

## 32 位微控制器

### HC32M120 系列的内部时钟校准器 CTC

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>CTC 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 CTC .....</b>	<b>5</b>
3.1	校准原理 .....	5
3.2	寄存器介绍 .....	6
<b>4</b>	<b>样例代码 .....</b>	<b>7</b>
4.1	代码介绍 .....	7
4.2	工作流程 .....	10
4.3	代码运行 .....	11
<b>5</b>	<b>总结 .....</b>	<b>13</b>
<b>6</b>	<b>版本信息 &amp; 联系方式 .....</b>	<b>14</b>

## 1 摘要

本篇应用笔记主要介绍 HC32M120 系列内部时钟校准器（CTC）模块，并简要说明如何使用 CTC 校准内部时钟。

## 2 CTC 简介

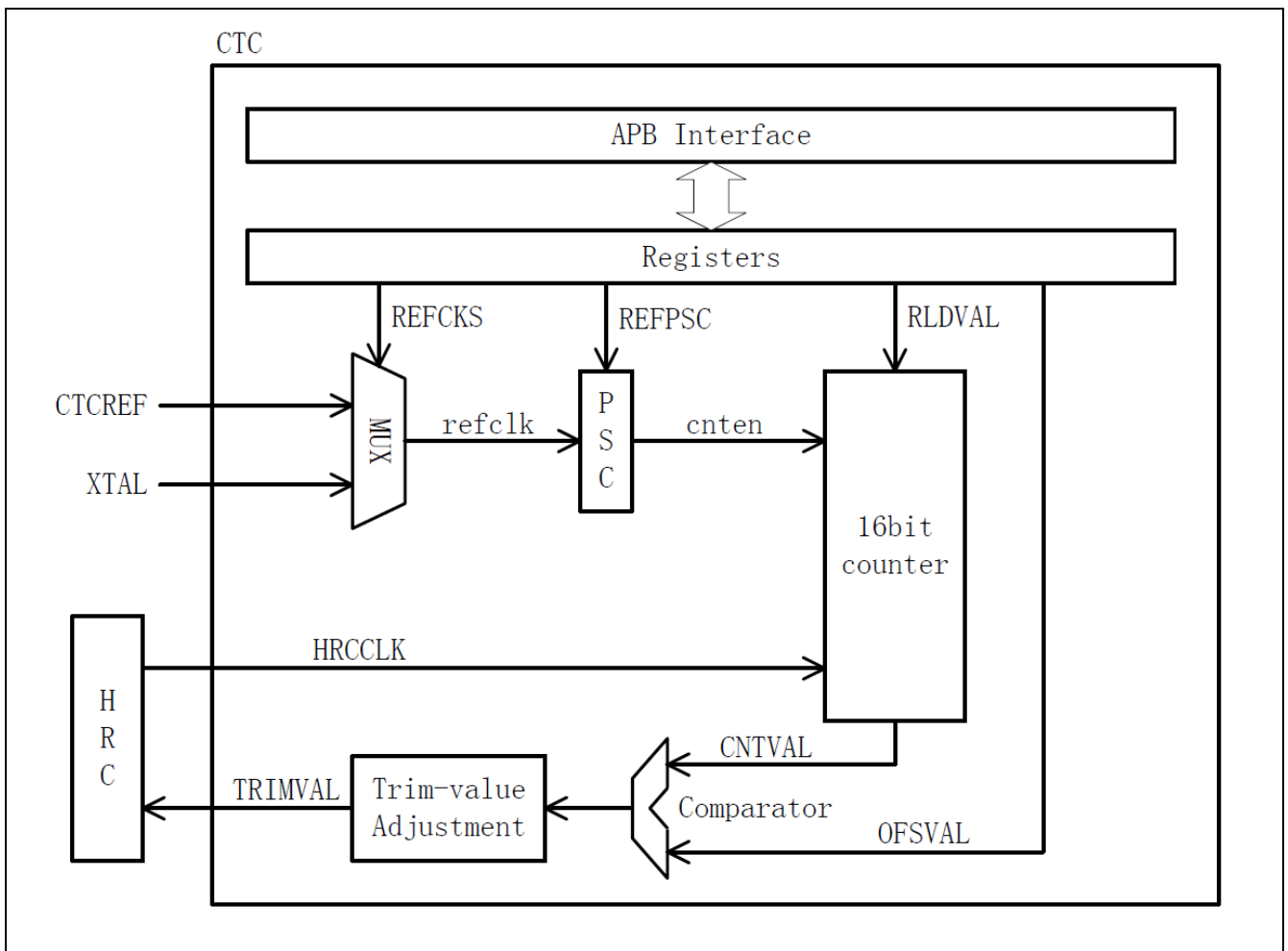
内部时钟校准器（Clock Trimming Controller, CTC）可以自动校准内部高速振荡器（HRC）。由于工作环境的影响 HRC 的频率可能会产生偏差，用 CTC 基于外部高精度参考时钟，采用硬件方式自动调整 HRC 的频率以得到一个精准的 HRC 时钟。

