

## CANOpen 系列教程 05

# CAN 总线同步与位填充、优先级的决定

作者: strongerHuang

申明: 该文档仅供个人学习使用

归类	CANOpen 系列教程
标签	CAN、 CANOpen、 CanFestival
网站	<a href="http://www.strongerhuang.com">http://www.strongerhuang.com</a>

版权所有: **禁止商用**

Copyright @2018 strongerHuang

## 目 录

一、 写在前面.....	3
二、 位填充.....	3
三、 优先级的决定.....	4
3.1 数据帧和遥控帧的优先级.....	4
3.2 标准格式和扩展格式的优先级.....	5
四、 其他.....	5
五、 说明.....	6
六、 最后.....	6

## 一、写在前面

继续为大家更新 CAN 总线协议相关内容，这些内容其实也比较重要，会直接影响到你后期 CAN 总线的编程，夸张一点的说会影响到你的软件架构。

本文内容相对来说比较容易理解，初学者有必要了解一下。

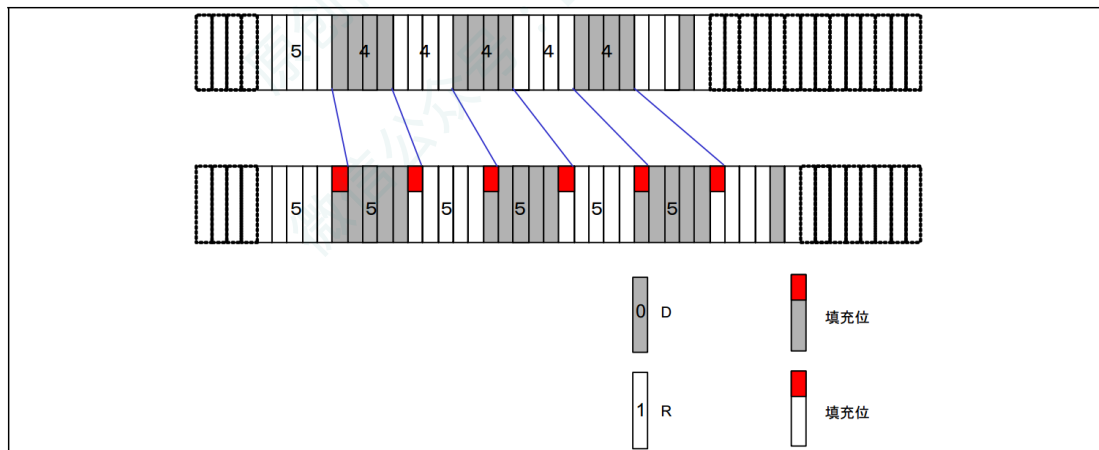
本文章收录于【[CANOpen 系列教程](#)】，在我的博客分类“CANOpen 系列教程”也能查找到。

为了方便大家平时公交、地铁、外出办事也能用手机随时随地查看该教程，该系列教程也同步更新于微信公众号【EmbeddedDevelop】，关注微信公众号回复【CANOpen 系列教程】即可查看。

