

## 32 位微控制器

# HC32L110 / HC32F003 / HC32F005 系列的 CRC

### 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

<b>1</b>	<b>摘要 .....</b>	<b>3</b>
<b>2</b>	<b>CRC 简介 .....</b>	<b>3</b>
<b>3</b>	<b>CRC .....</b>	<b>5</b>
3.1	简介 .....	5
3.2	说明 .....	6
3.2.1	寄存器介绍 .....	6
3.2.2	工作流程介绍 .....	6
<b>4</b>	<b>样例代码 .....</b>	<b>7</b>
4.1	代码介绍 .....	7
4.2	代码运行 .....	8
<b>5</b>	<b>总结 .....</b>	<b>10</b>
<b>6</b>	<b>版本信息 &amp; 联系方式 .....</b>	<b>11</b>

## 1 摘要

本篇应用笔记主要介绍如何使用 HC32L110 / HC32F003 / HC32F005 系列的 CRC 功能进行编码及校验。

## 2 CRC 简介

### 什么是 CRC?

CRC 即循环冗余校验码 (Cyclic Redundancy Check)：是数据通信领域中最常用的一种查错校验码，其特征是信息字段和校验字段的长度可以任意选定。循环冗余检查 (CRC) 是一种数据传输检错功能，对数据进行多项式计算，并将得到的结果附在帧的后面，接收设备也执行类似的算法，以保证数据传输的正确性和完整性。

(引自‘百度百科’，‘互动百科’，‘维基百科’)

### CRC 基本原理?

循环冗余校验码 (CRC) 的基本原理是：在  $K$  位信息码后再拼接  $R$  位的校验码，整个编码长度为  $N$  位，因此，这种编码也叫  $(N, K)$  码。对于一个给定的  $(N, K)$  码，可以证明存在一个最高次幂为  $N-K=R$  的多项式  $G(x)$ 。根据  $G(x)$  可以生成  $K$  位信息的校验码，而  $G(x)$  叫做这个 CRC 码的生成多项式。校验码的具体生成过程为：假设要发送的信息用多项式  $C(x)$  表示，将  $C(x)$  左移  $R$  位 (可表示成  $C(x)*2^R$ )，这样  $C(x)$  的右边就会空出  $R$  位，这就是校验码的位置。用  $C(x)*2^R$  除以生成多项式  $G(x)$  得到的余数就是校验码。

任意一个由二进制位串组成的代码都可以和一个系数仅为‘0’和‘1’取值的多项式一一对应。例如：代码 1010111 对应的多项式为  $x^6+x^4+x^2+x+1$ ，而多项式为  $x^5+x^3+x^2+x+1$  对应的代码 101111。

### CRC 的应用?

CRC 校验实用程序库在数据存储和数据通讯领域，为了保证数据的正确，就不得不采用检错的手段。在诸多检错手段中，CRC 是最著名的一种。CRC 的全称是循环冗余校验，其特点是：检错能力极强，开销小，易于用编码器及检测电路实现。从其检错能力来看，它所不能发现的错误的几率仅为 0.0047% 以下。从性能上和开销上考虑，均远远优于奇偶校验及算术和校验等方式。因而，在数据存储和数据通讯领域，CRC 无处不在：著名的通讯协议 X.25 的 FCS(帧检

错序列)采用的是 CRC-CCITT, WinRAR、NERO、ARJ、LHA 等压缩工具软件采用的是 CRC32, 磁盘驱动器的读写采用了 CRC16, 通用的图像存储格式 GIF、TIFF 等也都用 CRC 作为检错手段。

华大MCU交流群:164973950

## 3 CRC

### 3.1 简介

华大 HC32L110 / HC32F003 / HC32F005 系列 MCU 的 CRC 模块算法遵从 ISO/IEC13239 的定义，采用 16 位长度的 CRC，计算多项式为  $x^{16} + x^{12} + x^5 + 1$ ，16 位初始值为“FFFF”。

本模块功能包括：

- 提供 CRC 编码生成和 CRC 编码校验
- 提供 8/16/32 位宽的小端访问

以对数据串 001122334455667788aabbccddeeff 计算 CRC 为例：

- 8 位模式输入顺序：00, 11, 22 ....
- 16 位模式输入顺序：0x1100, 0x3322, 0x5544, 0x7766
- 32 位模式输入顺序：0x33221100, 0x77665544

