

# CAN-RS232 智能转换器 用户手册

## 1 产品介绍

### 1.1 概述

CAN-RS232智能协议转换器是用于CAN总线和RS-232总线之间数据交换的智能型协议转换器。

CAN-RS232智能协议转换器集成有1个RS-232通道、1个CAN通道，可以很方便地嵌入使用RS-232接口进行通讯的节点中，在不需改变原有硬件结构的前提下使设备获得CAN通讯接口，实现RS-232设备和CAN网络之间的连接、数据通讯。同时，CAN-RS232智能协议转换器也可作为配套模块直接嵌入用户的实际产品中。

转换器为用户的使用提供了足够的灵活性，用户可以根据实际需要设置RS-232通道和者CAN通道的通讯参数，能够满足用户在不同应用场合中对于RS-232数据与CAN数据之间数据转换的要求。CAN-RS232智能协议转换器的通讯参数由上位机软件配置，能使用户快速进入高效率的CAN通讯应用。

CAN-RS232的RS-232通道支持多种通讯波特率，CAN通道支持CiA推荐的10种标准通讯波特率，通讯速率范围为5Kbps~1000Kbps。RS-232通讯速率最高可达115200bps。

转换器采用表面安装工艺，板上自带光电隔离模块，完全电气隔离控制电路与CAN通讯电路，使CAN-RS232转换器具有很强的抗干扰能力，大大提高了系统在恶劣环境中使用的可靠性。

CAN-RS232智能协议转换器适合CAN-bus低速数据传输应用。CAN-RS232智能协议转换器具有体积小使用方便等特点，也是便携式系统用户的上佳选择。同样CAN-RS232智能协议转换器不仅适应基本CAN总线产品，也满足基于高层协议如DeviceNet、CANopen 等CAN总线产品的开发。

### 1.2 性能指标

- 支持CAN2.0A 和CAN2.0B 协议，符合ISO/DIS 11898 规范；
- 1 路 CAN 接口，波特率在5Kbps~1Mbps 之间可选；
- 1 路 RS-232 接口，波特率在 1200 bps~115200bps之间可选；
- 支持透明转换、透明带标识转换；
- 实现CAN和RS-232的双向转换；
- CAN接口采用光电隔离，1000 Vrms DC/DC 电源隔离；
- 工作电源：+9V~+24V DC；
- 工作温度：-40 °C~ +85°C；
- 安装方式：可选标准DIN导轨安装或简单固定方式；

### 1.3 典型应用

- 现有RS-232设备连接CAN-bus 网络
- 扩展标准RS-232 网络通讯长度
- PLC设备连接CAN-bus网络通讯
- CAN-bus与串行总线之间的网关网桥
- 工业现场网络数据监控
- CAN 教学应用远程通讯
- CAN 工业自动化控制系统
- 低速CAN 网络数据采集数据分析
- 智能楼宇控制数据广播系统等CAN-bus 应用系统

