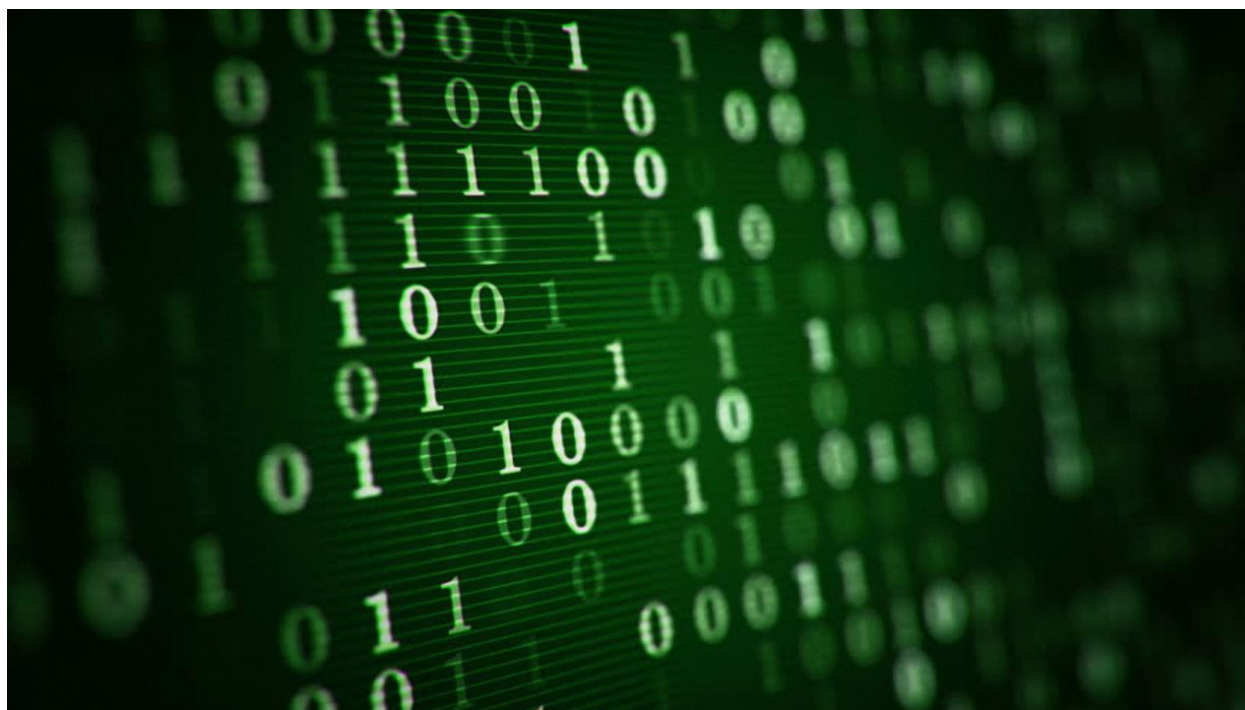


知乎

首发于  
电子电路开发学习

...

 写文章

## CRC校验原理及实现




王超

[关注他](#)

234 人赞同了该文章

### 目录

- 前言
- CRC算法简介

 赞同 234  26 条评论 分享 喜欢 收藏 申请转载

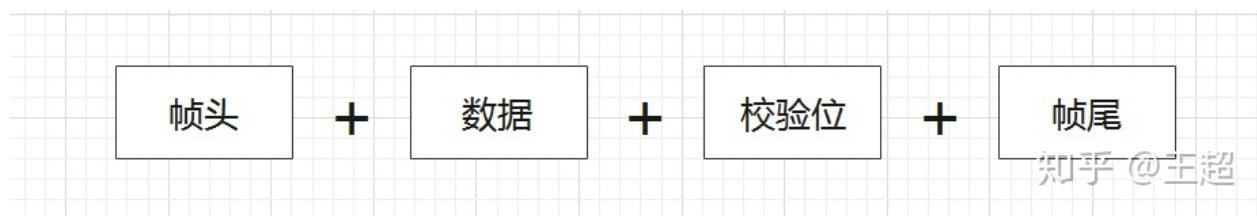
...