## 3.2 说明

本小节介绍 HC32L110 / HC32F003 / HC32F005 系列的 CRC 模块，包括寄存器和工作流程。

### 3.2.1 寄存器介绍

对于 CRC 模块的操作主要通过以下寄存器进行：

缩写	寄存器名称
CRC_RESULT	CRC 结果寄存器，计算完成后对该寄存器读取即获得结果。
CRC_DATA	CRC 数据寄存器，用于输入需要运算的数据

### 3.2.2 工作流程介绍

#### CRC 编码

CRC 编码产生就是对一串数据进行运算，产生 16 位的 CRC 编码结果。操作流程如下：

- 1) 将 CRC\_RESULT 寄存器赋初值 16'hFFFF。这可以通过直接写入 16'hFFFF 到 CRC\_RESULT[15:0]寄存器来实现，也可以通过复位来实现。
- 2) 将需要运算的数据依次写入 CRC\_DATA 寄存器，每次写操作对应输入 1 个数据（8 位、16 位或者 32 位）。比如，有 10 个数据，就对 CRC 数据寄存器写入 10 次，每次写入 1 个。  
CRC\_DATA 寄存器的地址是一个范围（0x80-0xFF），如果选择 32 位的数据传输方式，建议采用 STM 指令来加快数据输入，方法是选择本寄存器的一个地址（比如 8'h80），然后通过 STM 指令一次传送若干个数据到 CRC\_DATA 数据寄存器（注意 STM 指令中地址的变化，不要超出 CRC\_DATA 数据寄存器规定的地址范围）。
- 3) 在将所有需要运算的数据写入 CRC\_DATA 结果寄存器后，读取 CRC\_RESULT 结果寄存器的[15:0]可获得 16 位 CRC 编码。

#### CRC 校验

CRC 校验就是要对一串数据以及 16 位 CRC 编码进行判断，检验其是否正确。CRC 校验的操作与之前 CRC 编码产生的操作极为类似。只是在第三步之后再加上以下步骤：

- 1-3: 与 CRC 编码中 1-3 步骤相同。
- 4: 在将所有需要运算的数据写入 CRC\_DATA 结果寄存器后，再写入 16 位 CRC 编码。
- 5: 读取 CRC\_RESULT 结果寄存器[16]，如果为 1 则表示校验成功；如果为 0 则表示校验失败。

## 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 CRC 的样例代码直接使用 CRC 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) CRC 数据声明及初始化：

```
//CRC TEST DATA INIT
uint8_t au8CrcTestData[8] = {0x12, 0x34, 0x56, 0x78, 0x9a, 0xbc, 0xde, 0xf0};
uint32_t u32TestDataLen = 8;
uint16_t u16RefCrc16 = 0x5234;
```

#### 2) CRC 编码：

```
en_result_t CrcCodingTest(void)
{
    uint16_t u16CrcResult = 0;
    en_result_t enResult = Error;

    u16CrcResult = CRC16_Get8(au8CrcTestData, u32TestDataLen);
    if(u16RefCrc16 == u16CrcResult)
    {
        enResult = Ok;
    }

    return enResult;
}
```

#### 3) CRC 校验：

```
en_result_t CrcCheckTest(void)
{
    return CRC16_Check8(au8CrcTestData, u32TestDataLen, u16RefCrc16);
}
```

#### 4) CRC 编码及校验：

```
if(Ok != CrcCodingTest())
{
    u8TestFlag |= 0x01;
}
if(Ok != CrcCheckTest())
{
    u8TestFlag |= 0x02;
}
```

通过以上代码即可完成一次 CRC 的编码与校验。

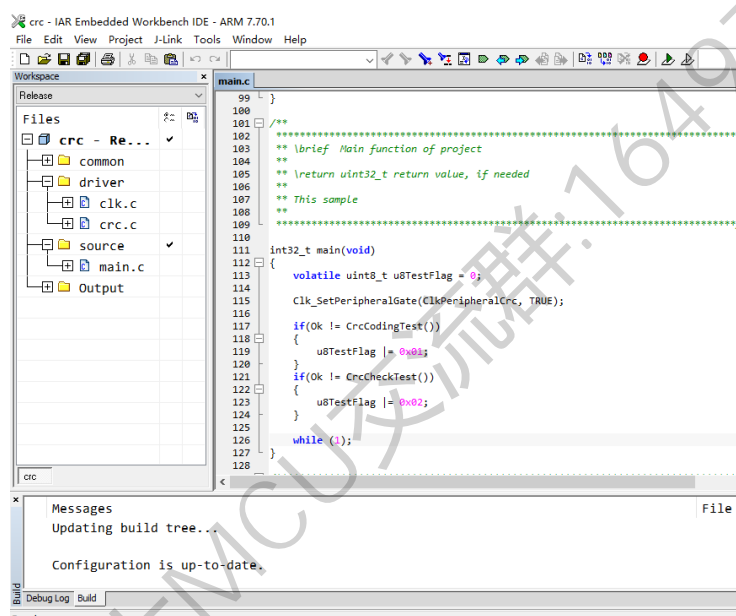
## 4.2 代码运行


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


以下部分主要介绍如何在评估板上运行 CRC 样例代码并观察结果：

- 一 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）。
- 一 从华大半导体网站下载 CRC 样例代码。
- 一 下载并运行样例代码：