### 2.1 主要特性

- 2 个外部参考时钟源：XTAL 和 CTCREF
- 用于频率测量并具有重载功能的 16 位校准计数器；
- 用于频率校准的 8 位校准偏差值和 6 位校准值；
- 用于提示校准失败的错误中断；

## 2.2 基本框图

CTC 的基本框图如下图所示。



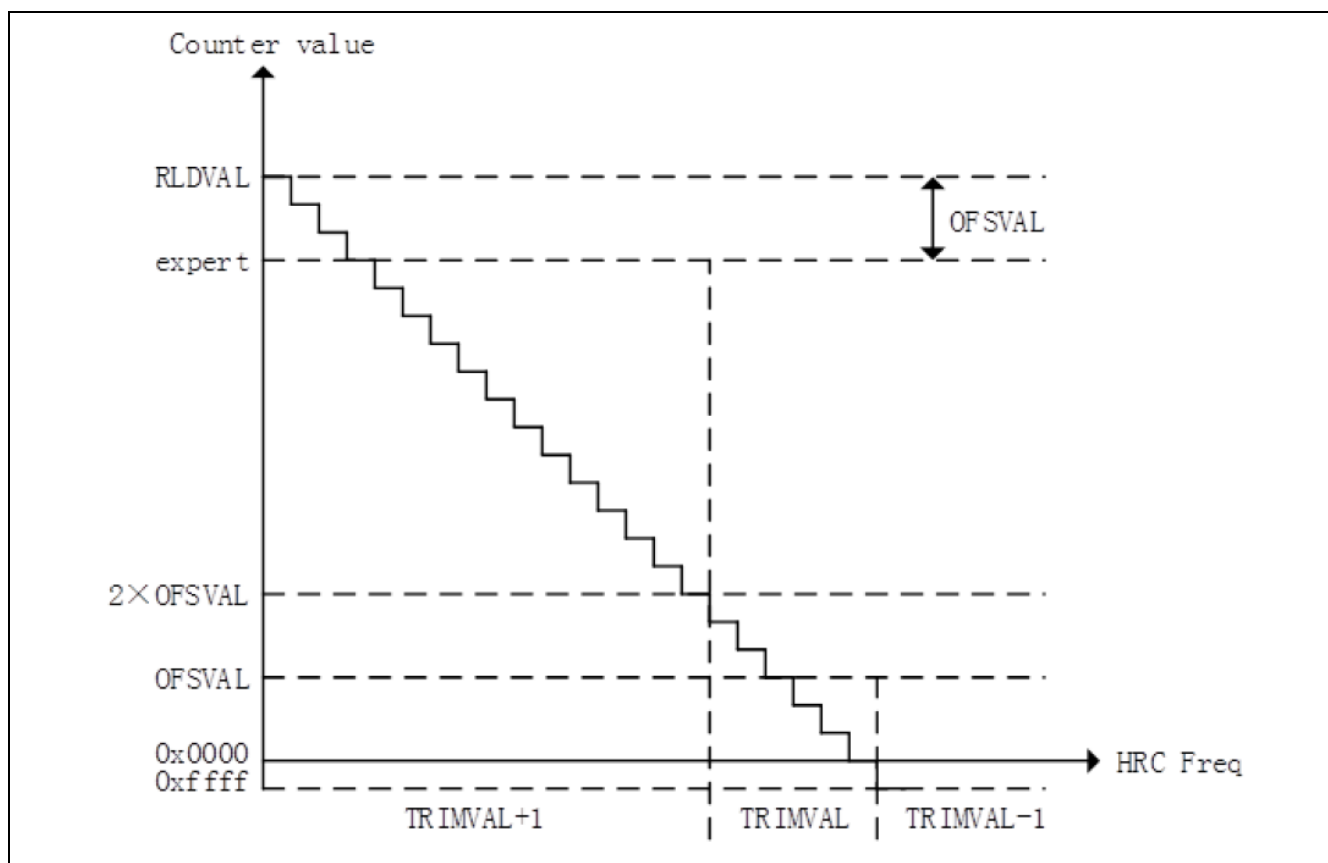
### 3 HC32M120 系列的 CTC

#### 3.1 校准原理

CTC 有 2 种外部时钟源可作为校准 HRC 频率的参考时钟，分别是外部高速时钟（XTAL）和外部基准时钟（CTCREF）。

在 CTC\_CR1 寄存器的 CTCEN 位置为 1 后，16 位校准计数器从 CTC\_CR2 寄存器中重载 RLDVAL 并向下计数，计数时钟由 HRC 提供。之后，如果检测到计数停止同步信号，16 位校准计数器将停止计数。如果始终检测不到计数停止同步信号，16 位校准计数器会持续计数直到下溢为 0xFFFF 时停止，并等待计数启动信号

CTC 校准示意图如下图所示：



- Counter > (OFSVAL × 2) 时停止计数；

16 位校准计数器的值大于 2×OFSVAL 且没有发生下溢，表示当前频率比期望频率慢，说明需增大 CTC\_CR1 寄存器中 TRMVAL[5:0] 的值来提高频率，CTC 会将 CTC\_CR1 寄存器中的 TRMVAL 的值自动加 1。

- $0 \leq \text{Counter} \leq (\text{OFSVAL} \times 2)$  时停止计数;

当前频率符合期望频率, CTC\_CR1 寄存器中的 TRMVAL 的值不变, CTC\_STR 寄存器中的 TRIMOK 位被置成 1。

- Counter = 0xFFFF;

16 位校准计数器的值发生下溢, 当前频率比期望频率快, 说明需要减小 TRMVAL 的值以降低频率, CTC 会将 CTC\_CR1 寄存器中的 TRMVAL 的值自动减 1。

## 3.2 寄存器介绍

内部时钟校准 CTC 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
CTC_CR1	时钟校准控制寄存器 1	0x00000000
CTC_CR2	时钟校准控制寄存器 2	0x00000000
CTC_STR	时钟校准状态寄存器	0x00000000

## 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 CTC 的校准功能。

以下部分主要基于 DDL 的 CTC 模块的输入捕获样例 `ctc_xtal_trimming` 代码，简要介绍 CTC 功能使用方法：

1) 配置时钟输出引脚：

```
/* Configure clock output system clock */
CLK_MCOConfig(CLK_MCOSOURCE_HRC, CLK_MCODIV_1);
/* Configure clock output pin */
GPIO_SetFunc(GPIO_PORT_1, GPIO_PIN_5, GPIO_FUNC_1_PULBUZ);
```

2) 配置系统时钟：

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

3) 初始化调试打印功能：

```
/* Initialize UART for debug print function. */
DDL_UartInit();
```

