

32 位 微控制器

HC32F146/HC32M140 系列的 CSV

适用对象

系列	产品型号
HC32F146	HC32F146F8TA
	HC32F146J8UA
	HC32F146J8TA
	HC32F146KATA
HC32M140	HC32M140F8TA
	HC32M140J8UA
	HC32M140J8TA
	HC32M140KATA

目 录

1	摘要	3
2	HC32F146 / HC32M140 系列的 CSV	3
2.1	简介.....	3
2.2	说明.....	3
2.2.1	功能介绍	3
2.2.2	工作流程介绍	4
2.2.3	寄存器介绍	7
3	样例代码	8
3.1	代码介绍.....	8
3.2	代码运行.....	9
4	总结	11
5	版本信息	12

1 摘要

本篇应用笔记主要介绍如何使用 HC32F146 / HC32M140 系列芯片的时钟监视器（CSV）。

2 HC32F146 / HC32M140 系列的 CSV

2.1 简介

HC32F146 / HC32M140 内部具有时钟监测单元，可以根据内部 RC 振荡器生成的时钟来监视外部时钟的异常，具有当检测出外部振荡时钟故障（时钟停止）时，产生复位，当检测出外部频率异常时，产生中断或者复位。

2.2 说明

2.2.1 功能介绍

时钟监视器包括以下两类功能：

1) 时钟失效检测（CFD）功能

此功能检测外部高速时钟（HSXT）和外部低速时钟（LSXT）。当一定时间内（不可配置）检测不到 HSXT 或者 LSXT 的上升沿时，则认为时钟已经失效，此时产生系统复位。

2) 时钟频率异常检测（AFD）功能

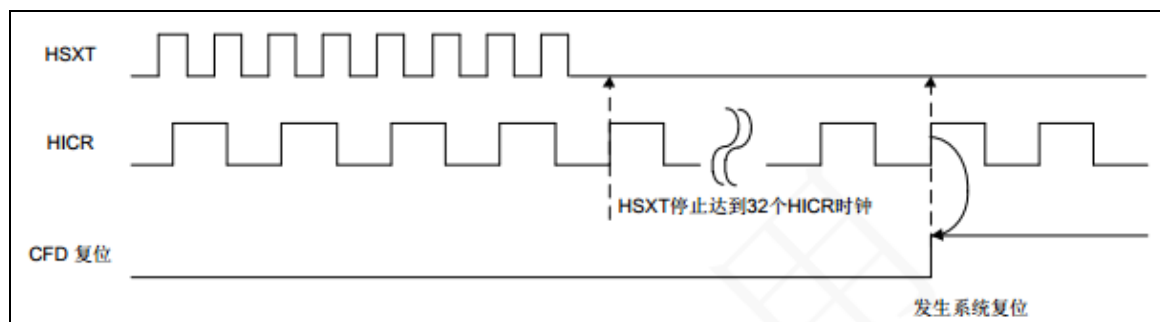
此功能检测外部高速时钟（HSXT）。当 HSXT 的实际频率超出设定的范围后，产生系统复位或中断（可以由寄存器选择）。

2.2.2 工作流程介绍

1) 时钟失效检测

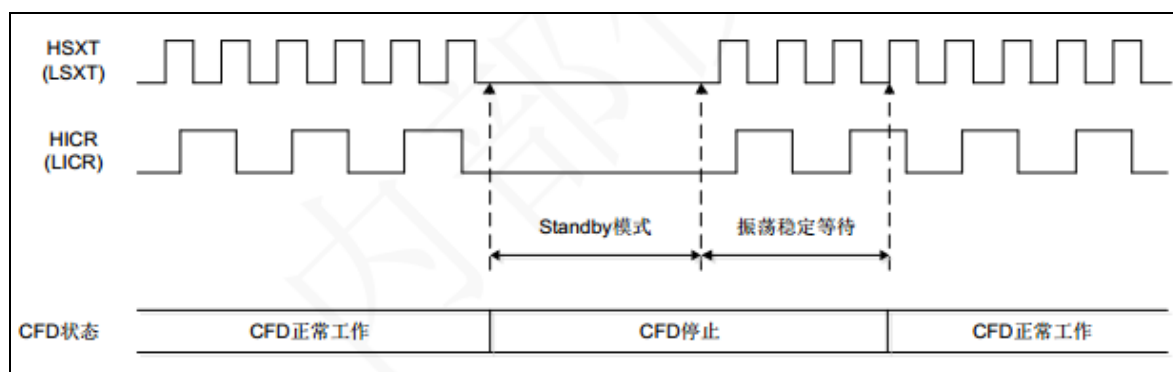
在一定的时间内，如果被检测时钟没有上升沿，则认为该时钟已经失效，此时产生系统复位（亦称 CFD 复位）。