- CRC计算的C语言实现
- CRC计算工具
- 总结

## 前言

最近的工作中，要实现对通信数据的CRC计算，所以花了两天的时间好好研究了一下，周末有时间整理了一下笔记。

一个完整的数据帧通常由以下部分构成：



校验位是为了保证数据在传输过程中的完整性，采用一种指定的算法对原始数据进行计算，得出的一个校验值。接收方接收到数据时，采用同样的校验算法对原始数据进行计算，如果计算结果和接收到的**校验值一致**，说明数据校验正确，这一帧数据可以使用，如果不一致，说明传输过程中出现了差错，这一帧数据丢弃，请求重发。

常用的校验算法有奇偶校验、校验和、CRC，还有LRC、BCC等不常用的校验算法。

以串口通讯中的奇校验为例，如果数据中1的个数为奇数，则奇校验位0，否则为1。

例如原始数据为：0001 0011，数据中1的个数（或各位相加）为3，所以奇校验位为0。这种校验

▲ 赞同 234 ▼ ● 26 条评论 ↗ 分享 ❤ 喜欢 ★ 收藏 📄 申请转载 ...

误。

	数据	奇校验位
发送端	00 <b>1</b> 0011	0
接收端	00 <b>1</b> 00011	0
接收端	<b>11</b> 010011	0

校验和同理也会有类似的错误：

	数据	校验和
发送端	<b>01</b> 10 <b>08</b> 20	39
接收端	<b>08</b> 10 <b>01</b> 20	39

一个好的校验方法，配合数字信号编码方式，如(差分)曼彻斯特编码，(不)归零码等对数据进行编码，可大大提高通信的健壮性和稳定性。例如以太网中使用的是CRC-32校验，曼彻斯特编码方式。本篇文章介绍CRC校验的原理和实现方法。

▲ 赞同 234 ▼ ● 26 条评论 ↗ 分享 ❤ 喜欢 ★ 收藏 📄 申请转载 ...

循环冗余校验（Cyclic Redundancy Check，CRC）是一种根据网络数据包或计算机文件等数据产生简短固定位数校验码的一种信道编码技术，主要用来检测或校验数据传输或者保存后可能出现的错误。它是利用除法及余数的原理来作错误侦测的。

CRC校验计算速度快，检错能力强，易于用编码器等硬件电路实现。从检错的正确率与速度、成本等方面，都比奇偶校验等校验方式具有优势。因而，CRC 成为计算机信息通信领域最为普遍的校验方式。常见应用有以太网/USB通信，压缩解压，视频编码，图像存储，磁盘读写等。

## CRC参数模型

不知道你是否遇到过这种情况，同样的CRC多项式，调用不同的CRC计算函数，得到的结果却不一样，而且和手算的结果也不一样，这就涉及到CRC的参数模型了。计算一个正确的CRC值，需要知道CRC的参数模型。

一个完整的CRC参数模型应该包含以下信息：WIDTH，POLY，INIT，REFIN，REFOUT，XOROUT。

- NAME：参数模型名称。
- WIDTH：宽度，即生成的CRC数据位宽，如CRC-8，生成的CRC为8位
- POLY：十六进制多项式，省略最高位1，如  $x^8 + x^2 + x + 1$ ，二进制为1 0000 0111，省略最高位1，转换为十六进制为0x07。
- INIT：CRC初始值，和WIDTH位宽一致。
- REFIN：true或false，在进行计算之前，原始数据是否翻转，如原始数据：0x34 = 0011 0100，如果REFIN为true，进行翻转之后为0010 1100 = 0x2c
- REFOUT：true或false，运算完成之后，得到的CRC值是否进行翻转，如计算得到的CRC值：0x97 = 1001 0111，如果REFOUT为true，进行翻转之后为11101001 = 0xE9。
- XOROUT：计算结果与此参数进行异或运算后得到最终的CRC值，和WIDTH位宽一致。

