



CAN 模块使用说明

CAN 模块配合服务器主机使用，服务器主机使用请查阅相应的产品说明书。

1. 硬件使用说明：

CAN 模块插在串口服务器主板上即可直接使用，CAN 接口端子接线定义如下

序号	定义	说明
1	CAN_H	CAN 接口 H 引脚
2	CAN_L	CAN 接口 L 引脚
3	SET1	设置信号 1。 SET1 和 SET2 短接 模块进入设置模式， 指示灯快闪
4	SET2	设置信号 2

CAN 模块的串口参数为：115200 ,none,8,1。串口服务器对应的接口【串口参数】要设置为此参数。例如：CAN 模块插在串口服务器第一个插槽上，对应的接口为 U1。服务器的 web 设置应该为：



2. 软件使用方法：

CANSet.exe 软件为 CAN 模块配套设置软件。包含设置参数，固件升级，通信测试和滤波设置等功能。



软件和 CAN 模块通信先要建立 TCP 连接，连接到串口服务器对应接口。输入串口服务器 IP 和端口号 (U1 对应 1 0 0 0 1，U2 对应 1 0 0 0 2，可以在服务器 web 中查询端口号)。点击【连接】按钮就可以发起连接。

2.1 【参数设置】

要设置模块参数，先将模块接口的 3 线和 4 线短接才能进入设置模式。如下图：



2.1.1 CAN 参数设置

【CAN 参数】中“CAN 通信波特率”设置模块 CAN 接口速率，“CAN 帧 ID”设置 CAN 模块的 ID。“帧格式”设置 CAN 通信的数据格式。



The screenshot shows a dialog box titled "CAN参数" (CAN Parameters). It contains three main settings: "CAN通信波特率:" (CAN communication baud rate) set to "125K", "CAN帧ID:" (CAN frame ID) set to "01 02 03 04", and "帧格式:" (Frame format) set to "扩展帧" (Extended frame). At the bottom, there are two buttons: "读CAN参数" (Read CAN parameters) and "设置CAN参数" (Set CAN parameters).

2.1.2 转换参数设置

【转换参数】 设置通信数据帧格式转换。

The screenshot shows a dialog box titled "转换参数" (Conversion Parameters). It contains several settings: "转换模式:" (Conversion mode) set to "透明带标识转换" (Transparent with ID conversion), "转换方向:" (Conversion direction) set to "双向转换" (Bidirectional conversion), "串行帧间隔ms:" (Serial frame interval in ms) set to "10", "CAN帧转发间隔ms:" (CAN frame forwarding interval in ms) set to "10", "CAN帧ID转换设置:" (CAN frame ID conversion setting) set to "允许" (Allow), "CAN帧信息转换设置:" (CAN frame information conversion setting) set to "允许" (Allow), "标识符在串行帧中的起始偏移:" (Identifier start offset in serial frame) set to "0", and "标识符在串行帧中的长度:" (Identifier length in serial frame) set to "4". At the bottom, there are two buttons: "读转换参数" (Read conversion parameters) and "设置转换参数" (Set conversion parameters).

转换模式:

有 4 种模式。1 透明转换：串口数据透明传输到 CAN 口；2：透明带标识转换：串口数据中带有 CAN 标识符 (ID) 3：格式转换，按指定的格式进行数据转换；4：Modbus 模式。

转换方向:

(1).双向：转换器将串行总线的数据转换到 CAN 总线，也将 CAN 总线的数据转换到 串行总线 (2).仅串口转 CAN：只将串行总线的数据转换到 CAN 总线，而不将 CAN 总线的数据 转换到串行总线。(3).仅 CAN 转串口：只将 CAN 总线的数据转换到串行总线，而不将 串行总线的数据 转换到 CAN 总线



串行帧间隔：

该参数主要用于串行帧断帧，最短限制为 3ms，即断帧时间必须大于或等于 3ms

can 帧转发间隔：

该参数主要是用于串口转发 CAN 帧的间隔时间，例如：当 CAN 总线频繁有数据传输时，模块接收到 CAN 数据第一时间往串行总线转发，那么串行总线几乎是同时发出两帧 CAN 数据，不方便用户断帧区分，当 CAN 帧时间设置不为 0 时，当模块接收到 CAN 帧数据时，会暂存到消息队列里面，然后以用户设置的时间间隔从串行总线转发出去，方便用户断帧