外部高速时钟失效举例如下图所示：



当外部高速时钟停止正当的时间达到 32 个 HICR 时钟后，产生 CFD 复位或者中断。

时钟失效检测功能在 standby 模式下的状态如下：

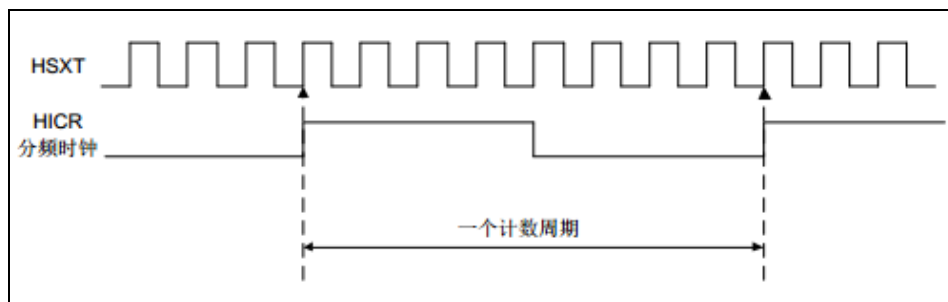


在 standby 模式下，HSXT 和 LICR 都停止振荡，同时 CFD 功能也会自动停止，退出 standby 后，HSXT 和 HICR 开始震荡并启动时钟振荡等待技术，等待稳定结束后，CFD 功能打开。

2) 异常频率检测

i. 工作原理

时钟异常检测的对象是外部高速时钟，以外部高速时钟为计数时钟，计算内部高速 RC 分频时钟的两个上升沿。如果计数值超出设定的范围，则认为被检测时钟频率异常，此时产生系统中断或者复位。产生的复位称为 AFD 复位，产生的中断称为 AFD 中断。工作原理图如下：



注意：

- HSXT 在一个计数周期内的计数值与 HICR 精度有关。因此，设置频率检测窗口寄存器时需考虑 HICR 的精度（参考数据手册）。

计算方法如下：

一个计数周期内的计数值表达式：

$$\text{计数值} = \frac{1}{\text{HICR分频时钟频率} \times (1 \pm \frac{\text{HICR精度}}{100})} \times \text{HSXT频率}$$

根据以上公式，可以得到计数值的上限 B 和下限 A。

设置频率检测窗口寄存器时，窗口下限应比 A 小，窗口上限应比 B 大。

计算举例：

假设 HSXT 频率是 4MHz，HICR 频率是 4MHz，HICR 精度为±2%，计数周期为 1024 分频（寄存器 FCM_CTL 的 AFDCC 默认值）。

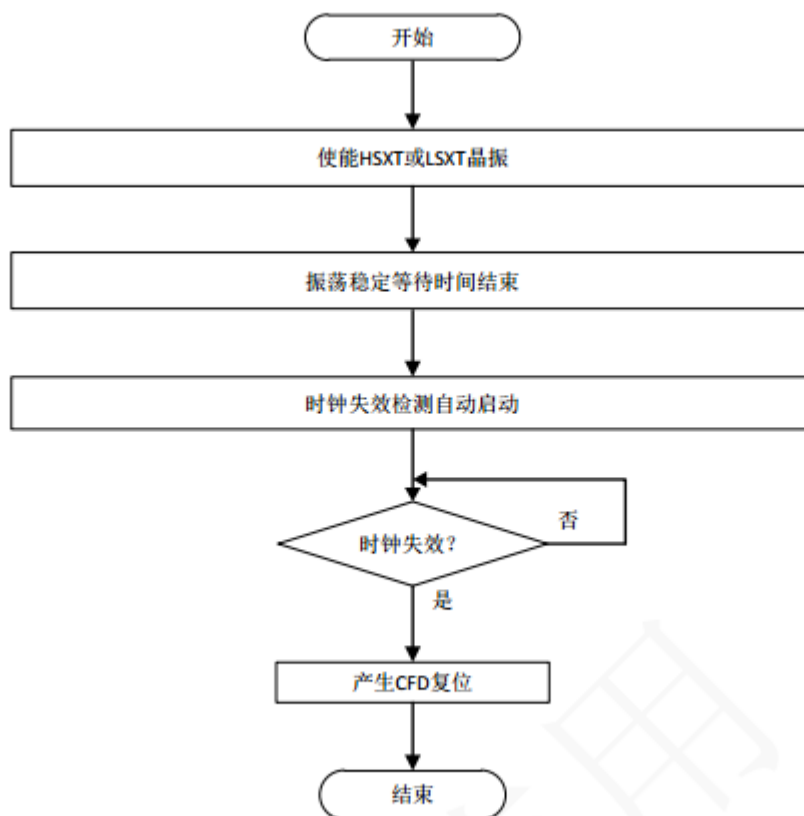
$$\begin{aligned} \text{计数值 A} &= \frac{1}{\frac{4 \times 10^6}{1024} \times (1 + \frac{2}{100})} \times 4 \times 10^6 \approx 1004 \\ \text{计数值 B} &= \frac{1}{\frac{4 \times 10^6}{1024} \times (1 - \frac{2}{100})} \times 4 \times 10^6 \approx 1045 \end{aligned}$$