常用的21个标准CRC参数模型：

CRC算法名称	多项式公式	WIDTH	POLY	INIT	XOROUT	REFIN	REFOUT
CRC-4/ITU	$x^4 + x + 1$	4	03	00	00	TRUE	TRUE
CRC-5/EPC	$x^5 + x^3 + 1$	5	09	09	00	FALSE	FALSE
CRC-5/ITU	$x^5 + x^4 + x^2 + 1$	5	15	00	00	TRUE	TRUE
CRC-5/USB	$x^5 + x^2 + 1$	5	05	1F	1F	TRUE	TRUE
CRC-6/ITU	$x^6 + x + 1$	6	03	00	00	TRUE	TRUE
CRC-7/MMC	$x^7 + x^3 + 1$	7	09	00	00	FALSE	FALSE
CRC-8	$x^8 + x^2 + x + 1$	8	07	00	00	FALSE	FALSE
CRC-8/ITU	$x^8 + x^2 + x + 1$	8	07	00	55	FALSE	FALSE
CRC-8/ROHC	$x^8 + x^2 + x + 1$	8	07	FF	00	TRUE	TRUE
CRC-8/MAXIM	$x^8 + x^5 + x^4 + 1$	8	31	00	00	TRUE	TRUE
CRC-16/IBM	$x^{16} + x^{15} + x^2 + 1$	16	8005	0000	0000	TRUE	TRUE
CRC-16/MAXIM	$x^{16} + x^{15} + x^2 + 1$	16	8005	0000	FFFF	TRUE	TRUE
CRC-16/USB	$x^{16} + x^{15} + x^2 + 1$	16	8005	FFFF	FFFF	TRUE	TRUE
CRC-16/MODBUS	$x^{16} + x^{15} + x^2 + 1$	16	8005	FFFF	0000	TRUE	TRUE
CRC-16/CCITT	$x^{16} + x^{12} + x^5 + 1$	16	1021	0000	0000	TRUE	TRUE
CRC-16/CCITT-FALSE	$x^{16} + x^{12} + x^5 + 1$	16	1021	FFFF	0000	FALSE	FALSE
CRC-16/X25	$x^{16} + x^{12} + x^5 + 1$	16	1021	FFFF	FFFF	TRUE	TRUE
CRC-16/XMODEM	$x^{16} + x^{12} + x^5 + 1$	16	1021	0000	0000	FALSE	FALSE
CRC-16/DNP	$x^{16} + x^{13} + x^{12} + x^{11} + x^{10} + x^8 + x^6 + x^5 + x^2 + 1$	16	3D65	0000	FFFF	TRUE	TRUE
CRC-32	$x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$	32	04C11DB7	FFFFFFFF	FFFFFFFF	TRUE	TRUE
CRC-32/MPEG-2	$x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$	32	04C11DB7	FFFFFFFF	0	FALSE	FALSE

CRC校验在电子通信领域非常常用，可以说有通信存在的地方，就有CRC校验：

- 美信(MAXIM)的芯片DS2401/DS18B20，都是使用的CRC-8/MAXIM模型
- SD卡或MMC使用的是CRC-7/MMC模型
- Modbus通信使用的是CRC-16/MODBUS参数模型
- USB协议中使用的CRC-5/USB和CRC-16/USB模型
- STM32自带的硬件CRC计算模块使用的是CRC-32模型

至于多项式的选择，初始值和异或值的选择，输入输出是否翻转，这就涉及到一定的编码和数学知识了。感兴趣的朋友，可以了解一下每个CRC模型各个参数的来源。至于每种参数模型的检错能力、重复率，需要专业的数学计算了，不在本文讨论的范畴内。

▲ 赞同 234 ▼ ● 26 条评论 ↗ 分享 ❤ 喜欢 ★ 收藏 📄 申请转载 ...

