

32 位微控制器

HC32L110 / HC32F003 / HC32F005 系列的 Base Timer

适用对象

系列	产品型号
HC32L110	HC32L110C6UA
	HC32L110C6PA
	HC32L110C4UA
	HC32L110C4PA
	HC32L110B6PA
	HC32L110B4PA
HC32F003	HC32F003C4UA
	HC32F003C4PA
HC32F005	HC32F005C6UA
	HC32F005C6PA
	HC32F005D6UA

目 录

1	摘要	3
2	Timer 简介	3
3	Base Timer	4
	3.1 简介	4
	3.2 说明	5
	3.2.1 寄存器介绍	5
	3.2.2 工作流程介绍	5
4	样例代码	6
	4.1 代码介绍	6
	4.2 代码运行	8
5	总结	10
6	版本信息 & 联系方式	11

1 摘要

本篇应用笔记主要介绍 HC32L110 / HC32F003 / HC32F005 系列的 Base Timer 的功能及使用方法。

2 Timer 简介

什么是 Timer?

Timer，定时器，功能是在指定的时间间隔内反复触发指定窗口的定时器事件。

（引自‘百度百科’，‘互动百科’，‘维基百科’）

Timer 基本原理?

定时器一般通过对已知输入时钟信号的脉冲个数进行计数或测量时间，每个定时器内部结构实际上就是一个可编程的加法计数器，外加一些控制逻辑，并由编程来设置其工作模式及状态。

Timer 的应用?

定时器，一种极为常见的外部设备，可以用来测量时间间隔，用于在特定时间产生事件或确定两个事件之间的时间间隔，也可用来对外部事件进行计数，产生特定的波形等。

3 Base Timer

3.1 简介

小华 HC32L110 / HC32F003 / HC32F005 系列 MCU 的 Base timer 包含三个定时器 TIM0/1/2。TIM0/1/2 功能完全相同。TIM0/1/2 是同步定时/计数器，可以作为 16 位自动重载功能的定时/计数器，也可以作为 32 位无重载功能的定时/计数器。TIM0/1/2 可以对外部脉冲进行计数或者实现系统定时。

本模块功能包括：

- 计数功能

用于测定某个事件发生的次数，输入信号被内部的 Pclk 采样。

- 定时功能

用于产生间隔定时，定时器在每个时钟累加一次，计数到最大值产生溢出中断。

- Buzzer 功能

通过定时器的翻转输出功能可以实现驱动 Buzzer 的功能。

- 互联功能

通过内部互联可实现 UART 波特率的自动识别，测量 VC 比较输出的宽度，实现外部控制计数等。

3.2 说明

在该部分将主要介绍 HC32L110 / HC32F003 / HC32F005 系列的 Base Timer，包括寄存器和工作流程。

3.2.1 寄存器介绍

对于 Base Timer 模块的操作主要通过以下寄存器进行：

缩写	寄存器名称
TIMx_ARR	TIM0/1/2重载寄存器
TIMx_CNT	TIM0/1/2 16位模式计数寄存器
TIMx_CNT32	TIM0/1/2 32位模式计数寄存器
TIMx_CR	TIM0/1/2控制寄存器
TIMx_IFR	TIM0/1/2中断标志
TIMx_ICLR	TIM0/1/2中断清除寄存器

3.2.2 工作流程介绍

Base Timer 统一工作流程：

1. 配置计数时钟、工作模式、门控等；
2. 设定计数初值或比较值，重载值；
3. 启动定时器；
4. 等待中断并处理结果（如需要）；
5. 停止定时器。

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，也可以直接通过小华半导体的网站下载到 BT（Base Timer）的样例代码直接使用 BT 驱动库提供的 API 函数进行编码及校验的应用。

以下部分简要介绍该代码的各个部分的功能：

1) BT 数据声明及初始化：

```
//BT TEST DATA INIT
stc_bt_config_t stcConfig;
en_result_t      enResult = Error;
uint32_t         u32InitCntData = 0xFFFF0000;
```