如果平率检测的范围是 5%，则窗口值应设为：

$$\text{窗口下限} = 1004 \times (1-5\%) = 953.8$$

$$\text{窗口上限} = 1045 \times (1+5\%) = 1097.25$$

ii. 设置步骤



2.2.3 寄存器介绍

1) FCM_CTL 时钟监视器控制寄存器

时钟监视器通用功能设置使能或者关闭。

2) FCM_STR 时钟监视器状态寄存器

可读取相关标志位，进行判断时钟是否失效。

3) AFDWH_CTL 频率检测窗口上限设置寄存器

设置检测窗口的上限值。

4) AFDWL_CTL 频率检测窗口下限设置寄存器

设置检测窗口的下限值。

5) AFDC_CTL 频率检测计数值寄存器

可以读取当前 AFD 的计数值。

3 样例代码

3.1 代码介绍

用户可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设备驱动库（Device Driver Library, DDL）的样例代码并使用其中的 CLK 的 Example 进行验证。

本文档中以 AFD 产生复位为例进行说明。

- 1) 定义 LVD 配置结构体：

```
stc_clk_csv_config_t stcCsvCfg;;
```

- 2) 配置时钟检测相关寄存器参数：

```
/* Initialize csv configuration structure */  
stcCsvCfg.enAfdDiv = ClkAfdDiv1024;  
stcCsvCfg.bAfdResetEn = TRUE;  
stcCsvCfg.bAfdEn = TRUE;
```

- 3) 设置检测窗口上限值和下限值：

```
Clk_SetAfdWindow(1024 - 100, 1024 + 100); /* normal window */
```