## CRC计算

好了，了解了CRC参数模型知识，下面手算一个CRC值，来了解CRC计算的原理。

**问：原始数据：0x34，使用CRC-8/MAXIN参数模型，求CRC值？**

**答：**根据CRC参数模型表，得到CRC-8/MAXIN的参数如下：

```
POLY = 0x31 = 0011 0001(最高位1已经省略)
INIT = 0x00
XOROUT = 0x00
REFIN = TRUE
REFOUT = TRUE
```

有了上面的参数，这样计算条件才算完整，下面来实际计算：

- 0.原始数据 = 0x34 = 0011 0100，多项式 = 0x31 = 1 0011 0001
- 1.INIT = 00，原始数据高8位和初始值进行异或运算保持不变。
- 2.REFIN为TRUE，需要先对原始数据进行翻转：0011 0100 > 0010 1100
- 3.原始数据左移8位，即后面补8个0：0010 1100 0000 0000
- 4.把处理之后的数据和多项式进行模2除法，求得余数：  
原始数据：0010 1100 0000 0000 = 10 1100 0000 0000  
多项式：1 0011 0001  
模2除法取余数低8位：1111 1011
- 5.与XOROUT进行异或，1111 1011 xor 0000 0000 = 1111 1011
- 6.因为REFOUT为TRUE，对结果进行翻转得到最终的CRC-8值：1101 1111 = 0xDF
- 7.数据+CRC：0011 0100 1101 1111 = 34DF，相当于原始数据左移8位+余数。

▲ 赞同 234 ▼ ● 26 条评论 ↗ 分享 ❤ 喜欢 ★ 收藏 📄 申请转载 ...



Handwritten long division of 100110001 by 101011 to calculate CRC-8. The remainder is 01111011.

101011

100110001

10110000000000

100110001

101000100

100110001

111010100

100110001

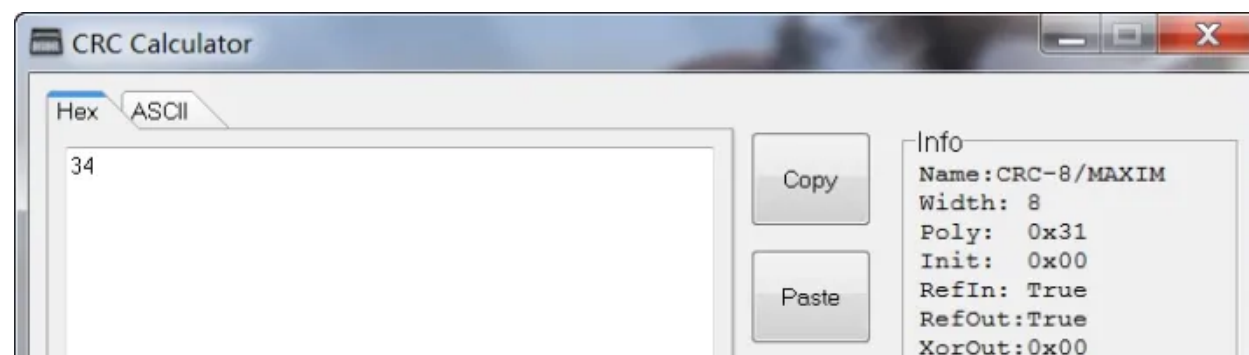
111001010

100110001

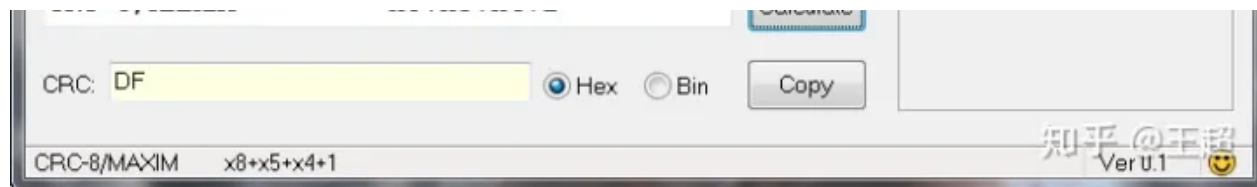
01111011

知乎 @王超

验证手算结果：



▲ 赞同 234 ▼ ● 26 条评论 ↗ 分享 ♥ 喜欢 ★ 收藏 📄 申请转载 ...