## 二、位填充

位填充是为防止突发错误而设定的功能。当同样的电平持续 5 位时则添加一个位的反型数据。

这个功能很好理解，如下图所示。位填充的数据不需要编程控制，由控制器自动完成。其目的也是提高 CAN 总线的稳定性。



### (1) 发送单元的工作

在发送数据帧和遥控帧时，SOF~CRC 段间的数据，相同电平如果持续 5 位，在下一个位（第 6 个位）则要插入 1 位与前 5 位反型的电平。

### (2) 接收单元的工作

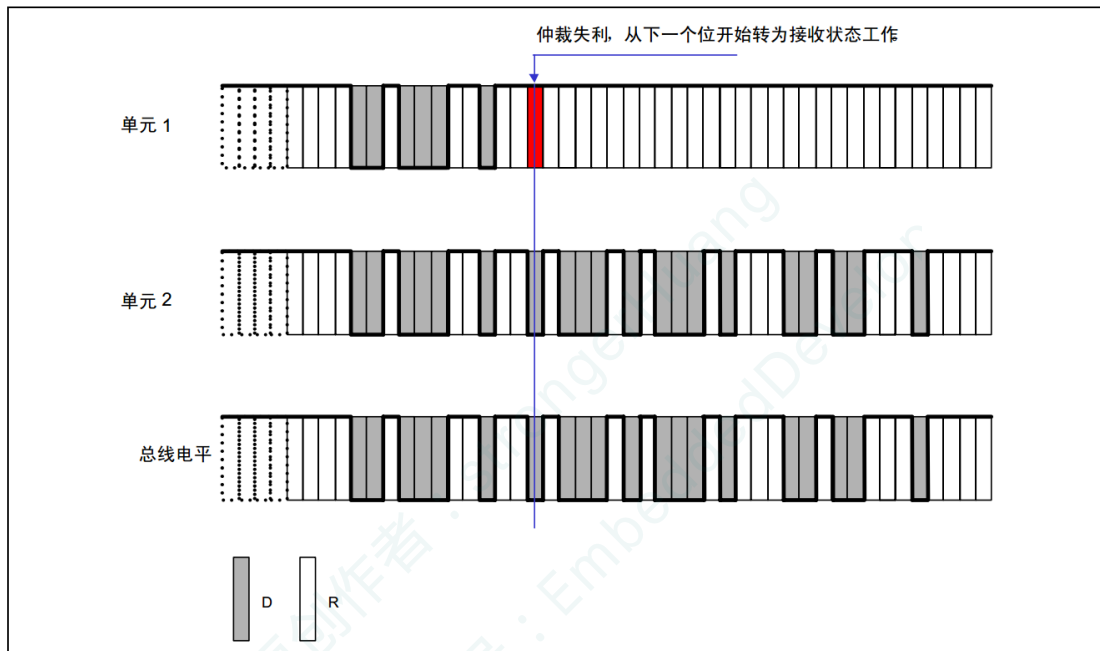
在接收数据帧和遥控帧时，SOF~CRC 段间的数据，相同电平如果持续 5 位，需要删除下一个位（第 6 个位）再接收。如果这个第 6 个位的电平与前 5 位相同，将被视为错误并发送错误帧。

## 三、优先级的决定

前面讲述“CAN 总线差分信号”时描述了显性和隐性，其实这是 CAN 总线的特点，也是决定 CAN 总线优先级比较重要的功能。

在 CAN 总线空闲态，最先开始发送消息的单元获得发送权。

多个单元同时开始发送时，各发送单元从仲裁段的第一位开始进行仲裁。连续输出显性电平最多的单元可继续发送。



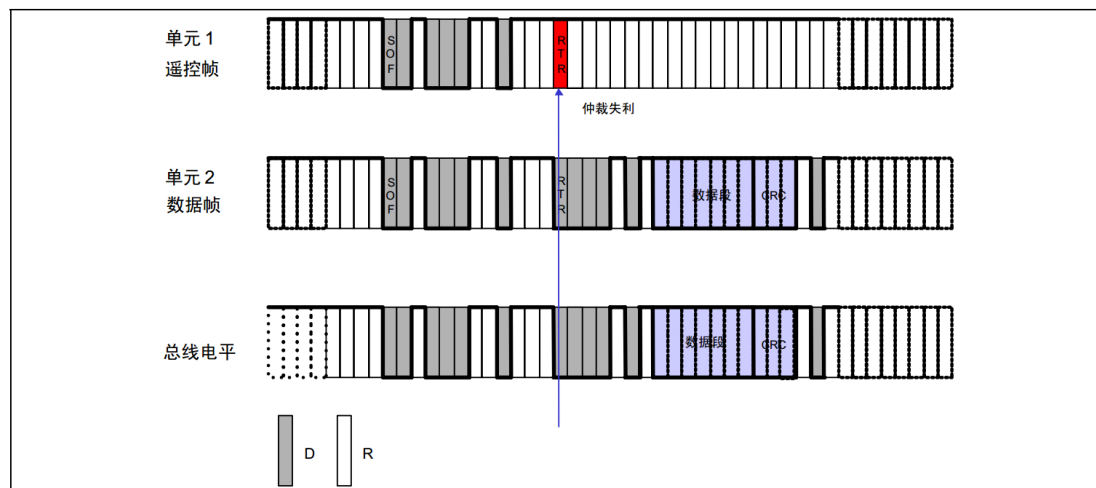
上图描述了 CAN 控制仲裁的过程，单元 2 最终获得了抢占 CAN 总线的权利，也就是说单元 2 具有更高的优先级。