- 4) 初始化配置：

```
Clk_CsvInit(&stcCsvCfg)
```

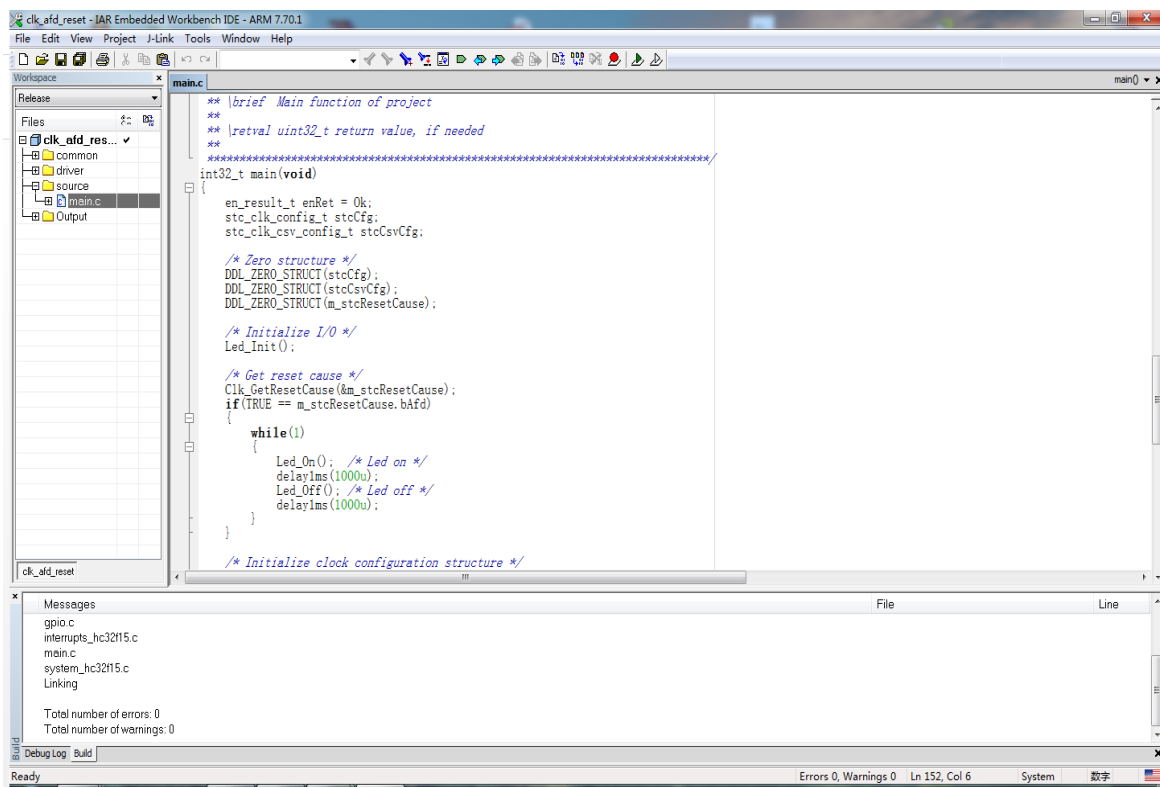

3.2 代码运行


用户可以通过华大半导体的网站下载到 CLK 的样例代码，并配合评估用板（SK-HC32F146-64L V10）运行相关代码学习使用 CLK 模块。


以下部分主要介绍如何在评估板上运行 CLK 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（请从华大半导体完整下载相应的安装包，并参考用户手册进行安装）。
- 从华大半导体网站下载 CLK 样例代码。
- 下载并运行样例代码：

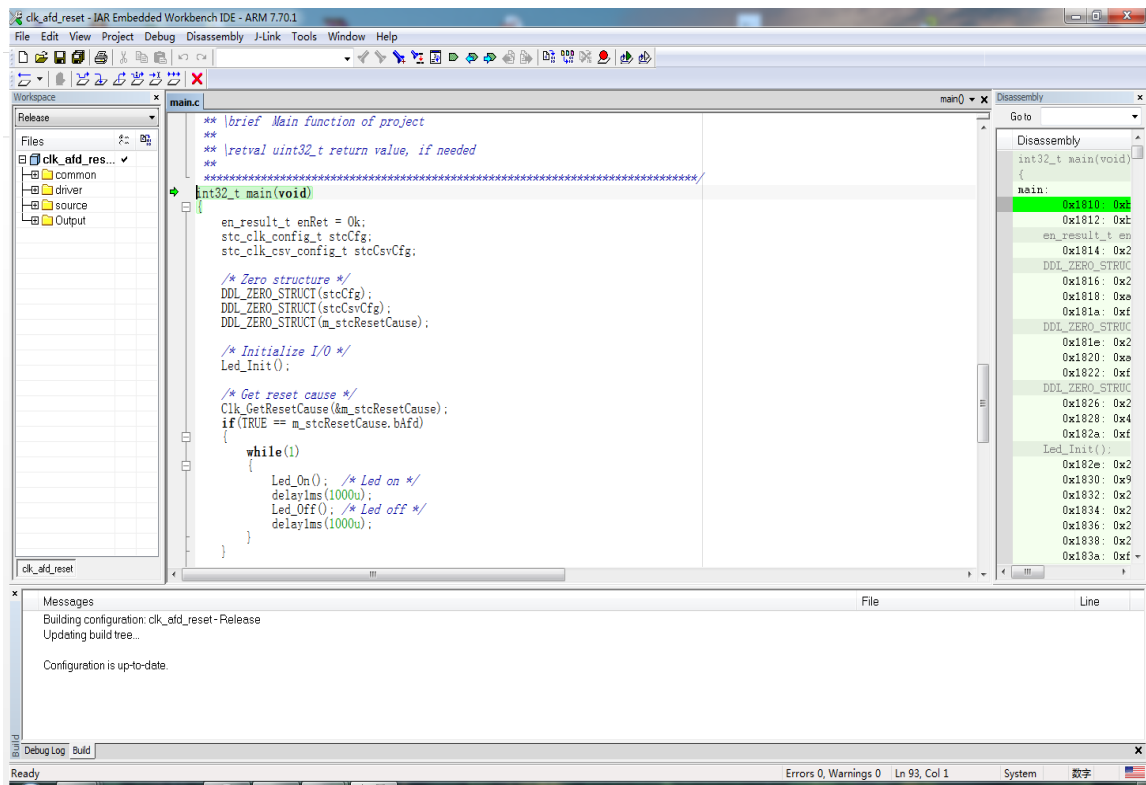
1) 打开 CLK-> clk_afd_reset 项目，并打开‘main.c’如下视图：