2) 中断及初始化配置编码：

```
//INT ENABLE
EnableNvic(TIM0_IRQn, 3, TRUE);
Bt_EnableIrq(TIM0);

//对外部ext0计数，GPIO配置
Gpio_SetFunc_EXT0_P34();

stcConfig.enGateP = BtPositive;
stcConfig.enGate  = BtGateDisable;
stcConfig.enPRS   = BtPCLKDiv8;
stcConfig.enTog   = TogDisable;
stcConfig.enCT    = BtCounter;
stcConfig.enMD    = BtModel1;

stcConfig.pfnTim0Cb = Bt0Int;

if (Ok != Bt_Init(TIM0, &stcConfig))
{
    enResult = Error;
}

Bt_Cnt32Set(TIM0, u32InitCntData);
```

3) BT 启动运行：

```
Bt_Run(TIM0);
```

4) BT 等待中断后运行停止及结果处理:

```
//进入中断.....  
while(1)  
{  
    if (0x01 == u32BtTestFlag)  
    {  
        u32BtTestFlag = (u32BtTestFlag & (~0x01));  
        Bt_Stop(TIM0);  
        enResult = Ok;  
        break;  
    }  
}
```

通过以上代码即可完成使用 BT 的 TIM0 对外部 32K clk 的计数功能操作。

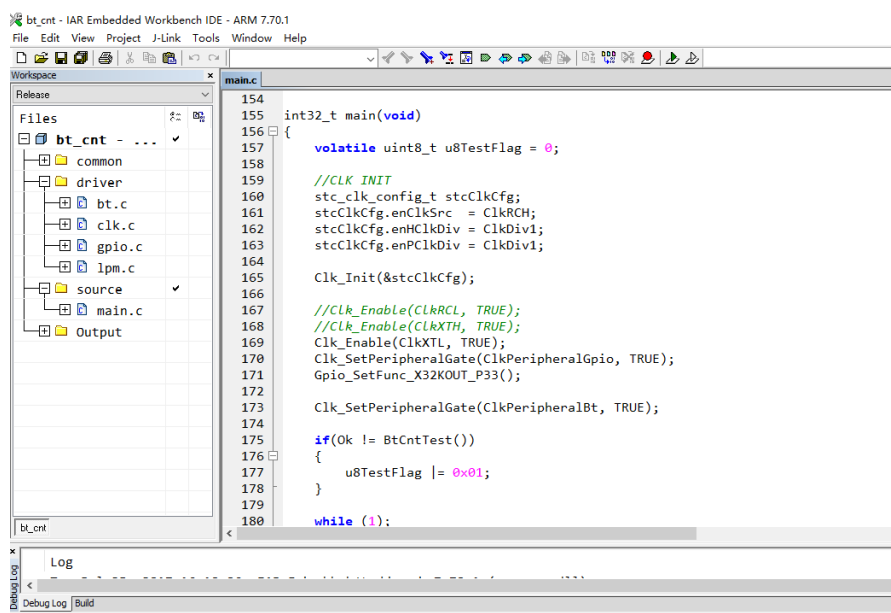
4.2 代码运行

用户可以通过小华半导体的网站下载到 BT 样例代码，并配合评估用板运行相关代码学习使用 BT 模块。

以下部分主要介绍如何在评估板上运行 BT 样例代码并观察结果：

- 确认安装正确的 IAR（或 Keil,此处使用 IAR 做样例说明，Keil 中操作方法类似）工具（请从小华半导体完整下载相应的安装包，并参考用户手册进行安装）。
- 从小华半导体网站下载 BT 样例代码。
- 下载并运行样例代码：