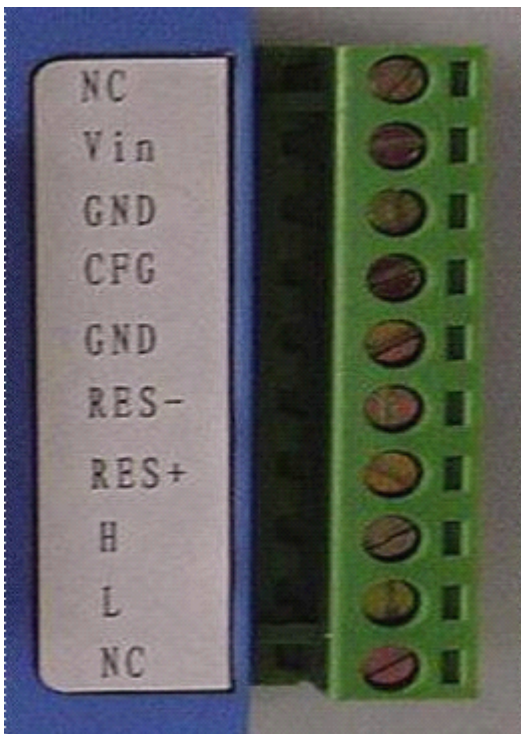
## 2 硬件介绍



### 接口描述

CAN-RS232 转换器具有两路用户接口。一路是CAN-bus接口，一路是RS-232接口。其接口引脚定义如下。

### CAN接口定义



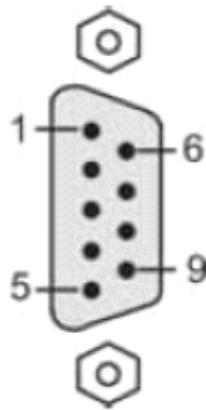
引脚含义（从上往下看） 1 NC-无连接 2 Vin 电源正 3 GND 电源地（0伏） 4 CFG 配置引脚 5 GND 电源地 6 Res- CAN网络匹配电阻端一 7 Res+ CAN网络匹配电阻端二 8 9 CANH CANL信号线连接端 10 NC-无连接

引脚2标示“Vin”接外部+9V~+24V直流电源，引脚3标示“GND”是接外部电源地。引脚4标示“CFG”是转换器的配置引脚。该脚悬空时上电后转换器进入正常转换模式；若该引脚和引脚5标示“GND”相连后，转换器上电即进入配置模式。

引脚6标示“Res-”和引脚7标示“Res+”接CAN网络的终端电阻。当CAN-RS232转换器作为CAN-bus网络终端时，两引脚间连接120欧姆的电阻；否则不用安装120欧姆的电阻。

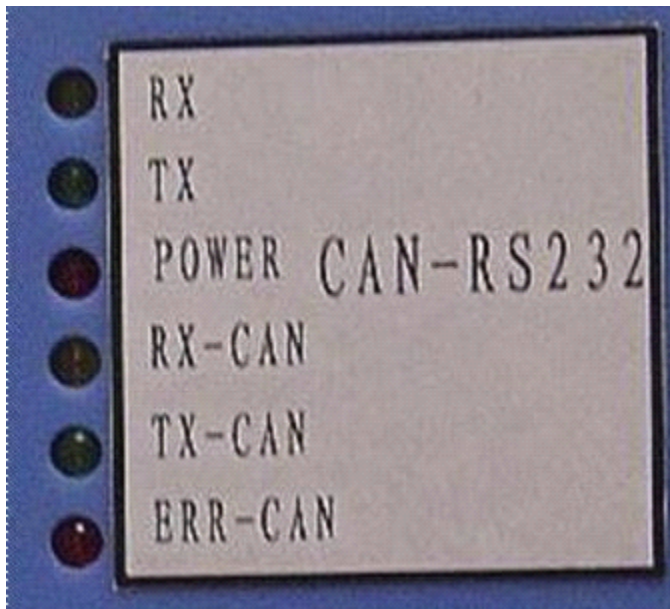
### RS-232接口引脚定义

RS-232 端口是标准的 DB9 孔座，引脚定义符合 RS-232 规范。这里采用的是三线连接，



引脚号	引脚名称	引脚含义
1		无连接
2	TXD	数据发送端
3	RXD	数据接收端
4		无连接
5	GND	地线
6		无连接
7		无连接
8		无连接
9		无连接

### 指示灯说明



RX代表串口的数据接收，平时为常亮，当串口有数据收到后闪烁。

TX代表串口的数据发送，平时为常亮，当串口有数据发送后闪烁。

POWER是电源指示灯，代表模块供电。

RX-CAN代表CAN的数据接收，平时为常灭，当模块收到CAN总线数据时闪烁。

TX-CAN代表CAN的数据发送，平时为常灭，当模块发送数据到CAN总线时闪烁。

ERR-CAN代表CAN总线错误指示灯，当CAN总线产生错误时灯亮，恢复正常后灯灭。

**【注】**：CAN 通讯线可以使用双绞线屏蔽双绞线若通讯距离超过1KM 应保证线的截面积大于 $1.0\text{mm}^2$  具体规格应根据距离而定常规是随距离的加长而适当加大。

### 3 配置说明

根据 CAN 和串口的通讯参数较多的特点，CAN-RS232 转换器也开放了大部分的参数让用户可以自行定义，以切合实际应用场合的需要。参数的配置是通过专门的配置软件完成，无需硬件跳线配置。