可以看出是一致的，当你手算的结果和工具计算结果不一致时，可以看看 INIT, XOROUT, REFINT, REFOUT 这些参数是否一致，有 1 个参数不对，计算出的 CRC 结果都不一样。

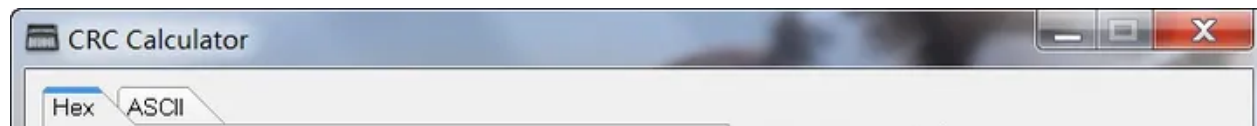
## CRC 校验

上面通过笔算的方式，讲解了 CRC 计算的原理，下面来介绍一下如何进行校验。

按照上面 CRC 计算的结果，最终的数据帧：0011 0100 1101 1111 = 34DF，前 8 位 0011 0100 是原始数据，后 8 位 1101 1111 是 CRC 结果。

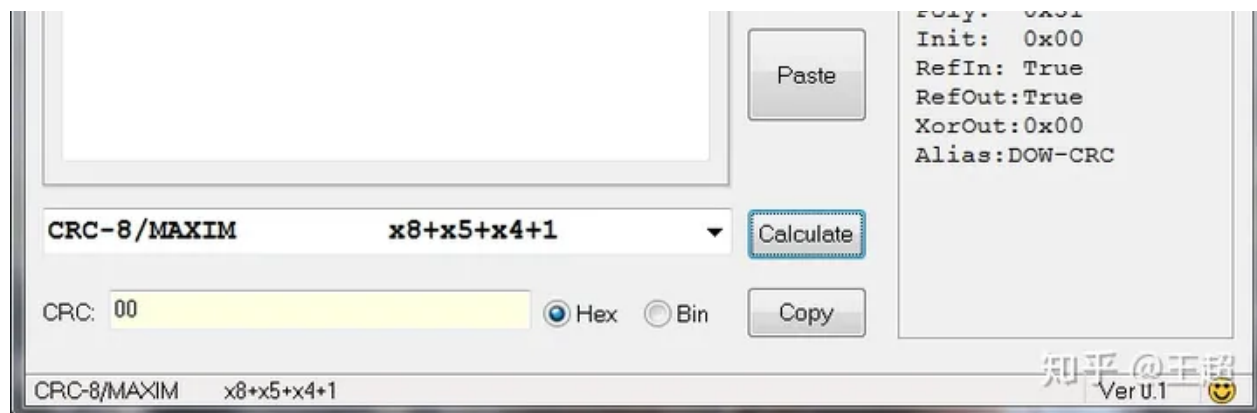
接收端的校验有两种方式，一种是和 CRC 计算一样，在本地把接收到的数据和 CRC 分离，然后在本地对数据进行 CRC 运算，得到的 CRC 值和接收到的 CRC 进行比较，如果一致，说明数据接收正确，如果不一致，说明数据有错误。

另一种方法是把整个数据帧进行 CRC 运算，因为是数据帧相当于把原始数据左移 8 位，然后加上余数，如果直接对整个数据帧进行 CRC 运算（除以多项式），那么余数应该为 0，如果不为 0 说明数据出错。



▲ 赞同 234 ▼ ● 26 条评论 ↗ 分享 ❤ 喜欢 ★ 收藏 📄 申请转载 ...