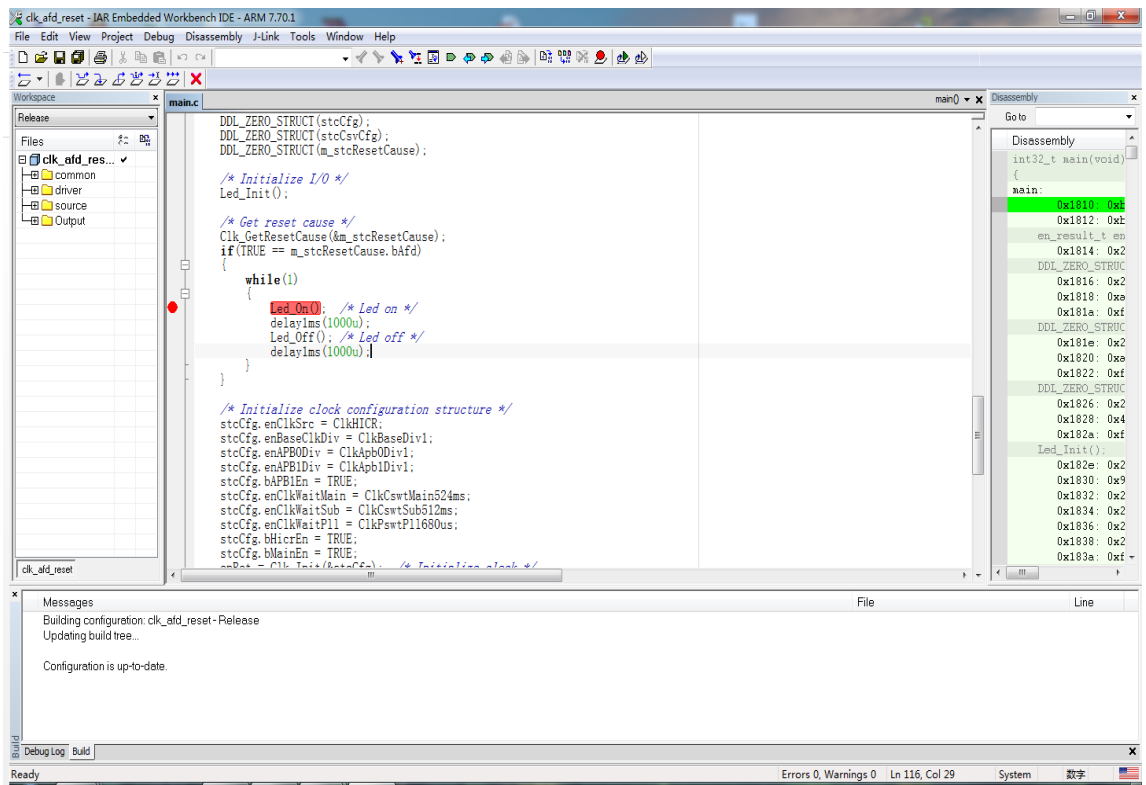
2) 点击  重新编译链接整个项目。

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) 当发生复位后程序会进入如图所示断点的代码中运行:



6) 点击  运行, 拔掉晶振, 会发现程序进入上图的位置中去。

- 7) 运行完毕后可以关闭项目文件。
- 8) 用户亦可通过修改代码中 **CLK** 的相关配置参数或初始化数据来进一步学习 **CLK** 模块的功能。

4 总结

以上章节简要介绍了 HC32F146 / HC32M140 系列的时钟检测功能，并详细说明了时钟监测模块的寄存器及操作流程，并且演示了如何使用 **CSV**，在实际开发中用户可以根据自己的需要配置和使用 **CSV**。

5 版本信息

日期	版本	修改记录
2017-11-23	Rev1.0	时钟监控器 CSV 应用笔记初版发布



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: 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