在正常使用之前，可以先配置好的转换器的转换参数，如果没有进行配置那么转换器执行的是上一次配置成功的参数（如果一次都没有配置，那么转换器执行默认的配置参数）。

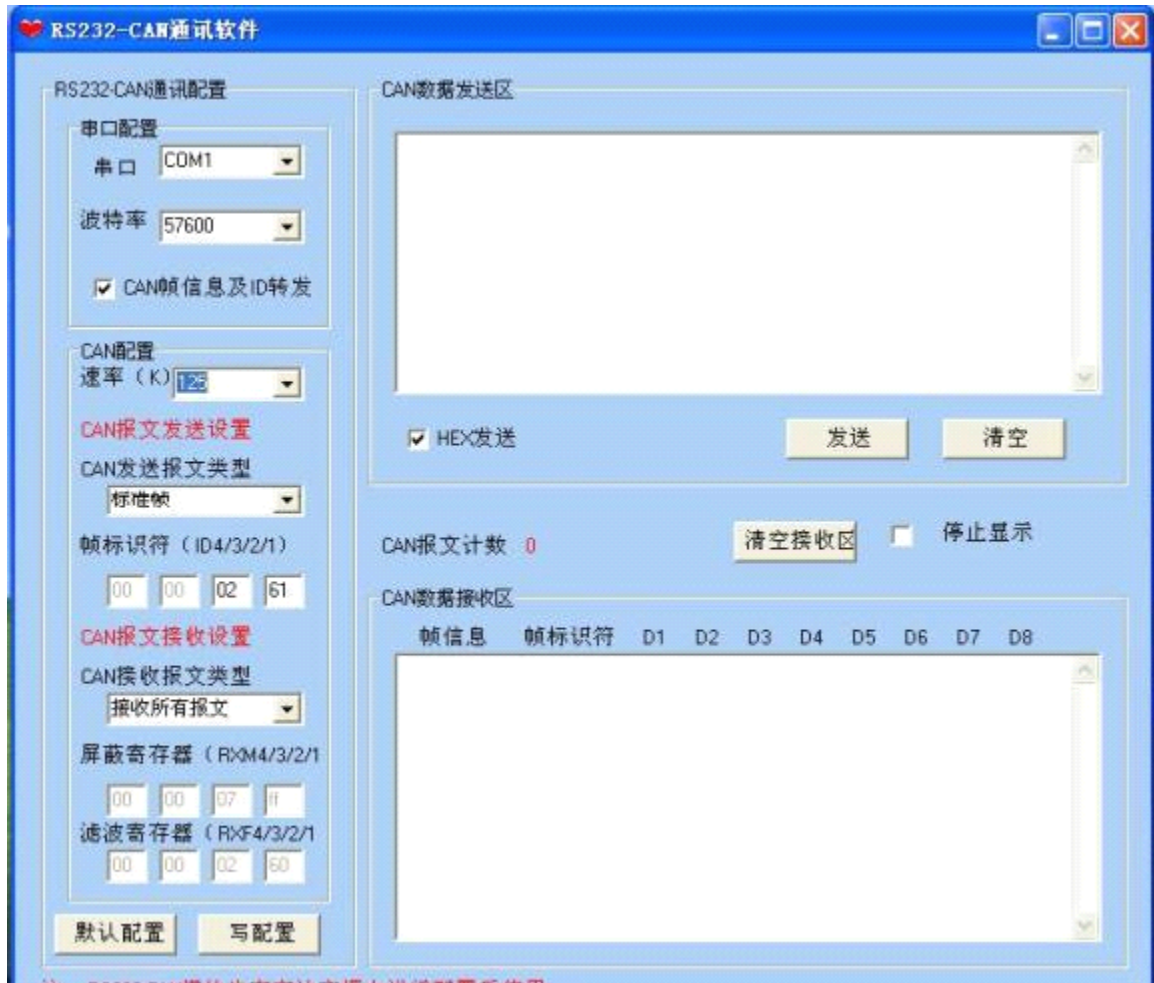
#### 3.1 配置方式

为了使转换器进入配置模式，设有一个专门的配置开关——CAN接口侧的引脚3标示“CFG”和引脚4标示“GND”。

“CFG”接地后，转换器上电进入“配置”模式；“CFG”脚悬空时，转换器上电进入“正常工作”模式。进入配置步骤如下：

1. 将转换器的CFG和GND用导线连通，然后上电。
2. 用串口线连接转换器和计算机。
3. 打开上位机配置软件，进行参数设定。

#### 3.2 软件说明

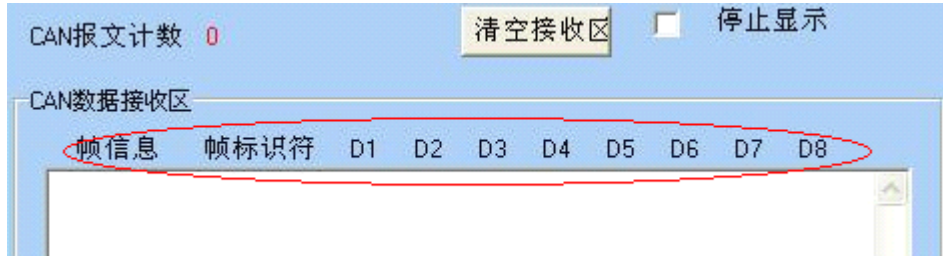


##### 3.2.1 串口参数设置

串口波特率：串口波特率在 1200bps~115200bps 间可选。

**CAN 帧信息及帧 ID 转发使能：**在 CAN 总线数据转换到串行总线的时候，该使能决定是否将 CAN 报文的帧信息及帧 ID 转换到串行数据中。

如果需要选择是(打钩√)，表示将 CAN 报文的帧信息及帧 ID 转换到串行数据中，此时



CAN接收区的 帧信息 帧标识符 D1-8会显示出来，接收框会相应的显示。

如果不需要，则不用选(去掉√)，表示CAN报文的帧信息及帧ID不会转换到串行数据中，此时不会显示 帧信息 帧标识符 D1-8，接收区和一般的串口接收一样，只接收数据字节。适合透明转换的应用，原来RS232的使用无须修改协议便可以直接使用。

### 3.2.2 CAN参数设置

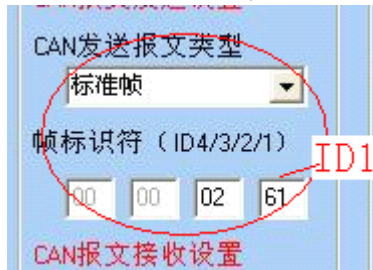