而且，不同位出错，余数也不同，可以证明，余数与出错位数的对应关系只与CRC参数模型有关，而与原始数据无关。

### CRC计算的C语言实现

无论是用C还是其他语言，实现方法网上很多，这里我找了一个基于C语言的CRC计算库，里面包含了常用的21个CRC参数模型计算函数，可以直接使用，只有 `crcLib.c` 和 `crcLib.h` 两个文件。

GitHub地址：[github.com/whik/crc-lib...](https://github.com/whik/crc-lib...)

使用方法非常简单：

```
#include <stdio.h>
#include <stdlib.h>
#include "crcLib.h"
```

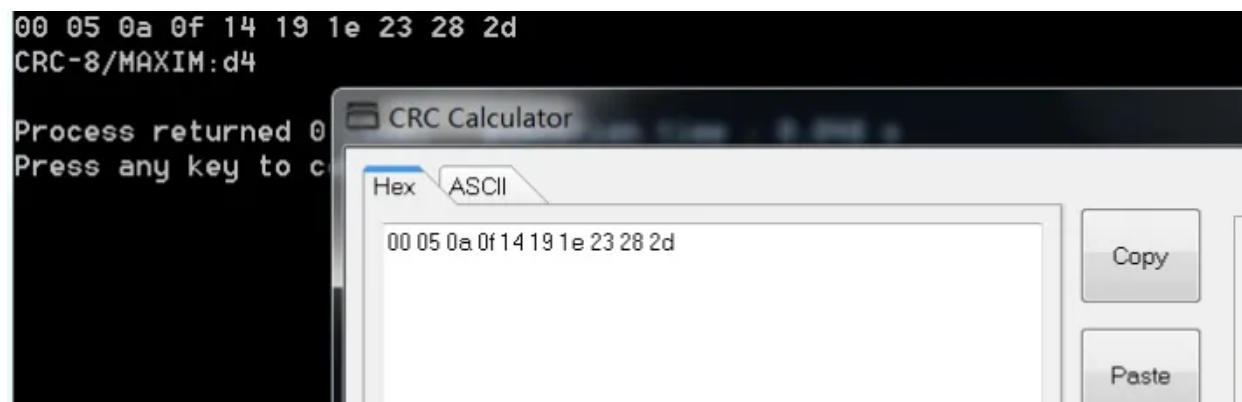
```
uint8_t LENGTH = 10;
uint8_t data[LENGTH];
uint8_t crc;

for(int i = 0; i < LENGTH; i++)
{
    data[i] = i*5;
    printf("%02x ", data[i]);
}
printf("\n");

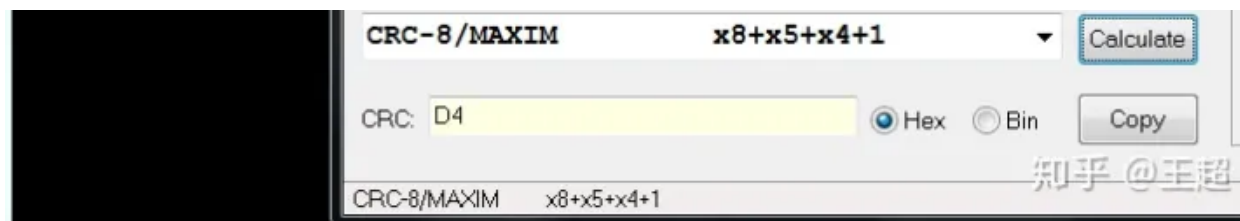
crc = crc8_maxim(data, LENGTH);

printf("CRC-8/MAXIM:%02x\n", crc);
return 0;
}
```

计算结果：



▲ 赞同 234 ▼ ● 26 条评论 ↗ 分享 ❤ 喜欢 ★ 收藏 📄 申请转载 ...



