

## CAN报文 Intel 格式与Motorola 格式的区别

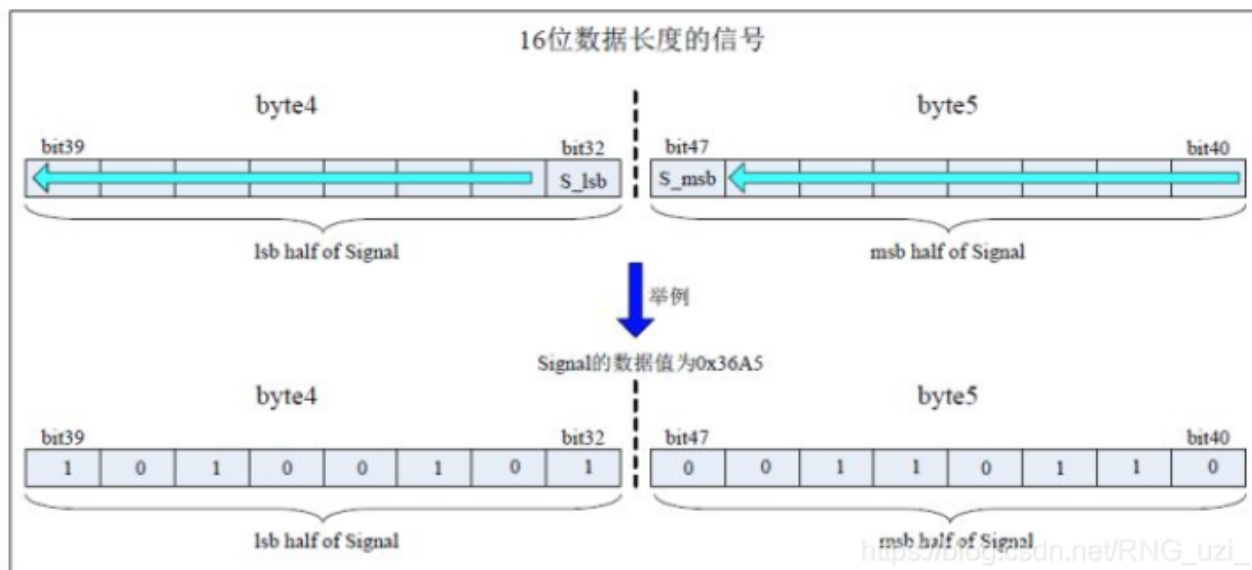
当一个信号的数据长度不超过 1 个字节（8 位）时，Intel 与 Motorola 两种格式的编码结果没有什么不同，完全一样。当信号的数据长度超过 1 个字节（8 位）时，两者的编码结果出现了明显的不同，信号的高位，即最能表达信号特性的因子，比如：车速信号 500km/h 按照给定的公式，转换成十六进制数为 0x6A5，因为 6 代表的数量级最大（162），那么其中 6 就是其信号的高位。信号的低位，即最不能表达信号特性的因子，比如：车速信号 500km/h 按照给定的公式，转换成十六进制数为 0x6A5，因为 5 代表的数量级最小（160），那么其中 5 就是其信号的低位。

信号的起始位，一般来讲，主机厂在定义整车 CAN 总线通信矩阵时，其每一个信号都从其最低位开始填写，这样也符合使用习惯。**所以信号的起始位就是信号的最低位**。这也与 CANoe 中 CANdb++的定义 Startbit 含义一致。

注：一般情况下，主机厂在定义CAN总线信号定义时，都会明确定义字节的发送顺序，即：以首先发送byte0（LSB），然后byte1，byte2，.....（MSB）的发送顺序

### Intel 格式

当一个信号的数据长度超过1 个字节（8 位）或者数据长度不超过一个字节但是采用跨字节方式实现时，该信号的高位（S\_msb）将被放在高字节（MSB）的高位，信号的低位（S\_lsb）将被放在低字节（LSB）的低位。这样，信号的起始位就是低字节的低位



如果已知是intel格式，知道起始位，以及信号长度就可以确定信号布局

如图中红线所指的，起始位是7，长度为15（），对与一个二进制15位的1010101010101010 1 分别放到下面的三行位置中。

×

信号布局:

|       | bit7 | bit6 | bit5 | bit4 | bit3 | bit2 | bit1 | bit0 |
|-------|------|------|------|------|------|------|------|------|
| BYTE0 | 7    | 6    | 5    | 4    | 3    | 2    | 1    | 0    |
| BYTE1 | 15   | 14   | 13   | 12   | 11   | 10   | 9    | 8    |
| BYTE2 | 23   | 22   | 21   | 20   | 19   | 18   | 17   | 16   |
| BYTE3 | 31   | 30   | 29   | 28   | 27   | 26   | 25   | 24   |
| BYTE4 | 39   | 38   | 37   | 36   | 35   | 34   | 33   | 32   |
| BYTE5 | 47   | 46   | 45   | 44   | 43   | 42   | 41   | 40   |
| BYTE6 | 55   | 54   | 53   | 52   | 51   | 50   | 49   | 48   |
| BYTE7 | 63   | 62   | 61   | 60   | 59   | 58   | 57   | 56   |