**1) CAN 波特率：**这里可以选择推荐的 10 种标准波特率，5K-1000K 可选。

**2) CAN 报文发送配置**包括发送报文类型和帧标识符配置。

发送报文类型分为标准帧发送和扩展帧发送。用户根据自己的需要选择要发送的报文类型即可。

选择标准帧，用户的数据将以标准帧格式发送，同时帧标识符是 11 位有效，所以 ID4, ID3 字节会变无效，只能配置 ID2, ID1；选择扩展帧，用户的数据则以扩展帧格式发送，同时帧标识符是 29 位有效，ID4/3/2/1 都要配置。

帧标识符即本模块发送到 CAN 总线上的 ID 号码，标准帧为 11 位 ID，扩展帧为 29 位 ID。



以上是标准帧的设置举例：报文类型—标准帧格式发送，节点 ID(ID2 为 XXXX X000, ID1 为 0000 0000，其中 X 是无效位，0 是用户需要设置的位)为 0x261。这里用户需要什么 ID，直接写入相应的框里，注意标准帧 11 位有效；扩展帧 29 位有效。

**3) CAN 报文接收配置**包括接收报文类型，屏蔽寄存器和滤波寄存器的设置。

接收报文类型分为接收所有报文，只接收标准帧和只接收扩展帧。用户根据自己需要选择接收报文。

选择接收所有报文，即关闭报文屏蔽滤波功能，此时，屏蔽寄存器和滤波寄存器无须设置(框变灰)，模块接收 CAN 总线上所有报文。

选择只接收标准帧，此时可以设置屏蔽寄存器和滤波寄存器来设置模块的屏蔽滤波功能。通过设置这两个寄存器，模块可以只接收符合条件的标准帧报文。



如上图,屏蔽寄存器是用来设置报文 ID 要校验的位(标准帧 11 位有效),1 表示该位需要校验,0 表示该位不需要校验.上面设置为 0x7ff,表示报文 ID 的 11 位都要校验.

滤波寄存器是配合屏蔽寄存器一起使用的,屏蔽寄存器需要校验的位,加上滤波寄存器在该位上设置的值.只有符合以下条件才会被接收.