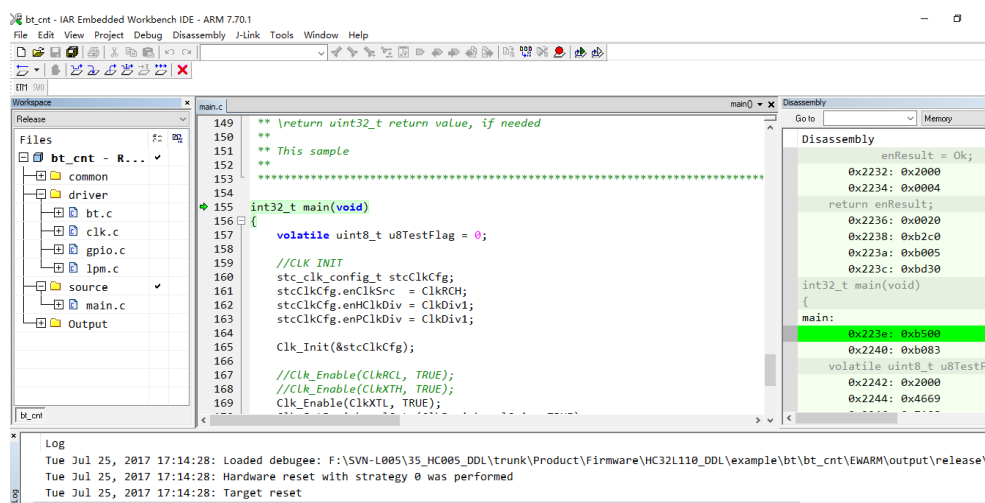
1) 打开 bt_cnt 工程，并打开‘main.c’如下视图：



2) 点击  重新编译链接整个项目。

3) 点击  将代码下载到评估板上。


4) 可以看见类似如下的视图:



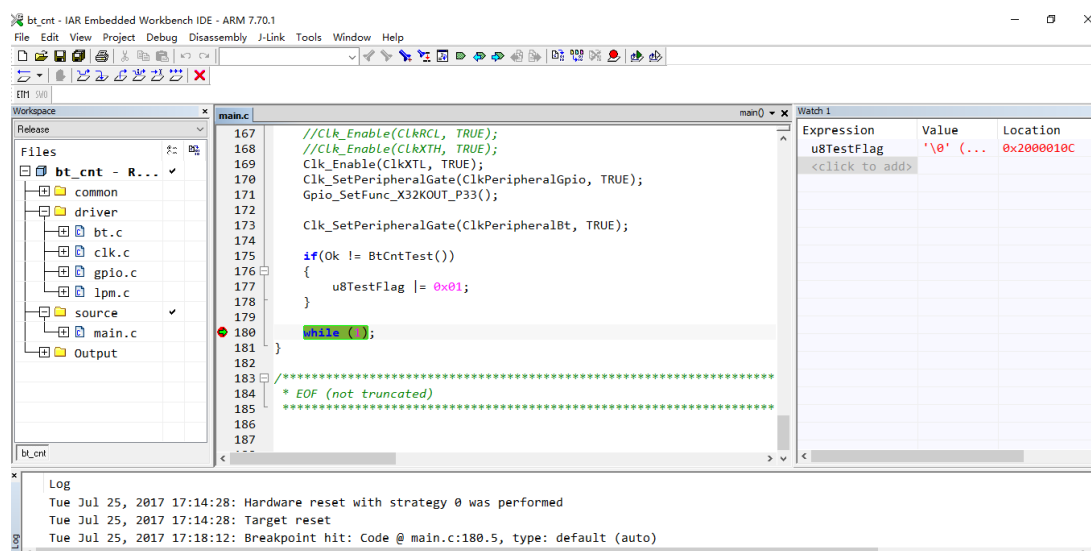
5) 在‘main(void)’的后一行设置断点:

6) 该样例需要在 Demo 板上用跳线连接 P33 与 P34 端口。

7) 点击“View -> Watch -> Watch1”打开一个‘watch1’窗口,并添加‘u8TestFlag’变量来观测其数值。

8) 点击  运行。

9) 代码运行并会停止在‘main(void)’的断点处,如果‘u8TestFlag = 0’,表示编码及校验功能正确执行,如下图:



10) 运行完毕后可以关闭项目文件。

11) 用户亦可通过其它样例或自己研究学习 Base Timer 的定时、Buzzer 等功能。

5 总结

以上章节简要介绍了 HC32L110 / HC32F003 / HC32F005 系列的 Base Timer，并详细说明了 Base Timer 模块的寄存器及操作流程，演示了如何使用相关的样例代码进行操作使用，在实际开发中用户可以根据自己的需要配置和使用 Base Timer 的其它功能。

6 版本信息 & 联系方式

日期	版本	修改记录
2018/6/4	Rev1.0	初版发布
2022/7/15	Rev1.1	公司 Logo 更新。



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@xhsc.com.cn

网址: <http://www.xhsc.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