4) 配置 LED：

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

#### 5) 配置 CTC 功能:

```
/* Enable peripheral clock */
CLK_FcgPeriphClockCmd(FUNCTION_CLK_GATE, Enable);

/* Wait CTC stop. */
while (CTC_GetFlag(CTC_FLAG_BUSY))
{
    ;
}

/* Initialize CTC function. */
CTC_Init(&m_stcCtcInit);

/* Register CTC error IRQ handler && configure NVIC. */
stcIrqRegiConf.enIRQn = CTC_ERR_IRQn;
stcIrqRegiConf.enIntSrc = INT_CTC_ERR;
stcIrqRegiConf.pfnCallback = CtcErrIrqCallback;
INTC_IrqRegistration(&stcIrqRegiConf);
NVIC_ClearPendingIRQ(stcIrqRegiConf.enIRQn);
NVIC_SetPriority(stcIrqRegiConf.enIRQn, DDL_IRQ_PRIORITY_03);
NVIC_EnableIRQ(stcIrqRegiConf.enIRQn);

CTC_ErrIntCmd(Enable);
```

#### 6) 等待按键触发，启动 CTC 校准，提示校准结果:

```
/* User key : SW2 */
while (Pin_Reset != GPIO_ReadInputPortPin(KEY_PORT, KEY_PIN))
{
}

InitTrmVal = CTC_GetTrmVal();

CTC_Cmd(Enable);

while (!CTC_GetFlag(CTC_FLAG_TRMOK | CTC_FLAG_BUSY));
{
    ;
}
```

```
}

TrimmingTrmVal = CTC_GetTrmVal();

CTC_Cmd(Disable);

/* Wait CTC stop. */
while (CTC_GetFlag(CTC_FLAG_BUSY));
{
    ;
}
printf("Triming value is 0x%02X; Triming result is 0x%02X. \r\n", InitTrmVal, TrimmingTrmVal);

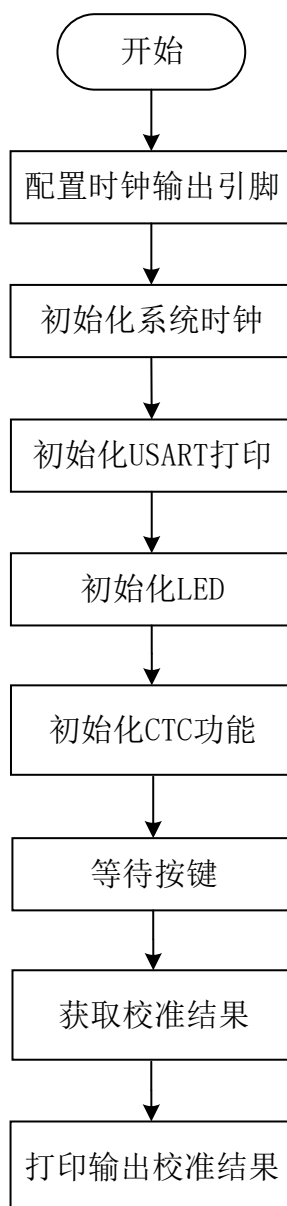
LED_G_ON();
```

7) CTC 错误中断回调函数：点亮红色 LED 灯，提示校准错误产生

```
/**
 * @brief CTC error IRQ callback.
 * @param None.
 * @retval None
 */
static void CtcErrIrqCallback(void)
{
    LED_R_ON();
}
```

## 4.2 工作流程

样例代码中 CTC 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（`ctc_xtal_trimming`），并配合评估板（‘`STK_HC32M120_LQFP48_050_V11`’）运行相关代码学习使用 CTC 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 `ctc_xtal_trimming` 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 HC32M120 DDL 代码。
- 下载并运行 `ctc\ctc_xtal_trimming` 中的项目文件：
  - 1) 将引脚 J6-P15 与示波器相连，该引脚输出内部时钟信号；
  - 2) 连接 DAP 模块 J9,J10；
  - 3) 打开串口助手软件，配置端口如下参数；