屏蔽位 n	过滤位 n	报文标识符位 n001	接受或拒绝位 n
0	X	X	接受
1	0	0	接受
1	0	1	拒绝
1	1	0	拒绝
1	1	1	接受

注: X = 可为任意值

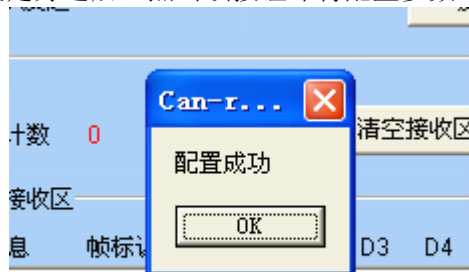
因此,上面滤波寄存器设置为 0x260 时,总线上只有报文 ID 为 0x260 的报文,本模块才会接收.此功能可以屏蔽一些报文的接收,提高总线的利用率.

选择只接收扩展帧和选择只接收标准帧情况相似,不作过多说明,就注意扩展帧是 29 位.

### 3.2.3 按键说明

默认配置: 可以将其参数恢复成出厂的默认值。

写配置: 在参数设定好之后, 点击该按钮即将配置参数写入 CAN-RS232 中。

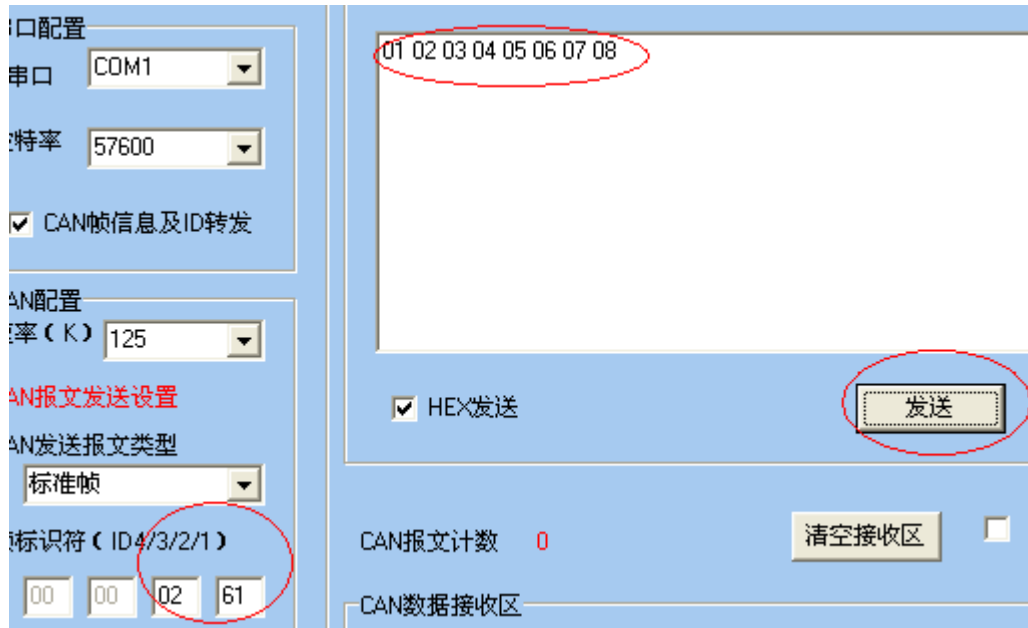


注意: 当点击默认配置和写配置按钮后出现如上提示, 表示配置成功, 若不出现, 请检查 CFG 跳线后再打开软件配置.

## 4 应用说明

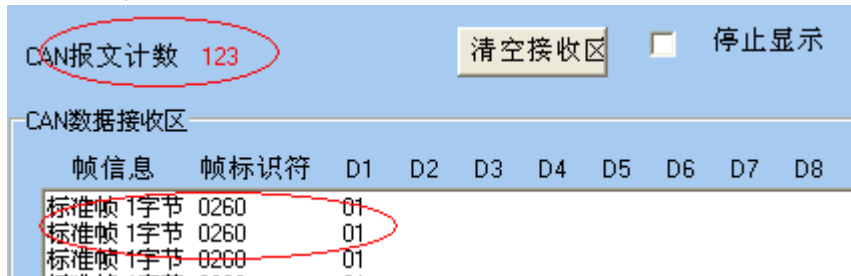
模块配置后, 改变 CFG 跳线, 重新上电后即进入正常工作模式. 这里把串口通讯功能分发送和接收两大块(用户也可以使用一般的串口调试工具). 若用此软件, CAN 发送区可以填写需要发送的数据字节, 点击发送按钮, 模块即把数据加上此前设置的帧类型和 ID 一起发送到 CAN 总线

上.



如上就是 标准帧信息字节+02+61+01+02+03+04+05+06+07+08 发到CAN总线上.

接收区设置了报文计数功能,即总线上收到多少帧报文的实时显示.还对报文进行了整理,按照CAN协议进行显示,直观反映CAN报文信息.



注意:这是使能了帧信息及ID转发的情况下,如果关闭此功能,收到的只是数据字节的内容,没有帧信息及ID,即实现真正的透明传输.

## 应用注意

- 建议在低速系统中使用,转换器不适用于高速数据传输。
- 在CAN232MB的使用或者测试过程中,CAN网络须加上一对终端电阻,否则可能致使通讯不成功。
- 在“配置模式”和“正常工作”模式切换之后,必须重新上电一次,否则仍然执行的是原来的工作模式,而不能成功的实现切换。上电断电过程不要太短暂。
- 由于CAN总线是半双工的,所以在数据转换过程中,必须保证两侧总线数据的有序性。如果两侧总线同时向转换器发送大量数据,将可能导致数据的转换不完全。
- 使用CAN232MB的时候,应该注意两侧总线的波特率和两侧总线发送数据的时间间隔的合理性,转换时应考虑波特率较低的总线的的数据承受能力。

比如在CAN总线数据转向串行总线的时候,CAN总线的速率能达到数千帧每秒,但是串行总线只能到数百帧每秒。所以当CAN总线的速率过快时会导致数据转换不完全。

一般情况下,CAN速度是串口速度的2-3倍,数据流透明转换时,串口发送方的速

度要≤串口接收方。

## 附录

### CAN2.0B 协议帧格式

#### CAN2.0B标准帧

CAN标准帧帧信息是11个字节，包括帧的信息和帧数据两部分。前3字节为帧的信息部分。格式如下表：

		7	6	5	4	3	2	1	0
字节 1	帧信息	FF	RTR	x	x	DLC (数据长度)			
字节 2	帧 ID1	(报文识别码) ID.10-ID.3							
字节 3	帧 ID2	ID.2-ID.0			x	x	x	x	x
字节 4	数据 1	数据							
字节 5	数据 2	数据							
字节 6	数据 3	数据							
字节 7	数据 4	数据							
字节 8	数据 5	数据							
字节 9	数据 6	数据							
字节 10	数据 7	数据							
字节 11	数据 8	数据							

- 字节1 为帧信息，第7 位 (FF) 表示帧格式，在标准帧中FF=0；第6 位 (RTR) 表示帧的类型，RTR=0表示为数据帧，RTR=1表示为远程帧。DLC表示在数据帧时实际的数据长度。
- 字节2~3 为报文识别码，其高11 位有效。
- 字节4~11 为数据帧的实际数据，远程帧时无效

#### CAN2.0B扩展帧

CAN标准帧帧信息是13个字节，包括帧的信息和帧数据两部分。前5字节为帧的信息部分。

		7	6	5	4	3	2	1	0
字节 1	帧信息	FF	RTR	x	x	DLC (数据长度)			
字节 2	帧 ID1	(报文识别码) ID.28-ID.21							
字节 3	帧 ID2	ID.20-ID.13							
字节 4	帧 ID3	ID.12-ID.5							
字节 5	帧 ID4	ID.4-ID.0				x	x	x	
字节 6	数据 1	数据							
字节 7	数据 2	数据							
字节 8	数据 3	数据							
字节 9	数据 4	数据							
字节 10	数据 5	数据							
字节 11	数据 6	数据							
字节 12	数据 7	数据							
字节 13	数据 8	数据							

- 字节1 为帧信息，第7 位（FF）表示帧格式，在扩展帧中FF= 0；第6 位（RTR）表示帧的类型，RTR=0表示为数据帧，RTR=1表示为远程帧。DLC表示在数据帧时实际的数据长度。
- 字节2~5 为报文识别码，其高13 位有效。
- 字节6~13 为数据帧的实际数据，远程帧时无效

**注意：**标准帧和扩展帧 I D 号和我软件中的 I D 并不一样，我已经由模块进行处理，因为这个并怎么好理解，而我的软件中的 I D 设置比较直观。