * @param None
 * @retval None
 */
static void TimeraUnit1Overflow_IrqCallback(void)
{
    LED_R_TOGGLE();
    TIMERA_ClearFlag(TIMERA_UNIT1, TIMERA_FLAG_OVF);
}

/**
 * @brief TIMERA 1 unit underflow interrupt callback function.
 * @param None
 * @retval None
 */
static void TimeraUnit1Underflow_IrqCallback(void)
{
    LED_G_TOGGLE();
    TIMERA_ClearFlag(TIMERA_UNIT1, TIMERA_FLAG_UDF);
}
```

4.2 工作流程

样例代码中 TIMERA 操作流程如下图所示：

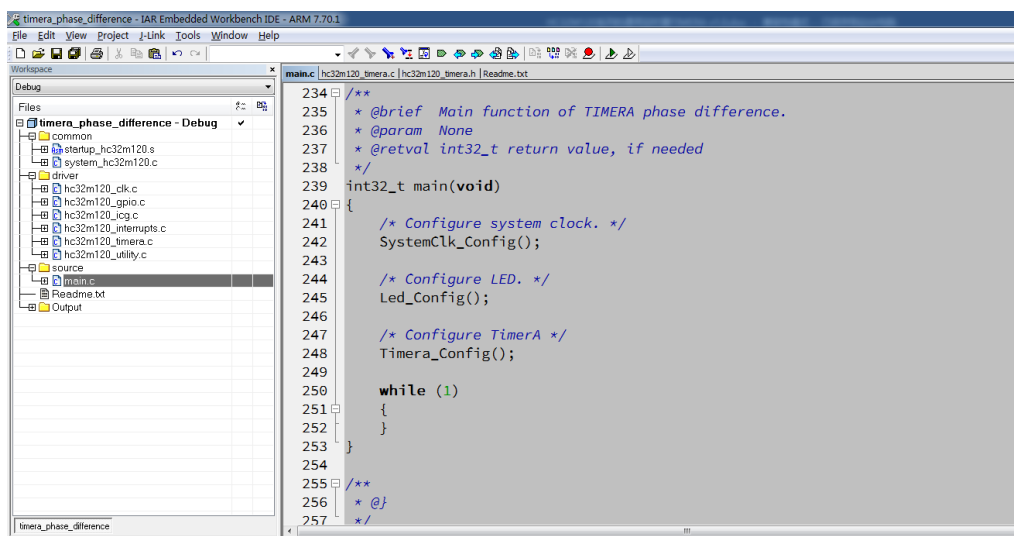




4.3 代码运行

用户可以通过华大半导体的网站下载到 DDL 的样例代码（timera_base_timer、timera_capture_input、timera_compare_output、timera_phase_difference），并配合评估用板（比如‘STK_HC32M120_LQFP48_050_V11’）运行相关代码学习使用 TIMERA 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 timera_phase_difference 样例代码并观察结果：

- 一 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）。
- 一 获取‘STK_HC32M120_LQFP48_050_V11’评估板。
- 一 从华大半导体网站下载 HC32M120 DDL 代码。
- 一 下载并运行 timera\timera_phase_difference\中的项目文件：
 - 1) 打开 timera_phase_difference\项目，并打开‘main.c’如下视图：



- 2) 点击  重新编译整个项目；
- 3) 点击  将代码下载到评估板上，全速运行；
- 4) 配置函数发生器，通道 A、B 输出频率（1000HZ）、占空比相同，将函数发生器输出通道分别连接到 P75（CLKA）和 P74（CLKB）引脚；
- 5) 修改函数发生器相位配置，B 通道的相位超前 A 通道 1/4 周期，观察 LED_R 闪烁；
- 6) 修改函数发生器相位配置，B 通道的相位滞后 A 通道 1/4 周期，观察 LED_G 闪烁。

5 总结

以上章节简要介绍 HC32M120 系列的 TIMERA 寄存器、功能模式。演示了如何操作 TIMERA 正交编码计数样例代码，在开发中用户可以根据自己的实际需要使用 TIMERA 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

网址: <http://www.hdsc.com.cn/mcu.htm>

通信地址: 上海市浦东新区中科路 1867 号 A 座 10 层

邮编: 201203