## CRC计算工具

下面这几款工具都可以自定义CRC算法模型，而且都有标准CRC模型可供选择。如果自己用C语言或者Verilog实现校验算法时，非常适合作为标准答案进行验证。

- 在线计算：[ip33.com/crc.html](http://ip33.com/crc.html)
- 离线计算工具：CRC\_Calc v0.1.exe或者GCRC.exe

格西CRC计算器：



▲ 赞同 234 ▼ ● 26 条评论 ↗ 分享 ♥ 喜欢 ★ 收藏 📄 申请转载 ...



知乎 @王超

## 总结

CRC校验并不能100%的检查出数据的错误，非常低的概率会出现CRC校验正确但数据中有错误位的情况。这和CRC的位数，多项式的选择等等有很大的关系，所以在实际使用中尽量选择标准CRC参数模型，这些多项式参数都是经过理论计算得出的，可以提高CRC的检错能力。CRC校验可以检错，也可以纠正单一比特的错误，你知道纠错的原理吗？

## 参考资料

- [cnblogs.com/liushui-sky...](http://cnblogs.com/liushui-sky...)
- [segmentfault.com/a/1190...](http://segmentfault.com/a/1190...)

发布于 2020-09-23 21:25

CRC CRC校验 C/C++



评论千万条，友善第一条

26 条评论

默认 最新



曙神

博主讲的很好，博主讲和这些都很解了

赞同 234

26 条评论

分享

喜欢

收藏

申请转载

...

**知乎用户ShLdH8**

...

3.原始数据左移8位，即后面补8个0：0010 1100 0000 0000

请教下：CRC8,16,32的计算中，是否都要对原值进行如此左移以补充足16位的操作？

2022-07-09

回复 2

**然莫之能御**

...

我实际验证了一下CRC16-MODBUS，在仿真环境中可以观察到每一步的计算结果，结果是CRC16-MODBUS的校验方式，仍然只需左移8位即可。我之前手算的时候按左移16位补的，结果一直不对，仿真调试才发现问题。

2022-11-30

回复 赞

**freeneuro**

...

请问CRC校验位数与数据位数有关系吗？比如说在使用8位校验位的情况下，对32位数据的检错能力会比64位数据的检错能力更强吗？（懂CRC计算，不懂背后的数学原理 - -！）

2021-02-25

回复 1

**zhmkw**

...

你好，请问你现在懂它俩之间的关系了吗？

2022-03-26

回复 赞

**zhmkw**

...

找到了个资料，它们之间有个关系。[CRC校验（一）--原理 - 菜鸟学院](#)

2022-03-26

回复 赞

**淡淡**

...

极化码看了这个，去理解了模二除法才看懂，很神奇

2021-01-26

回复 1

**王超** 作者

...

极化码是什么呀

赞同 234 26 条评论 分享 喜欢 收藏 申请转载 ...

**王超** 作者

...

应该是软件自动计算完成之后组包发送的吧？接收数据也是自动校验吧

2020-09-24

回复 1

**Bruce**

...

是的

2020-09-25

回复 赞

**徐瑞**

...

为啥翻转我看不懂啊，为啥34反转后是2c啊，翻转不等于取反吗？

04-23

回复 赞

**小白**

...

相当于倒序😎

05-08

回复 赞

**向生**

...

请问一下，这个怎么进行联调呢？

02-20

回复 赞

**开水**

...

“1.INIT = 00，原始数据高8位和初始值进行异或运算保持不变。”这句话是什么意思呢？

2022-11-11

回复 赞

**哼哼哈嘿**

...

第二种方法的结果校验：(34DF)0011 0100 1101 1111/(31)100110001,为什么手算出来的结果就是不为0，我用工具直接计算34DF的结果是0，请指教一下我是不是忽略什么细节？求指点，快哭了

2022-09-26

回复 赞

赞同 234 26 条评论 分享 喜欢 收藏 申请转载 ...

的。

2022-09-26

回复 3



哼哼哈嘿

...