1) 打开 CRC 工程，并打开 ‘main.c’ 如下视图：

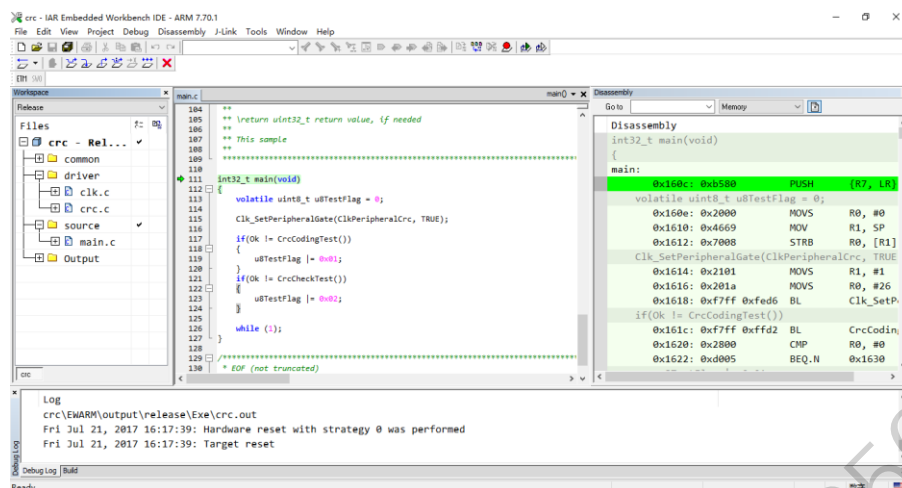


2) 点击  重新编译链接整个项目。

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:

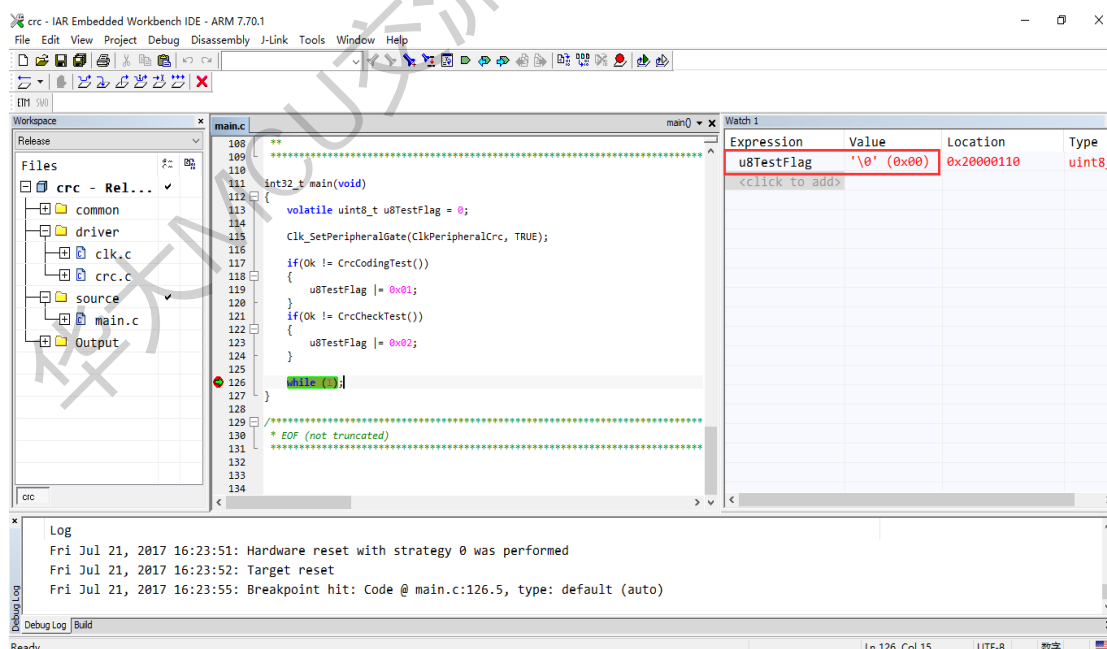


5) 在 ‘main(void)’ 的后一行设置断点:

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

7) 点击  运行。

8) 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: (本样例中的测试数据均来自标准 CRC 软件编码工具计算结果)。



9) 运行完毕后可以关闭项目文件。

10) 用户亦可通过修改代码中 CRC 的测试数据(数据及长度)来进一步学习 CRC 模块的功能。

## 5 总结

以上章节简要介绍了 HC32L110 / HC32F003 / HC32F005 系列的 CRC，并详细说明了 CRC 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编码及校验，在实际开发中用户可以根据自己的需要配置和使用 CRC 的编码及校验功能。

华大MCU交流群:164973950

## 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



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：上海市张江高科园区碧波路 572 弄 39 号

邮编：201203