.CAN 帧 ID 转换：

该参数仅在“透明转换”模式下使用，当选中该项后，转换器工作时会将 CAN 报文的帧 ID 添加在串行帧的帧数据之前，帧信息之后（如果允许帧信息转换）。未选中时不转换 CAN 的帧 ID。

CAN 帧信息转换设置：

该参数仅在“透明转换”模式下使用，当选中该项后，转换器工作时会将 CAN 报文的帧信息添加在串行帧的第一个字节。未选中时不转换 CAN 的帧信息。

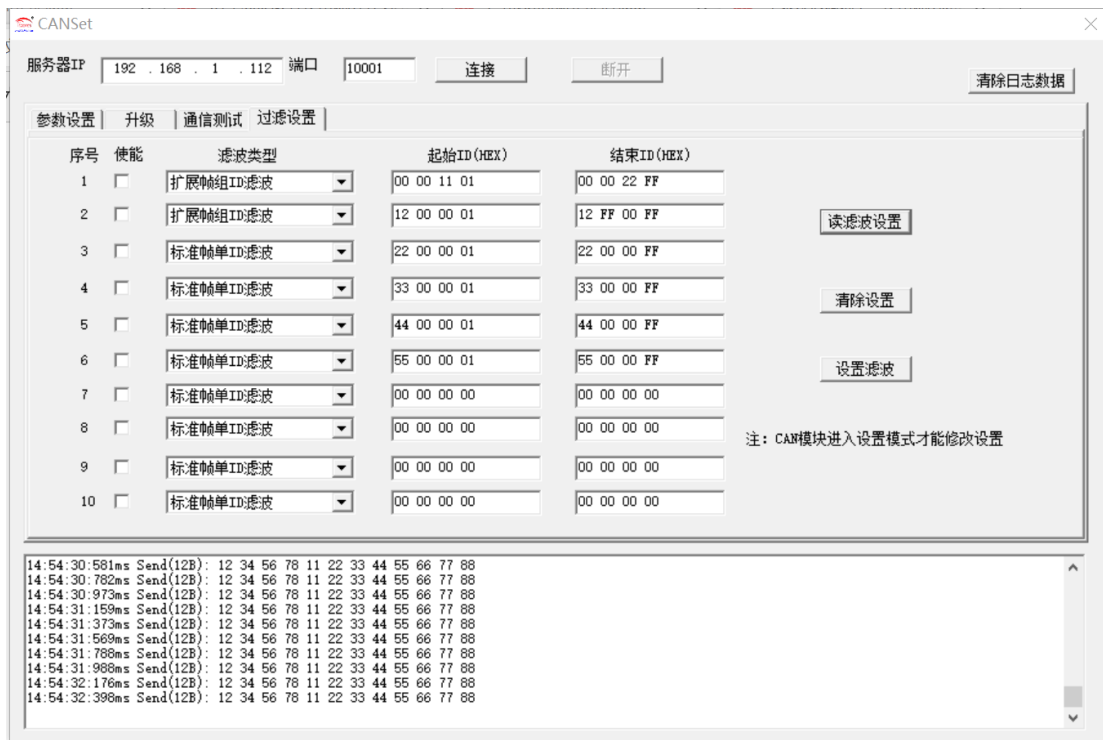
标识符在串行帧中的起始偏移：

参数仅在“透明带标识转换”模式下使用。在串口数据转换成 CAN 报文时，CAN 报文的帧 ID 的起始字节在串行帧中的偏移地址和帧 ID 的长度（参见 4.2 透明带标识转换）。

标识符在串行帧的长度 帧 ID 长度在标准帧的时候可填充 1 到 2 个字节，分别对应 CAN 报文的 ID1，ID2，在扩展帧的时候可以填充 1~4 个字节 ID1，ID2，ID3 和 ID4。

2.2 滤波设置

模块提供十组滤波，选择性接收，能够最大程度上减小自网络的网络负载。设置使能后，只有使能条件中的 ID 数据才转发。其它 ID 的数据丢弃。





2.3 升级

用于升级 CAN 模块固件程序，选择固件文件，点击【升级】按键即可完成模块程序更新。

2.4 通信测试

用于数据通信调试。勾选【帧 ID】时会在数据前加上 ID 标识符转发，本窗口暂不支持地址偏移测试。



3. 转换示例

3.1 透明转换：

4.1.透明转换：

4.1.1 串行帧转 CAN 报文

串行帧的全部数据依序填充到 CAN 报文帧的数据域里。转换器一检测到串行总线上有数据后就立即接收并转换。转换成的 CAN 报文帧信息（帧类型部分）和帧 ID 来自用户事先的配置，并且在转换过程中帧类型和帧 ID 一直保持不变。数据转换对应格式如图 4.1 所示。如果收到串的行帧长度小于等于 8 字节，依序将字符 1 到 n（n 为串行帧长度）填充到 CAN 报文的数据域的 1 到 n 个字节位置（如图 4.1 n 为 7）。如果串行帧的字节数大于 8，那么处理器从串行帧首个字符开始，第一次取 8 个字符依次填充到 CAN 报文的数据域。将数据发至 CAN 总线后，再转换余下的串行帧数据填充到 CAN 报文的数据域，直到其数据被转换完。

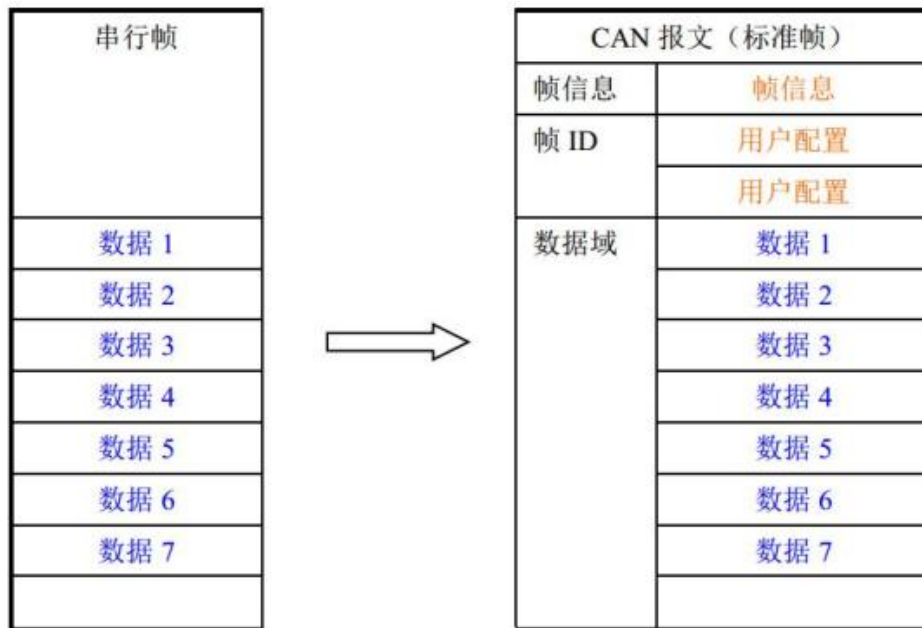


图 4.1 串行帧转换成 CAN 报文 (透明方式)

4.1.2. CAN 报文转串行帧

对于 CAN 总线的报文也是收到一帧就立即转发一帧。数据格式对应如图 4.2 所示转换时将 CAN 报文数据域中的数据依序全部转换到串行帧中。如果在配置的时候，“帧信息转换”项选择了“允许”，那么转换器会将 CAN 报文的“帧信息”字节直接填充至串行帧。如果“帧 ID 转换”项选择了“允许”，那么也将 CAN 报文的“帧 ID”字节全部填充至串行帧。



图 4.2 CAN 报文转换成串行帧 (透明方式)



4.2.透明带标识符传输

透明带标识转换是透明转换的特殊用法,有利于用户通过转换器更方便的组建自己的网络,使用自定的应用协议.该方式把串行帧中的地址信息自动转换成 CAN 总线的帧 ID.只要在配置中告诉转换器该地址在串行帧的起始位置和长度,转换器在转换时提取出这个帧 ID 填充在 CAN 报文的帧 ID 域里, 作为该串行帧的转发时的 CAN 报文的 ID。

在 CAN 报文转换成串行帧的时候也把 CAN 报文的 ID 转换在串行帧的相应位置。

4.2.1 串行帧转 CAN 报文串行帧中所带有的 CAN 的标识在串行帧中的起始地址和长度可由配置设定。起始地址的范围是 0~7, 长度范围分别是 1~2 (标准帧) 或 1~4 (扩展帧)。

转换时根据事先的配置将串行帧中的 CAN 帧 ID 对应全部转换到 CAN 报文的帧 ID 域中 (采用大端存储的方式, 如果所带帧 ID 个数少于 CAN 报文的帧 ID 个数, 那么在 CAN 报文中帧 ID 的低字节补 0), 其它的数据依序转换, 如图所示。如果一帧 CAN 报文未将串行帧数据转换完,则仍然用相同的 ID 作为 CAN 报文的帧 ID 继续转换直到将串行帧转换完成。

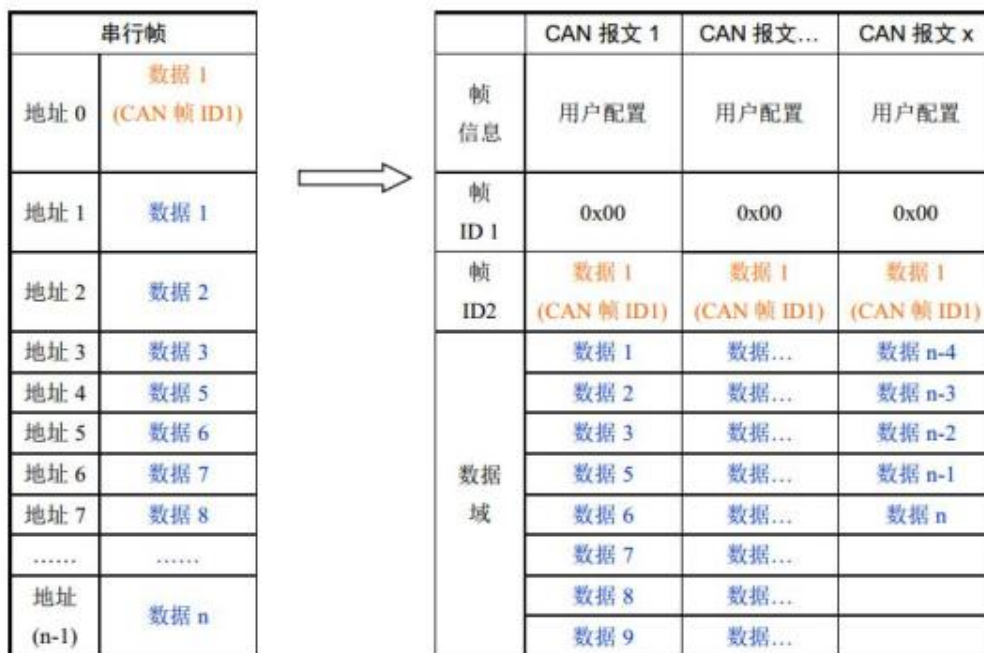


图 4.3 串行帧转 CAN 报文 (透明带标识)

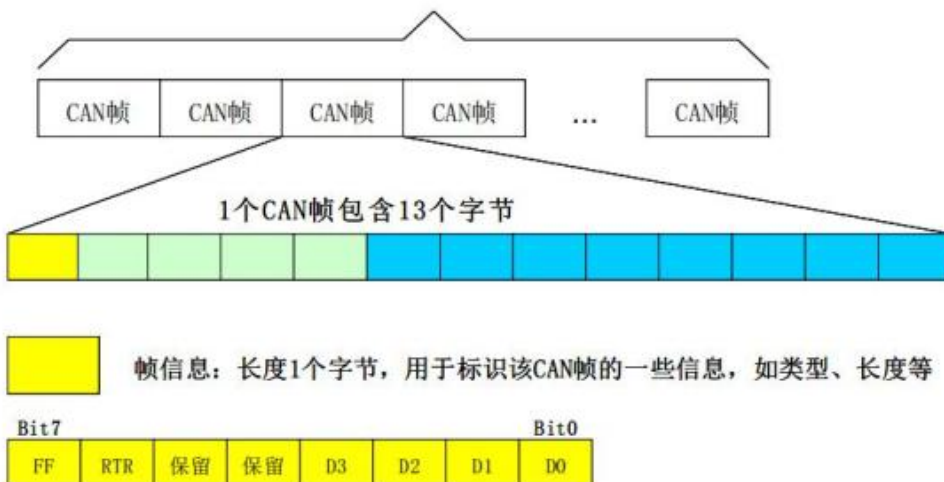
4.2.2 CAN 报文转串行帧对于 CAN 报文, 收到一帧就立即转发一帧,每次转发的时候也是根据事先配置的 CAN 帧 ID 在串行帧中的位置和长度把接收到的 CAN 报文中的 ID 作相应的转换。其它数据依序转发, 如图 4.4, 值得注意的是, 无论是串行帧还是 CAN 报文在应用的时候其帧格式 (标准帧还是扩展帧) 应该符合事先配置的帧格式要求, 否则可能致使通讯不成功。



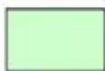
图 4.4 CAN 报文转串行帧 (透明带标识)

4.3 格式转换

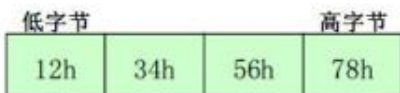
每一帧固定是 13 个字节，如果不足的必须补 0。同一串行数据帧中满足 13 个字节格式的串行数据对应一个 CAN 报文，不足 13 字节的串行数据帧不进行转换。所以要确保进行转换的串行数据帧以 13 字节对齐。在串行帧转 CAN 报文的过程中，如果以 13 字节对齐的串行数据帧中，某段 13 字节的数据格式不标准，将会不对这 13 字节进行转换，接着转换后面的数据。如果转换后发现少了某些 CAN 报文，请检查对应报文的 13 字节串行数据格式是否符合标准格式。



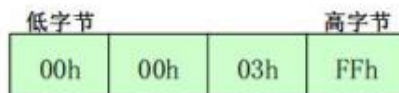
FF: 标准帧和扩展帧的标识, 1为扩展帧, 0为标准帧。
RTR: 远程帧和数据帧的标识, 1为远程帧, 0为数据帧。
保留值为0, 不可写入1。
D3~D0 : 标识该CAN帧的数据长度。



帧ID：长度4个字节，标准帧有效位是11位，扩展帧有效位是29位。



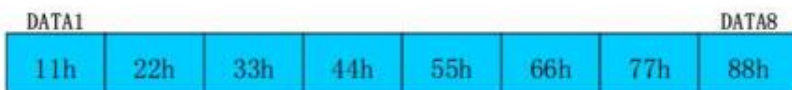
如上为扩展帧ID号
0x12345678的表示方式



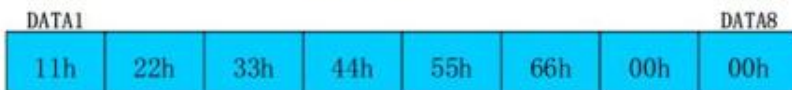
如上为标帧ID号
0x3FF的表示方式



帧数据：长度8个字节，有效长度由帧信息的D3~D0的值决定。



如上为8个字节有效数据
的表示方式



如上为6个字节有效数据
的表示方式

4.4 modbus 转换

Modbus 协议是一种标准的应用层协议，广泛应用于各种工控场合。该协议开放，实时性强，通讯验证机制好，非常适用于通信可靠性要求较高的场合。转换器在串口侧使用的是标准的 Modbus RTU 协议格式，所以转换器不仅支持用户使用 Modbus RTU 协议，转换器也可以直接和其它支持 Modbus RTU 协议的设备相接口。在 CAN 侧，制定了一个简单易用的分段通讯格式来实现 Modbus 的通讯。转换器在其中扮演的角色仍然是作协议验证和转发，支持 Modbus 协议的传输，而不是 Modbus 的主机或者从机，用户按照 Modbus 协议通讯即可。注意在该转换模式下，配置软件的“CAN 参数”项的“CAN ID”无效，因为此时发送的标识符（帧 ID）由 Modbus RTU 串行帧中的地址域填充。CAN 总线帧格式说明如下：



位编号	7	6	5	4	3	2	1	0
帧信息	FF	RTR	X	X	DLC (数据长度)			
帧 ID1	X	X	X	ID.28 - ID.24				
帧 ID2	ID.23 - ID.16							
帧 ID3	ID.15 - ID.8							
帧 ID4	ID.7 - ID.0							
数据 1	分段标记	分段类型		分段计数器				
数据 2	字符 1							
数据 3	字符 2							
数据 4	字符 3							
数据 5	字符 4							
数据 6	字符 5							
数据 7	字符 6							
数据 8	字符 7							

分段报文标记：表明该报文是否是分段报文。该位为 0 示单独报文，为 1 表示属于被分段报文中的一帧。注：当 CAN 报文为单帧的时候，分段标志位值为 0x00。

分段类型：表明是第一段、中间段的还是最后段。其值定义如下表格所示。

位值	含义	说明
0	第一个分段	如果分段计数器包含值 0，那么这是分段系列中的第一段。
1	中间分段	表明这是一个中间分段
2	最后分段	标志最后一个分段

转换说明：在串口侧向 CAN 侧转换的过程中，转换器只会在接收到一完整正确的 ModbusRTU 才会进行转换，否则无动作。如图 4.5，Modbus RTU 协议的地址域转换成 CAN 报文中帧 ID 的 ID4（扩展帧）ID2（标准帧），在转换该帧的过程中标识不变。

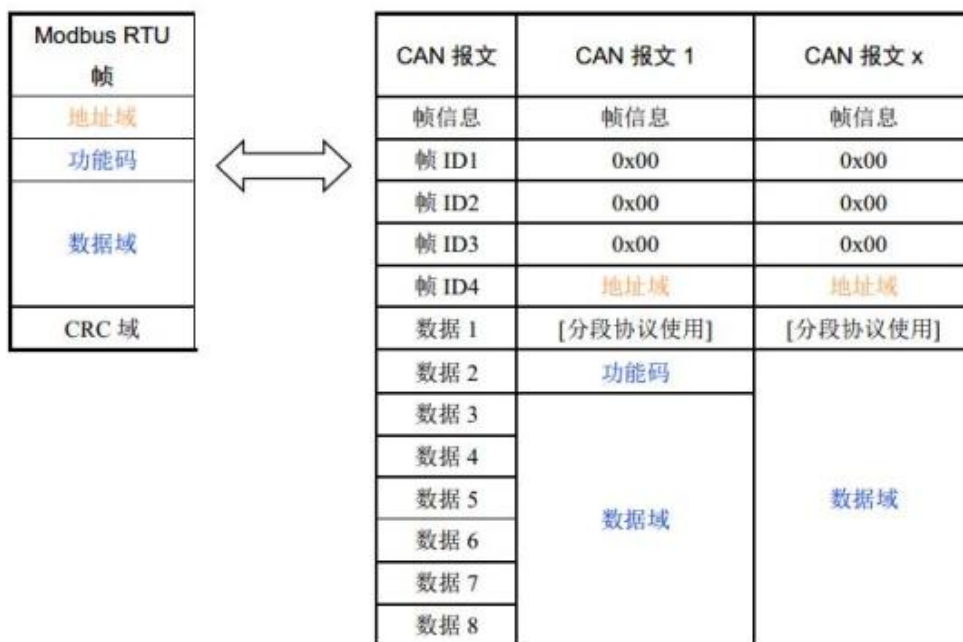


图 4.5 通信帧相互转换格式（Modbus 方式）

而 CRC 校验字节不转换到 CAN 报文中，CAN 的报文中也不必带有串行帧的校验字节，因为 CAN 总线本身就有较好的校验机制。转换的是 Modbus RTU 的协议内容——功能码和数据域，转换时将它们依次转换在 CAN 报文帧的数据域（从第二个数据字节开始，第一个数据字节为分段协议使用）里，由于 Modbus RTU 帧的长度根据功能码的不同而不同。而 CAN 报文一帧只能传送 7 个数据，所以转换器会将较长的 Modbus RTU 帧分段转换成 CAN 的报文后用上述的 CAN 分段协议发出。用户在 CAN 的节点上接收时取功能码和数据域处理即可。对于 CAN 总线的 Modbus 协议数据，无需做循环冗余校验（CRC16），转换器按照分段协议接收，接收完一帧解析后自动加上循环冗余校验（CRC16），转换成 Modbus RTU 帧发送至串行总线。如果接收到的数据不符合分段协议，则将该组数据丢弃不予转换。