当单元 1 仲裁失利时，自动转换为接收（本来该发送，优先级比别人低，就开始接收数据了。同时，控制器将会在下一轮继续发送这次未完成的数据）。

### 3.1 数据帧和遥控帧的优先级

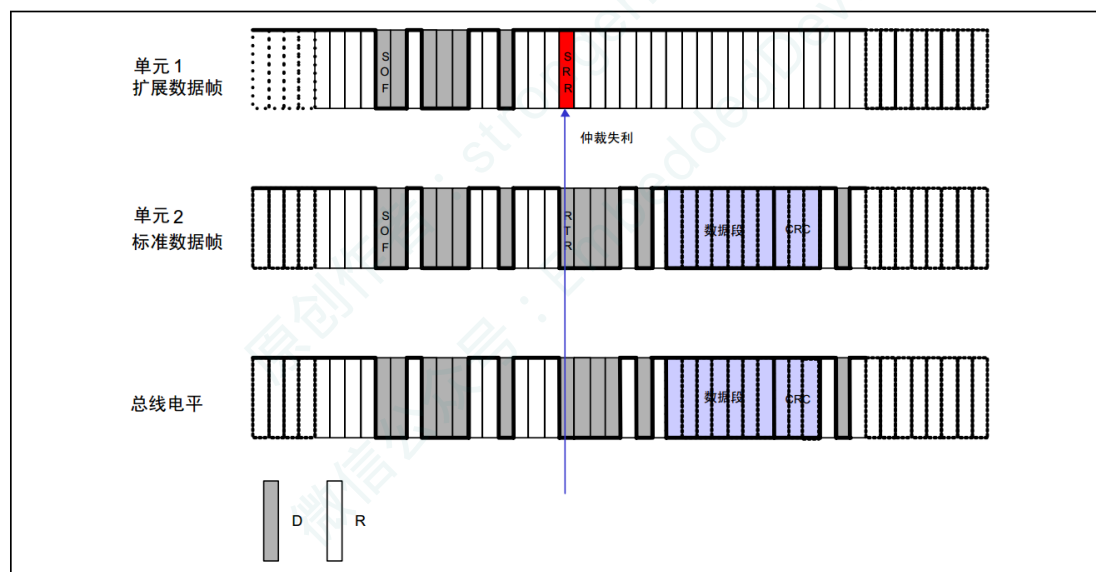
具有相同 ID 的数据帧和遥控帧在总线上竞争时，仲裁段的最后一位（RTR）为显性位的数据帧具有优先权，可继续发送。

数据帧和遥控帧的仲裁过程如下图：



### 3.2 标准格式和扩展格式的优先级

标准格式 ID 与具有相同 ID 的遥控帧或者扩展格式的数据帧在总线上竞争时，标准格式的 RTR 位为显性位的具有优先权，可继续发送。



#### 提示:

不明白数据帧和遥控帧，标准格式和扩展格式之间差异的朋友，建议参看我上一篇文章《[CANOpen 系列教程 04](#)》。

## 四、其他

CAN 总线协议的重要内容在这前面基本是讲述的差不多了，还有一些不是很重要的内容，比如硬件设计需要在 CAN\_H 和 CAN\_L 之间添加一个 120 欧电

阻，总线的传输距离等。

接下来的文章会结合 STM32 芯片集成 CAN 控制器、例程讲述如何编程控制 CAN 总线数据的收发等。

## 五、说明

- 1.该文档部分文字来自网络，仅供个人学习使用，版权所有，禁止商用。
2. 本文由我一个人编辑并整理，难免存在一些错误。
- 3.本教程收录于微信公众号「嵌入式专栏」，关注微信公众号回复【CANOpen 系列教程】即可查看全系列教程。

## 六、最后

我的博客: <http://www.strongerhuang.com>

我的 GitHub: <https://github.com/EmbeddedDevelop>

我的微信公众号 (ID: strongerHuang) 还在分享 STM8、STM32、Keil、IAR、FreeRTOS、UCOS、RT-Thread、CANOpen、Modbus...等更多精彩内容，如果想查看更多内容，可以关注我的微信公众号。