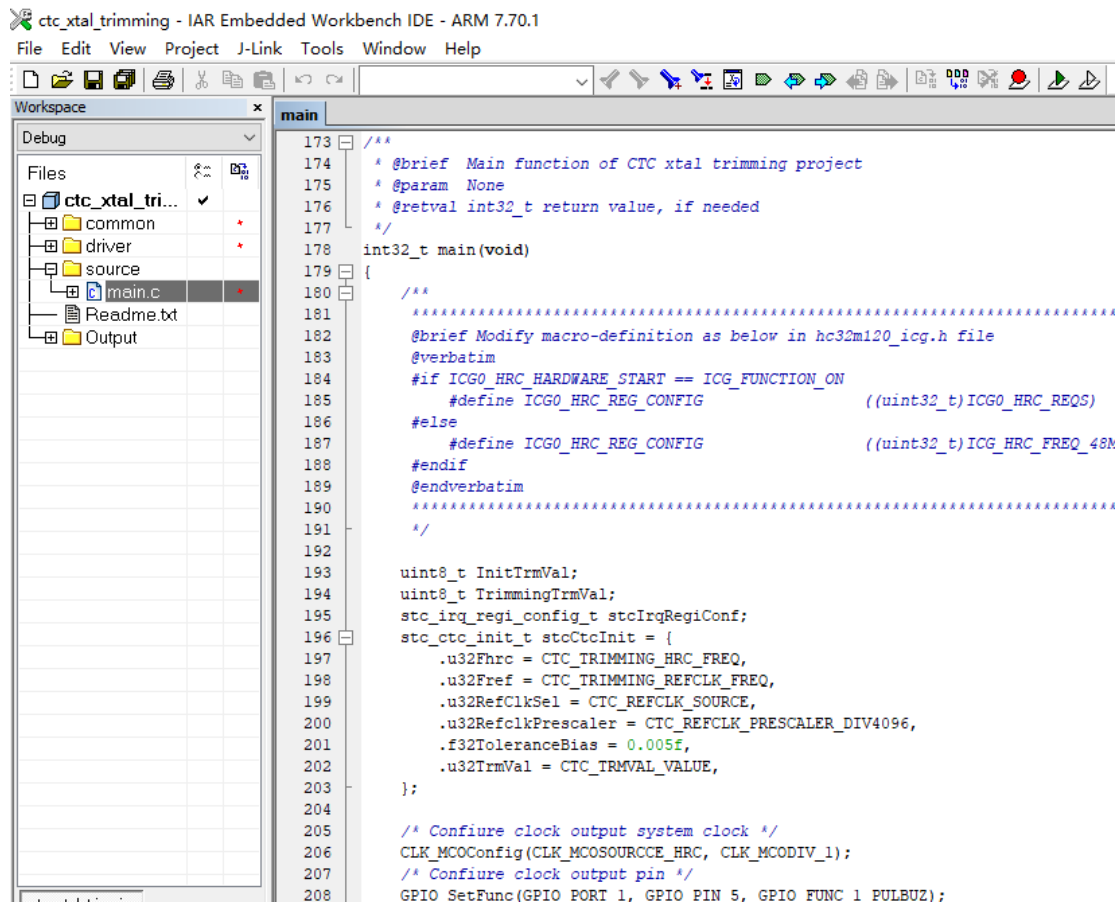
波特率：115200

数据位：8

校验位：None

停止位：1

4) 打开 ctc\_xtal\_trimming\项目，并打开 ‘main.c’ 如下视图：



5) 修改文件 hc32m120\_icg.h 如下宏定义：


```
#if ICG0_HRC_HARDWARE_START == ICG_FUNCTION_ON


#define ICG0_HRC_REG_CONFIG      ((uint32_t)ICG0_HRC_REQS)

#else

#define ICG0_HRC_REG_CONFIG      ((uint32_t)ICG_HRC_FREQ_48MHZ)

#endif
```

6) 点击  重新编译整个项目；

7) 点击  将代码下载到评估板上，全速运行；

8) 按下 SW2 按键；

9) 观察测试板绿色 LED 和示波器波形，绿色 LED 灯点亮，表示校准成功。

## 5 总结

以上章节简要介绍 HC32M120 系列的 CTC 功能。演示了如何操作 CTC 校准样例代码，在开发中用户可以根据自己的实际需要使用 CTC 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

---