我分别将34和DF的二进制数分别翻转后成为0010 1100 1111 1011之后就能计算为0了，不是拿结果直接模二除法吗？？难道这个翻转规则是一半一半的翻转0011 0100 1101 1111--0010 1100 1111 1011, 而不是整体翻转成1111 1011 0010 1100? 这个真的不懂，刚接触这个快哭了

2022-09-26

回复 赞



qazwsx

...

你好，我想请问一下，初始值和Crc校验码的初态是一回事吗？我看标准里的初始值要么是0000，要么是Ffff，这是固定的吗？

2022-09-22

回复 赞



王超 作者

...

是一个，初始值是0或F

2022-09-24

回复 赞



不知为不知

...

你那个手算求余数的是不是求错了？已经被借一位的0做减法的话从高位借一位为2-1=1，然后1-0=1，但是你那都是连续几个0

2022-08-30

回复 赞



不知为不知

...

.....我意识到自己错了，那是异或运算，不是减法运算.....尴尬😓😓😓

2022-08-30

回复 赞



敬风

...

参数（REFIN: true或false，在进行计算之前，原始数据是否翻转）咱写的时候可以完整点不（REFIN: true或false，在进行计算之前，原始数据 每个字节 是否翻转），代码写出来试了好几次就是不对😓😓

赞同 234 26 条评论 分享 喜欢 收藏 申请转载 ...



**王超** 作者

...

好吧，谢谢提醒，是每个字节啊？

2022-05-20

回复 赞

**我什么都不懂奥**

...

请问为啥原始数据要左移8位呢？这个有什么原则吗？

2022-05-04

回复 赞

**开水**

...

CRC是8位数据，总的传输数据 = 原始数据+8位CRC，低八位是CRC数据，所以原始数据需要左移八位；

2022-11-11

回复 赞

**Bruce**

...

MThings提供的自动组包功能，让调试者不用再关心检验数据计算，自动运算CRC16/LRC，并自动组包

2020-09-24

回复 赞



评论千万条，友善第一条

**电子电路开发学习**

公众号：电子电路开发学习，同步更新。

## 推荐阅读

赞同 234 26 条评论 分享 喜欢 收藏 申请转载 ...

## CRC校验原理及实现

随着数据采集系统的功能日益强大，以及微型计算机的普及，在现代工业中，利用微机进行数据通讯的工业控制应用得也越来越广泛。特别是在大规模高精度数据采集系统中，对数据进行分析 and 计算将…

TIAP

发表于程序员

```
1 unsigned short crc_calculate_crc(unsigned char *data,unsigned int size,const unsigned short crc_poly,
2 unsigned short init_value,unsigned char ref_flag)
3 {
4     unsigned short crc_reg = init_value,tmp = 0;
5     unsigned char j,byte = 0;
6     while (size--){
7         byte = *data++;
8         if (ref_flag)
9             invert_uint8(byte,&byte);
10        crc_reg ^= byte << 8;
11        for ( j = 0; j < 8; j++){
12            tmp = crc_reg & 0x000000ff;
13            crc_reg >>= 8;
14            if (tmp)
15                crc_reg ^= crc_poly;
16        }
17    }
18    if (ref_flag)
19        invert_uint16(crc_reg,&crc_reg);
20 }
```

## CRC校验原理及代码实现（一）

SeeDeer

## CRC-16校验原理与说明

1、循环校验码（CRC码）：是数据通信领域中最常用的一种差错校验码，其特征是信息字段和校验字段的长度可以任意选定。2、生成CRC码的基本原理：任意一个由二进制位串组成的代码都可以和一…

工控达人

## CRC码计算及校验原理的最通俗诠释

CRC校验原理 CRC校验原理看起来比较复杂，好难懂，因为大多数书上基本上是以二进制的多项式形式来说明的。其实很简单的问题，其根本思想就是先在要发送的帧后面附加一个数（这个就是用来校…

小太阳