信号名称: COUNTER

起始位: 7

大小: 15

大小头: INTEL 符号: unsigned

值类型: int

应用

信号值编辑器:

物理值描述

逻辑值描述

比例因子: 1

偏移量: 0

最小值: 0

最大值: 255

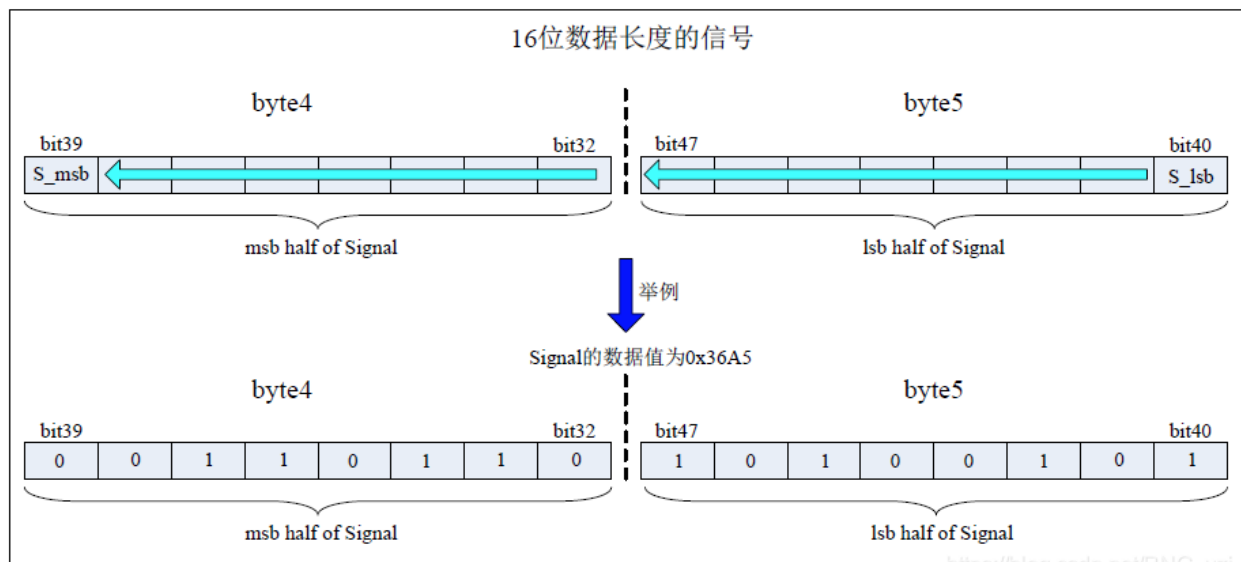
物理单位:

保存变更

接收节占.

## Motorola 格式

当一个信号的数据长度超过 1 个字节（8 位）或者数据长度不超过一个字节但是采用跨字节方式实现时，该信号的高位（S\_msb）将被放在低字节（MSB）的高位，信号的低位（S\_lsb）将被放在高字节的低位。这样，信号的起始位就是高字节的低位。



如果已知是motorola格式，知道起始位，以及信号长度就可以确定信号布局（这里之前还绕了许久）

如图中红线所指的，起始位是7，长度为15（），对与一个二进制15位的（0x5554）10101010 1010100 分别放到下面的二行位置中。

byte0 放的就是10101010 (0x55)

byte1 10101010 (0x54)

收到的数据为0x55 0x54 .....

解析数据的时候

0x5554十进制为21844 就是0x55 x256+0x54

注意：这里应该注意的已知是motorola格式，知道起始位，以及信号长度就可以确定信号布局，应该遵循，从起始位开始，从左至右，从上至下。

信号编辑器

信号列表:

TRACK\_A\_1

COUNTER

LAT\_DIST

LONG\_DIST

NEW\_TRACK

REL\_SPEED

VALID

CHECKSUM

信号布局:

|       | bit7 | bit6 | bit5 | bit4 | bit3 | bit2 | bit1 | bit0 |
|-------|------|------|------|------|------|------|------|------|
| BYTE0 | 7    | 6    | 5    | 4    | 3    | 2    | 1    | 0    |
| BYTE1 | 15   | 14   | 13   | 12   | 11   | 10   |      |      |
| BYTE2 | 23   | 22   | 21   | 20   | 19   | 18   | 17   | 16   |
| BYTE3 | 31   | 30   | 29   | 28   | 27   | 26   | 25   | 24   |
| BYTE4 | 39   | 38   | 37   | 36   | 35   | 34   | 33   | 32   |
| BYTE5 | 47   | 46   | 45   | 44   | 43   | 42   | 41   | 40   |
| BYTE6 | 55   | 54   | 53   | 52   | 51   | 50   | 49   | 48   |
| BYTE7 | 63   | 62   | 61   | 60   | 59   | 58   | 57   | 56   |

信号名称: COUNTER

起始位: 7

大小: 15

大小头: MOTO

符号: unsigned

值类型: int

应用

信号值编辑器:

物理值描述

逻辑值描述

比例因子: 1

偏移量: 0

最小值: 0

最大值: 255

物理单位:

保存变更

接收节点: