

# CANFD-Serial 系列产品 转换模式说明

	内容
关键词	
摘要	

修订历史

版本	日期	原因
V1.0.0	2019/6/6	创建文档

## 目 录

1 四种转换模式说明 .....	4
1.1 透明转换 .....	5
1.1.1 帧格式 .....	5
1.1.2 转换方式 .....	5
1.1.3 转换示例 .....	8
1.2 透明带标识转换 .....	11
1.2.1 帧格式 .....	11
1.2.2 转换方式 .....	12
1.2.3 转换示例 .....	14
1.3 格式转换 .....	17
1.4 Modbus 转换 .....	19
1.4.1 帧格式 .....	19
1.4.2 转换方式 .....	21
1.4.3 转换示例 .....	23
2. 免责声明 .....	25

## 1 四种转换模式说明

CANFDRS-100IE 和 CANFDUM-100A 转换器是一款智能协议转换器，支持串口与 CAN(CANFD)之间转换。使用前需要先设置好要转换的 CAN 类型是普通 CAN 还是 CANFD。转换器给出了四种转换模式供选择，包括：透明转换、透明带标识转换、格式转换、Modbus 转换。在对转换器进行配置时可以进行参数的选择和设置。注意，相同转换模式下，CAN 类型不同，转换方式会有变化。比如，同是“透明转换”，普通 CAN 时，串口最多接收 8 字节就要转换成一个 CAN 报文。如果是 CANFD，则可以最多接收 64 字节数据转换成 CANFD 报文。

**透明转换：**转换器仅仅是将一种格式的总线数据原样转换成另一种总线的的数据格式，而不附加数据和对数据做修改。这样既实现了数据格式的交换又没有改变数据内容，对于两端的总线来说转换器如同透明的一样。这种方式下不会增加用户通讯负担，而能够实时的将数据原样转换，能承担较大流量的数据的传输。

**透明带标识转换：**透明转换的一种特殊的用法，也不附加协议。这种转换方式是根据通常的串行帧和 CAN(CANFD)报文的共有特性，使这两种不同的总线类型也能轻松的组建同一个通信网络。该方式能将串行帧中的“地址”转换到 CAN(CANFD)报文的标识域中，其中串行帧“地址”在串行帧中长度可配置，所以在这种方式下，转换器能最大限度地适应用户的自定义协议。

**格式转换：**是一种最简单的使用模式，数据格式约定为固定长度。CAN 时固定长度 13 个字节，CANFD 时固定长度为 69 个字节。即设置 CAN 类型为普通 CAN 时，固定 13 个字节的串行帧数据对应一个 CAN 报文。当为 CANFD 时，固定 69 个字节的串行帧数据对应一个 CANFD 报文。固定的串行帧内容包括帧信息（1 字节）+ID（4 字节）+数据（CAN 时 8 字节，CANFD 时 64 字节）。通过正确配置帧信息（第一个字节的数据），CAN 类型为普通 CAN 时，可以灵活地发出 CAN 的标准帧、扩展帧甚至远程帧。CAN 类型为 CANFD 时可以发出 CANFD 的标准帧、扩展帧。通过正确解析固定字节的串行帧可以得到标准帧、扩展帧甚至远程帧的细节。

**Modbus 转换：**将 Modbus 协议的 UART 数据和 CAN 数据之间进行转换。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 协议的设备接口。在 CAN(FD)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。

在“透明转换”和“格式转换”时，使用一个字节的帧信息来标识该 CAN 帧的一些信息，如类型、格式、长度等。帧信息格式如表 4.1 所示。

表 4.1 帧格式

FF	RTR	EDL	BRS(CANFD 有效)	L3	L2	L1	L0
0:标准帧 1:扩展帧	0:数据帧 1:远程帧	0:CAN 1:CANFD	0:禁能加速 1:使能加速	标识数据长度相当于 DLC			

## 1.1 透明转换

透明转换方式下，转换器接收到一侧总线的数据就立即转换发送至另一总线侧。这样以数据流的方式来处理，最大限度地提高了转换器的速度，也提高了缓冲区的利用率，因为在接收的同时转换器也在转换并发送，又空出了可以接收的缓冲区。

### 1.1.1 帧格式

#### 1. 串行总线帧

可以是数据流，也可以是带协议数据。根据用户定义 CAN 类型决定转发成 CAN 还是 CANFD 报文，转发到 CAN(CANFD)报文中的帧信息及帧 ID 由用户事先定义。

#### 2. CAN(CANFD)总线帧

CAN (CANFD) 报文帧的格式不变。其中的帧信息及帧 ID 需要通过配置是否使能，即是否将帧信息和帧 ID 转发到串行帧中。

### 1.1.2 转换方式

在透明转换模式，CAN 和 CANFD 报文的转换方式大致相同，区别主要在于一个 CANFD 报文可以转换成更多串行帧数据。

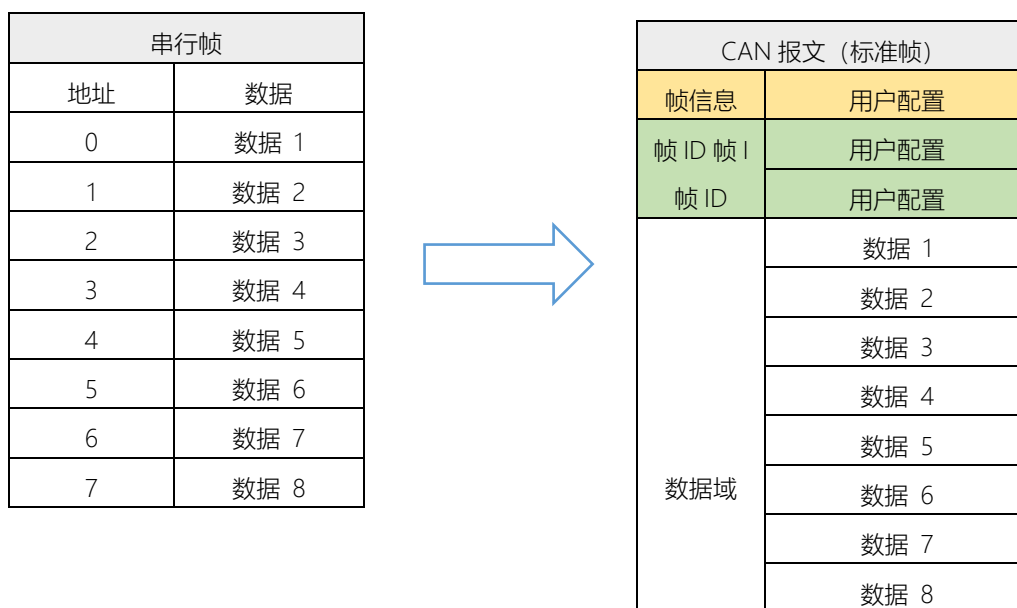
#### 1. 串行帧转 CAN (CANFD) 报文

串行帧的全部数据依序填充到 CAN (CANFD) 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 (CANFD) 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。

CAN 模式下数据转换对应格式如表 4.2 所示。如果收到串的行帧长度小于等于 8 字节，转换器会根据用户配置的串行帧之间的时间间隔进行超时，超时后还没接收到更多数据，那就把当前的数据依序将字符 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（如表 4.2 中 n 为 8）。

如果串行帧的字节数大于 8，那么转发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。

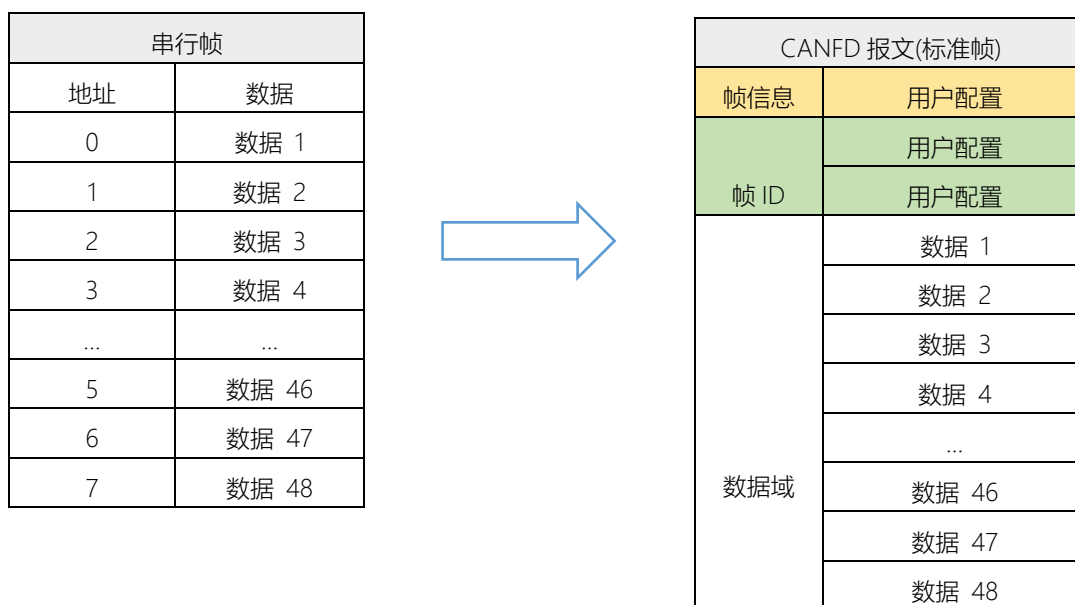
表 4.2 串行帧转换成 CAN 报文 (透明方式)



CANFD 模式下数据转换对应格式如表 4.3 所示。如果收到串的行帧长度小于等于 64 字节，转换器会根据用户配置的串行帧之间的时间间隔进行超时，超时后还没接收到更多数据，那就把当前的数据依序将字符 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 (如表 4.3 中 n 为 48)。注意，在 CANFD 模式时，如果串行帧的长度超过 8 字节，必须符合 CANFD 的 DLC 能编码的长度才能保证准确转换，即长度为 12、16、20、24、32、48、64，否则转换器会自动将拆分成合适长度的几个 CANFD 报文。比如，串行帧长度 62，CANFD 的 DLC 无法编码表示长度为 62，最接近只能是 48。所以转换出一个数据长度为 48 的 CANFD 报文。剩下数据长度为 14，CANFD 的 DLC 无法编码表示长度为 14，最接近只能是 12，所以再拆分出一个数据长度 12 的 CANFD 报文。最后剩下 2 字节数据转换成一个数据长度为 2 的 CANFD 报文。综上，62 字节串行帧会被拆分成 3 个 CANFD 报文。

如果串行帧的字节数大于 64，那么处理器从串行帧首个字符开始，第一次取 64 个字符依次填充到 CANFD 报文的数据域。将数据发至 CANFD 总线后，再转换余下的串行帧数据填充到 CANFD 报文的数据域，直到其数据被转换完。

表 4.3 串行帧转换成 CANFD 报文(透明转换)



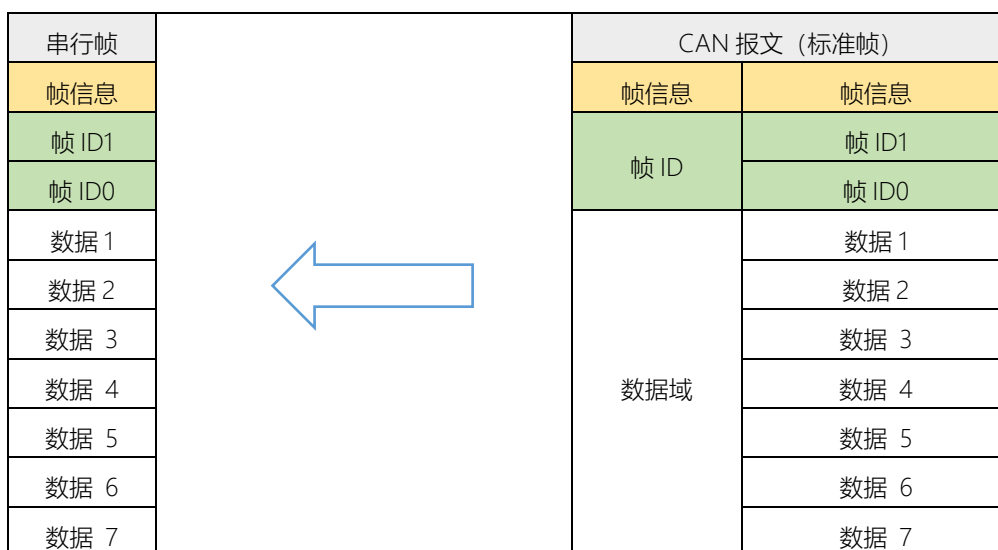
## 2. CAN (CANFD) 报文转串行帧

对于 CAN (CANFD) 总线的报文也是收到一帧就立即转发一帧。CAN 报文时，数据格式对应如表 4.4 所示。CANFD 报文时，数据格式对应如表 4.5 所示。

转换时将 CAN (CANFD) 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，“帧信息转换使能”项选择了“使能”，那么转换器会将 CAN (CANFD) 报文的“帧信息”字节直接填充至串行帧。

如果“帧 ID 转换使能”项选择了“使能”，那么也将 CAN (CANFD) 报文的“帧 ID”字节全部填充至串行帧。

表 4.4 CAN 报文转换成串行帧 (透明转换)



数据 8			数据 8
------	--	--	------

表 4.5 CANFD 报文转换成串行帧（透明转换）

串行帧		CANFD 报文（标准帧）	
		帧信息	帧信息
帧 ID1	←	帧 ID	帧 ID1
帧 ID0			帧 ID0
数据 1		数据域	数据 1
数据 2			数据 2
数据 3			数据 3
...			...
数据 61			数据 61
数据 62			数据 62
数据 63			数据 63
数据 64			数据 64

### 1.1.3 转换示例

#### 1. 串行帧转 CAN 报文

假设配置 CAN 类型为普通 CAN，帧信息为“标准帧”，帧 ID1, ID0 分别为“0x12, 0x34”，串行帧数据为 0x01 ~ 0x0C，那么转换格式如表 4.6 所示。CAN 报文的帧 ID 为 0x1234（用户配置），帧信息：标准帧（用户配置），串行帧中的数据部分将不作任何修改地转换到 CAN 报文中。

表 4.6 串行帧转 CAN 报文示例（透明方式）


串行帧			CAN 报文 1	CAN 报文 2
0x01		→	帧信息	0x08
0x02	帧 ID		0x12	0x12
0x03			0x34	0x34
0x04	数据域		0x01	0x09
0x05			0x02	0x0A
0x06			0x03	0x0B
0x07			0x04	0x0C
0x08			0x05	
0x09			0x06	
0x0A			0x07	
0x0B		0x08		
0x0C				



## 2. 串行帧转 CANFD 报文

假设配置 CAN 类型为 CANFD，帧信息为“标准帧”，使能波特率加速，帧 ID1，ID0 分别为“0x12，0x34”，串行帧数据为 0x01 ~ 0x46，共 70 个数据，那么转换格式如表 4.7 所示。CAN 报文的帧 ID 为 0x1234（用户配置），帧信息：标准帧（用户配置），CANFD 报文对应 EDL 位为 1，使能波特率加速对应 BRS 位为 1，串行帧中的数据部分将不作任何修改地转换到 CANFD 报文中，因为有 70 字节数据而一个 CANFD 报文最大数据量为 64 字节，所以分成两个 CANFD 报文。

表 4.7 串行帧转 CANFD 报文示例（透明方式）

串行帧			CANFD 报文 1	CANFD 报文 2	
0x01		帧信息	0x3F	0x36	
0x02		帧 ID	0x12	0x12	
0x03			0x34	0x34	
0x04		数据域		0x01	0x41
0x05				0x02	0x42
0x06				0x03	0x43
0x07				0x04	0x44
...				0x05	0x45
0x42				0x06	0x46
0x43				...	
0x44				0x3E	
0x45				0x3F	
0x46				0x40	

## 3. CAN 报文转串行帧

该示例配置的 CAN 报文“帧信息”转换，“帧 ID”转换。假设 CAN 报文中帧 ID1 为 0x12，帧 ID0 为 0x34，数据为 0x11，0x22，0x33，0x44，0x55，0x66，0x77，0x88，则 CAN 报文和转换后的串行帧表 4.8 所示。

1 表 4.8 CAN 报文转换成串行帧示例 (透明转换)

串行帧	←	CAN 报文 (标准帧)	
0x08		帧信息	0x08
0x12		帧 ID	0x12
0x34			0x34
0x11		数据域	0x11
0x22			0x22
0x33			0x33
0x44			0x44
0x55			0x55
0x66			0x66
0x77			0x77
0x88			0x88

如表 4.8 所示, CAN 报文的帧信息 (0x08) 转换到串行帧中的第 1 个字节 (0x08), CAN 报文中的 ID: 0x1234 依次转换到串行帧中的第 2 个字节 (0x12) 及第 3 个字节 (0x34)。CAN 报文的数据域将不作任何修改地转换到串行帧中的数据部分。

#### 4. CANFD 报文转串行帧

该示例配置的 CAN 类型为 CANFD, 使能报文“帧信息”转换, “帧 ID”转换。假设 CANFD 报文中帧 ID1 为 0x12, 帧 ID0 为 0x34, 数据为 0x01~0x40, 共 64 个数据。则 CANFD 报文和转换后的串行帧表 4.9 所示。

表 4.9 CANFD 报文转换成串行帧示例 (透明转换)

串行帧	←	CANFD 报文 (标准帧)	
0x3F		帧信息	0x3F
0x12		帧 ID	0x12
0x34			0x34
0x01		数据域	0x01
0x02			0x02
0x03			0x03
0x04			0x04
...			...
0x3E			0x3E
0x3F			0x3F
0x40			0x40

如表 4.9 所示，CANFD 报文的帧信息 (0x3F) 转换到串行帧中的第 1 个字节 (0x3F)，CANFD 报文中的 ID: 0x1234 依次转换到串行帧中的第 2 个字节 (0x12) 及第 3 个字节 (0x34)。CANFD 报文的数据域将不作任何修改地转换到串行帧中的数据部分。

## 1.2 透明带标识转换

透明带标识转换是透明转换的特殊用法，在串行帧中带 CAN (CANFD) 报文的 ID 信息，可以根据需要发送不同 ID 的 CAN (CANFD) 报文。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 ID 信息自动转换成 CAN (CANFD) 总线的帧 ID。只要在配置中告诉转换器该 ID 信息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 (CANFD) 报文的帧 ID 域里，作为该串行帧的转发时的 CAN (CANFD) 报文的 ID。在 CAN (CANFD) 报文转换成串行帧的时候也把 CAN (CANFD) 报文的 ID 转换在串行帧的相应位置。

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) 由上述的串行帧中的 ID 信息数据填充。

### 1.2.1 帧格式

#### 1. 串行总线帧

带标识转换时，必须取得完整的串行数据帧，转换器以两帧间的时间间隔作为帧的划分。并且该间隔可由用户设定，帧格式如图 4.1 所示。串行帧最大长度为缓冲区的长度：5000 字节。

// 无图

图 4.1 串行帧时间格式 (透明带标识转换)

转换器在串行总线空闲状态下检测到的首个数据作为接收帧的首个字符。传输过程中，同一帧的字符间的时间间隔必须小于用户设置的串行帧之间的时间间隔。

转换器在接收到一个字符后，如果小于用户设置的串行帧之间的时间间隔内没有字符再被接收到，转换器就认为此帧传输结束，将该字符作为此帧的最后一个字符。超时之后的字符不属于该帧，而是下一帧的内容。

#### 2. CAN(CANFD)总线帧

CAN(CANFD)报文的格式不变，CAN(CANFD)报文中相应的帧 ID 也会被转换到串行帧中，但帧信息不被转换。

## 1.2.2 转换方式

### 1. 串行帧转 CAN (CANFD) 报文

串行帧中所带有的 CAN(CANFD)报文的“帧 ID”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(CANFD)报文“帧 ID”对应全部转换到 CAN(CANFD)报文的帧 ID 域中 (如果所带帧 ID 个数少于 CAN(CANFD)报文的帧 ID 个数，那么在 CAN(CANFD)报文中帧 ID 的高字节补 0)，其它的数据依序转换，如果一帧 CAN(CANFD)报文未将串行帧数据转换完，则仍然用相同的 ID 作为 CAN(CANFD)报文的帧 ID 继续转换直到将串行帧转换完成。

串行帧 CAN 如表 4.10 所示，串行帧转 CANFD 如表 4.11 所示。注：标准帧的帧 ID 范围为：0x000-0x7fff，分别表示为帧 ID1、帧 ID0，其中帧 ID1 为高字节，扩展帧的帧 ID 范围为：0x00000000-0xffffffff，分别表示为帧 ID3、帧 ID2、帧 ID1、帧 ID0，其中帧 ID3 为高字节。

表 4.10 串行帧转 CAN 标准帧报文 (透明带标识)

串行帧			CAN 报文 1	CAN 报文 2	CAN 报文 3
地址 0	数据 1 (CAN 帧 ID1)	帧信息	用户配置	用户配置	用户配置
		帧 ID 1	数据 1 (CAN 帧 ID1)	数据 1 (CAN 帧 ID1)	数据 1 (CAN 帧 ID1)
地址 1	数据 2 (CAN 帧 ID0)	帧 ID 0	数据 2 (CAN 帧 ID0)	数据 2 (CAN 帧 ID0)	数据 2 (CAN 帧 ID0)
地址 2	数据 3	数据域	数据 3	数据...	数据 n-4
地址 3	数据 4		数据 4	数据...	数据 n-3
地址 4	数据 5		数据 5	数据...	数据 n-2
地址 5	数据 6		数据 6	数据...	数据 n-1
地址 6	数据 7		数据 7	数据...	数据 n
...	...		数据 8	数据...	
地址 (n-1)	数据 n		数据 9	数据...	
			数据 10	数据...	

表 4.11 串行帧转 CANFD 标准帧报文 (透明带标识)

串行帧			CAN 报文 1	CAN 报文 2	CAN 报文 3
地址 0	数据 1 (CANFD 帧 ID1)	帧信息	用户配置	用户配置	用户配置
		帧 ID 1	数据 1 (CANFD 帧 ID1)	数据 1 (CANFD 帧 ID1)	数据 1 (CANFD 帧 ID1)
地址 1	数据 2 (CANFD 帧 ID0)	帧 ID0	数据 2 (CANFD 帧 ID0)	数据 2 (CANFD 帧 ID0)	数据 2 (CANFD 帧 ID0)
地址 2	数据 3	数据域	数据 3	数据...	数据 n-4
地址 3	数据 4		数据 4	数据...	数据 n-3
地址 4	数据 5		数据 5	数据...	数据 n-2
地址 5	数据 6		数据 6	数据...	数据 n-1
地址 6	数据 7		数据 7	数据...	数据 n
...	...		数据 8	数据...	
地址 (n-1)	数据 n		数据 9	数据...	
			数据 10	数据...	

## 2. CAN(CANFD)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, 假设设置的 CAN 类型为普通 CAN, 报文为标准帧, “帧 ID”起始地址为 0, 长度为 2, 则转换方式如表 4.12 所示。假设设置的 CAN 类型为 CANFD, 报文为标准帧, “帧 ID” 起始地址为 0, 长度为 2, 则转换方式如表 4.13 所示。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。

表 4.12 CAN 标准帧报文转换成串行帧 (透明带标识)

串行帧		CAN 报文 (标准帧)		
帧 ID1	←	帧信息	帧信息	
帧 ID0		帧 ID	帧 ID1	
数据 1			帧 ID0	
数据 2		数据域		数据 1
数据 3				数据 2
数据 4				数据 3
数据 5				数据 4
数据 6				数据 5
数据 7			数据 6	

数据 8			数据 7
			数据 8

表 4.13 CANFD 标准帧报文转换成串行帧（透明带标识）

串行帧			CANFD 报文（标准帧）	
帧 ID1			帧信息	帧信息
帧 ID0		帧 ID	帧 ID1	
数据 1			帧 ID0	
数据 2		数据域	数据 1	
数据 3			数据 2	
...			数据 3	
数据 61			...	
数据 62			数据 61	
数据 63			数据 62	
数据 64			数据 63	
			数据 64	

### 1.2.3 转换示例

#### 1. 串行帧转 CAN 报文

假定 CAN 标识在串行帧中的起始地址是 0，长度是 3（扩展帧情况下），串行帧发送的数据分别为 0x00~0x0E，串行帧和转换成 CAN 报文结果如表 4.14 所示。其中，两帧 CAN 报文用相同的 ID 进行转换。

表 4.14 串行帧转 CAN 报文示例（透明带标识方式）

串行帧			CAN 报文 1	CAN 报文 2
地址 0	0x00		帧信息	0x88
地址 1	0x01	帧 ID 3	0x00	0x00
地址 2	0x02	帧 ID 2	0x00	0x00
地址 3	0x03	帧 ID 1	0x01	0x01
地址 4	0x04	帧 ID 0	0x02	0x02
地址 5	0x05	数据域	0x03	0x0B
地址 6	0x06		0x04	0x0C
地址 7	0x07		0x05	0x0D
地址 8	0x08		0x06	0x0E
地址 9	0x09		0x07	

地址 10	0x0A			0x08	
地址 11	0x0B			0x09	
地址 12	0x0C			0x0A	
地址 13	0x0D				
地址 14	0x0E				

如表 4.14 所示，因为事先配置的 CAN 报文为扩展帧，所以在串行帧转 CAN 报文中的帧 ID 为 4 个字节。由于配置串行帧中的起始地址是 0，长度是 3，因此从串行帧中地址为 0 连续取 3 个字节，在本例中分别为 0x00，0x01，0x02，在转换时依次将这三个字节填充到 CAN 报文的帧 ID2、帧 ID1、帧 ID0，但帧 ID3 被填充为 0x00。串行帧中其余的数据将不作任何修改地转换到 CAN 报文的数据域。

## 2. 串行帧转 CANFD 报文

假定 CANFD 标识在串行帧中的起始地址是 0，长度是 3（扩展帧情况下），串行帧发送的数据分别为 0x00~0x45，共 70 个数据。串行帧和转换成 CAN 报文结果如表 4.15 所示。其中，两帧 CAN 报文用相同的 ID 进行转换。

表 4.15 串行帧转 CANFD 报文示例（透明带标识方式）

串行帧			CANFD 报文 1	CANFD 报文 2
地址 0	0x00	帧信息	0xBF	0xB3
地址 1	0x01	帧 ID 3	0x00	0x00
地址 2	0x02	帧 ID 2	0x00	0x00
地址 3	0x03	帧 ID 1	0x01	0x01
地址 4	0x04	帧 ID 0	0x02	0x02
地址 5	0x05	数据域	0x03	0x43
地址 6	0x06		0x04	0x44
地址 7	0x07		0x05	0x45
...	...		0x06	
地址 64	0x40		...	
地址 65	0x41		0x40	
地址 66	0x42		0x41	
地址 67	0x43		0x42	
地址 68	0x44			
地址 69	0x45			

如表 4.15 所示，因为事先配置 CAN 类型为 CANFD，帧格式为扩展帧，所以在串行帧转 CANFD 报文中的帧 ID 为 4 个字节。由于配置串行帧中的起始地址是 0，长度是 3，因此

从串行帧中地址为 0 连续取 3 个字节，在本例中分别为 0x00, 0x01, 0x02，在转换时依次将这三个字节填充到 CAN 报文的帧 ID2、帧 ID1、帧 ID0，但帧 ID3 被填充为 0x00。串行帧中其余的数据将不作任何修改地转换到 CANFD 报文的数据域。

### 3. CAN 报文转串行帧

假定配置的 CAN 标识在串行帧中的起始地址是 0，长度是 3（扩展帧情况下），CAN 报文的帧 ID 为 0x00123456，数据为 0x00 ~ 0x07。CAN 报文和转换成串行帧的结果如表 4.16 所示。

表 4.16 CAN 扩展帧报文转换成串行帧（透明带标识）

串行帧			CAN 报文 1	
地址 0	0x12	←	帧信息	
地址 1	0x34		帧 ID	
地址 2	0x56			0x00
地址 3	0x00			0x12
地址 4	0x01			0x34
地址 5	0x02		数据域	0x56
地址 6	0x03			0x00
地址 7	0x04			0x01
地址 8	0x05			0x02
地址 9	0x06			0x03
地址 10	0x07	0x04		
		0x05		
		0x06		
		0x07		

### 4. CANFD 报文转串行帧

假定配置的 CANFD 标识在串行帧中的起始地址是 0，长度是 3（扩展帧情况下），CANFD 报文的帧 ID 为 0x00123456，数据为 0x00 ~ 0x3F。CANFD 报文和转换成串行帧的结果如表 4.17 所示。

表 4.17 CANFD 扩展帧报文转换成串行帧（透明带标识）

串行帧			CANFD 报文 1	
地址 0	0x12	←	帧信息	
地址 1	0x34		帧 ID	
地址 2	0x56			0x00
地址 3	0x00			0x12
地址 4	0x01			0x34
			0x56	



地址 5	0x02		数据域	0x00
地址 6	0x03			0x01
...	...			0x02
地址 64	0x3D			0x03
地址 65	0x3E			...
地址 66	0x3F			0x3D
				0x3E
				0x3F

如表 4.17 所示，在这种模式下，明显可以看到 CANFD 报文的帧信息将不转换到串行帧中。由于配置的串行帧中的起始地址是 2，长度是 3（扩展帧情况下）。因此在转换时，将 CANFD 报文中的帧 ID2、帧 ID1、帧 ID0 依次转换到串行帧为地址 0、地址 1、地址 2 中的内容。CAN 报文的数据域将不作任何修改地依次转换到串行帧中未被填充部分。

### 1.3 格式转换

格式转换时根据设置的 CAN 类型，分 CAN 格式转换和 CANFD 格式转换两种情况，区别是对应的格式固定的字节数不同。CAN 格式转换固定 13 字节表示一个 CAN 帧，CANFD 格式转换固定 69 字节表示一个 CANFD 帧，如表 4.18 所示每一个 CAN 帧包含 13 个字节，13 个字节内容包括 CAN 帧信息 + 帧 ID + 帧数据。如表 4.19 所示每一个 CANFD 帧包含 69 个字节，69 个字节内容包括 CANFD 帧信息 + 帧 ID + 帧数据。注意在该转换模式下，配置的 CAN ID 无效，因为此时发送的标识符（帧 ID）由上述的格式串行帧中的帧 ID 数据填充。配置的帧类型也无效，由格式串行帧中的帧信息来决定帧类型。

表 4.18 CAN 固定格式串行帧

CAN 固定格式串行帧 (13 字节)		
帧信息	帧 ID	帧数据
1Byte	4Byte	8Byte

表 4.19 CANFD 固定格式串行帧

CANFD 固定格式串行帧 (69 字节)		
帧信息	帧 ID	帧数据
1Byte	4Byte	64Byte

帧信息: 如表 4.20 所示，长度为 1 字节，用于标识该 CAN(CANFD)报文的一些信息，如类型、长度等。通过正确配置帧信息（第一个字节的数据），可以灵活地发出标准帧、扩展帧甚至 CAN 远程帧。通过正确解析固定字节的格式串行帧可以得到标准帧、扩展帧甚至远程帧的细节。

表 4.20 帧信息

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
FF	RTR	EDL	BRS	DLC3	DLC2	DLC1	DLC0

- ◆ FF: 标准帧和扩展帧的标识, 0 为标准帧, 1 为扩展帧;
- ◆ RTR: 远程帧和数据帧的标识, 0 为数据帧, 1 为远程帧, CANFD 时只能为 0;
- ◆ EDL: CAN 和 CANFD 标识, 0 为 CAN, 1 为 CANFD;
- ◆ BRS: 波特率切换使能标识, 0 为不转换速率, 1 为转换可变速率, 仅 CANFD 时有效;  
CAN 时此位应该为 0;
- ◆ DLC3~DLC0: 标识该 CAN(CANFD)报文数据长度。

帧 ID: 长度为 4 字节, 标准帧有效位是 11 位, 扩展帧有效位是 29 位。

扩展帧 ID 号 0x12345678				标准帧 ID 号 0x3FF			
0x12	0x34	0x56	0x78	0x00	0x00	0x03	0xFF

帧数据: 在 CAN 格式转换时, 长度固定为 8 字节。CANFD 格式转换时, 长度固定为 64 字节。有效长度由 DLC3~DLC0 的值决定, 有效数据不足固定长度时, 需要补 0 到固定长度。如表 4.22 所示, 为 CAN 格式转换时 6 字节有效数据的表示方式, 最后两字节补 0。

此模式下, 要注意严格按照固定字节的格式串行数据格式才能转换成功, CAN 模式转换可参考示例如表 4.23 和表 4.24 所示, CANFD 模式时类似, 只是帧数据为固定 64 字节。转换时**首先要确保帧信息无误, 数据长度表示无误, 否则不会进行转换。**

CAN 格式转换每一帧固定是 13 个字节, CANFD 格式转换每一帧固定是 69 个字节。如果不足的必须补 0。同一串行数据帧中满足固定字节格式的串行数据对应一个 CAN(CANFD) 报文, 不足固定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对应的固定字节对齐。如果不足固定字节会进行 100ms 左右的超时, 超时还没接收到够固定字节的数据, 就会把之前接收到的数据清空。

在串行帧转 CAN(CANFD)报文的过程中, 如果以固定字节(CAN 时固定 13 字节, CANFD 时固定 69 字节)对齐的串行数据帧中, 某段固定字节的数据格式不标准, 将会不对固定字节长度进行转换, 接着转换后面的数据。如果转换后发现少了某些 CAN 报文, 请检查对应报文的固定字节长度串行数据格式是否符合标准格式。

表 4.22 数据帧

CAN 格式转换 6 字节有效数据表示方式							
0x11	0x22	0x33	0x44	0x55	0x66	0x00	0x00

表 4.23 CAN 格式转换扩展帧示例

以下例子是一个 CAN 扩展数据帧，ID 为 0x12345678，包含 8 字节有效数据(0x11、0x22、0x33、0x44、0x55、0x66、0x77、0x88)的帧表示方式												
0x88	0x12	0x34	0x56	0x78	0x11	0x22	0x33	0x44	0x55	0x66	0x77	0x88

表 4.24 CAN 格式转换成标准帧示例

以下例子是一个 CAN 标准数据帧，ID 为 0x3FF，包含 6 字节有效数据(0x11、0x22、0x33、0x44、0x55、0x66)的帧表示方式												
0x06	0x00	0x00	0x03	0xFF	0x11	0x22	0x33	0x44	0x55	0x66	0x00	0x00

## 1.4 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(CANFD)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，设置的 CAN(CANFD) 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。

### 1.4.1 帧格式

#### 1. 串行总线帧

串行接口采用的是标准的 Modbus RTU 协议，所以用户帧符合此协议即可。如果传输的帧不符合 Modbus RTU 格式，那么转换器会将接收到的帧丢弃，而不予转换。

转换器采用的 Modbus RTU 传输格式是 1 起始位、8 数据位和 1 停止位。Modbus RTU 帧长度最大为缓冲区长度：5000 字节。

#### 2. CAN(CANFD)总线帧

CAN(CANFD)侧的设备要采用 Modbus 协议则需要为之定义一种可靠的传输格式，这里采用一种分段协议实现，其定义了一个长度大于一个 CAN(CANFD)报文最大数据长度的信息进行分段以及重组的方法。

分段传送协议的制定参考了 DeviceNet 中分段报文的传送协议。CAN 报文时，分段报文格式如表 4.25 (以 CAN 扩展帧为例，标准帧只是帧 ID 的长度不同而已，其他格式

相同)，传输的 Modbus 协议内容即可从“数据 2”字节开始，如果协议内容大于 7 个字节，那么将剩下的协议内容照这种分段格式继续转换，直到转换完成。

表 4.25 CAN 扩展帧分段报文格式

位编号	7	6	5	4	3	2	1	0
帧信息	FF	RTR	0x00	0x00	DLC (数据长度)			
帧 ID3	X	X	X	ID.28 - ID.24				
帧 ID2	ID.23 - ID.16							
帧 ID1	ID.15 - ID.8							
帧 ID0	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							

CANFD 报文时，分段报文格式如表 4.25 (以 CANFD 扩展帧为例，标准帧只是帧 ID 的长度不同而已，其他格式相同)，传输的 Modbus 协议内容即可从“数据 2”字节开始，如果协议内容大于 63 个字节，那么将剩下的协议内容照这种分段格式继续转换，直到转换完成。

表 4.26 CANFD 扩展帧分段报文格式

位编号	7	6	5	4	3	2	1	0
帧信息	FF	RTR	0x00	0x00	DLC (数据长度)			
帧 ID3	X	X	X	ID.28 - ID.24				
帧 ID2	ID.23 - ID.16							
帧 ID1	ID.15 - ID.8							
帧 ID0	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 62	字符 61
数据 63	字符 62
数据 64	字符 63

- ◆ 分段报文标记：表明该报文是否是分段报文。该位为 0 表示单独报文，为 1 表示属于被分段报文中的一帧。
- ◆ 分段类型：表明是第一段、中间段的还是最后段。其值定义如表 4.27 所示。

表 4.27 分段数据类型

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

- ◆ 分段计数器：每一个段的标志，表示该段在整个报文中的序号，如果是第几个段，那么计数器的值就是几。这样在接收时就能够验证是否有分段被遗失。共使用 5Bit，范围为 0~31。

## 1.4.2 转换方式

在串口侧向 CAN(CNAFD) 侧转换的过程中，转换器只会在接收到一完整正确的 Modbus RTU 才会进行转换，否则无动作。

CAN 模式转换时如表 4.28 所示，CANFD 模式转换时如表 4.29 所示，Modbus RTU 协议的地址域转换成 CAN 报文中帧 ID 的 ID0，在转换该帧的过程中标识不变。

表 4.28 CAN 通讯帧相互转换格式 (Modbus 方式)

Modbus RTU 帧		CAN 报文	CAN 报文 1	CAN 报文 x
地址域	↔	帧信息	用户配置	用户配置
功能码		帧 ID3	0x00	0x00
数据域		帧 ID2	0x00	0x00
CRC 域		帧 ID1	0x00	0x00
		帧 ID0	地址域	地址域
		数据 1	[分段协议使用]	[分段协议使用]
		数据 2	功能码	数据域
		数据 3	数据域	
		数据 4		
		数据 5		
	数据 6			
	数据 7			
	数据 8			

表 4.29 CANFD 通信帧相互转化格式 (Modbus 方式)

Modbus RTU 帧		CANFD 报文	CANFD 报文 1	CANFD 报文 x
地址域	↔	帧信息	用户配置	用户配置
功能码		帧 ID3	0x00	0x00
数据域		帧 ID2	0x00	0x00
CRC 域		帧 ID1	0x00	0x00
		帧 ID0	地址域	地址域
		数据 1	[分段协议使用]	[分段协议使用]
		数据 2	功能码	数据域
		数据 3	数据域	
		数据 4		
		...		
	数据 62			
	数据 63			
	数据 64			

Modbus RTU 帧的 CRC 校验字节不转换到 CAN(CANFD) 报文中, CAN(CANFD) 的报文中也不必带有串行帧的校验字节, 因为 CAN(CANFD) 总线本身就有较好的校验机制。

转换的数据是 Modbus RTU 的协议内容的功能码和数据域, 转换时将它们依次转换在 CANFD 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。因为第一字节用来做分段标识, 所以 CAN 报文一帧只能传送 7 个字节数据, CANFD 报文一帧只能传送 63 字节数据。转换器会将较长的 Modbus RTU 帧分段转换成 CAN(CANFD) 的报文后用上述的 CAN(CANFD) 分段协议发出。用户在 CAN(CANFD)的节点上接收时, 根据分段协议重新组合数据再提取功能码和数据域处理即可。

对于 CAN(CANFD)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。

### 1.4.3 转换示例

在配置为 CAN 模式, 设置帧类型为扩展帧的情况下, 如表 4.30 所示, 在 Modbus RTU 帧转换成 CAN 报文时, 将地址域 0x08 直接填充到帧 ID0, 其他帧 ID 填 0x00, 即 CAN ID 为 0x08, 在转换该 Modbus 帧的过程中保持此帧 ID 不变。

如果功能码 + 数据域的数据少于 7 字节则一个单独报文就能发送, 如表 4.31 所示。当一帧 CAN 报文处理不完一帧 Modbus 报文时, CAN 报文采用分段协议。

每个 CAN 报文的“数据 1”都用来填充分段信息 (0x81, 0xC2), 该信息不转换到 Modbus RTU 帧当中, 仅作为帧格式用来确认帧的信息。功能码和数据域的值则依次填入 CAN 报文的数据 2~8 中。

CANFD 模式时与 CAN 类似, 只是数据超过 63 字节才使用分段协议分报文。

表 4.30 CAN 扩展帧通信帧相互转换格式示例 (Modbus 方式分段报文)

Modbus RTU 帧			CAN 报文	CAN 报文 1	CAN 报文 2
地址域	0x08	↔	帧信息	0x18	0x14
功能码	0x11		帧 ID3	0x00	0x00
数据域	0x00		帧 ID2	0x00	0x00
	0x01		帧 ID1	0x00	0x00
	0x00		帧 ID0	0x08	0x08
	0x02		数据 1	0x81	0xC2
	0x04		数据 2	0x11	0x0A

	0x00		数据 3	0x00	0x01
	0x0A		数据 4	0x01	0x02
	0x01		数据 5	0x00	
	0x02		数据 6	0x02	
CRC 域	0xED		数据 7	0x04	
	0x69		数据 8	0x00	

表 4.31 CANFD 扩展帧通信帧相互转换格式示例 (Modbus 方式单独报文)

Modbus RTU 帧			CANFD 报文	CAN 报文 1
地址域	0x08	↔	帧信息	0x16
功能码	0x11		帧 ID3	0x00
数据域	0x00		帧 ID2	0x00
	0x01		帧 ID1	0x00
	0x00		帧 ID0	0x08
	0x02		数据 1	0x00
CRC 域	0x2D		数据 2	0x11
	0x51		数据 3	0x00
			数据 4	0x01
			数据 5	0x00
		数据 6	0x02	



## 2. 免责声明

北京成石创新科技有限公司本着为用户提供更好服务的原则, 在本手册中将尽可能地为 用户呈现详实、准确的产品信息。但鉴于本手册的内容具有一定的时效性, 北京成石不能完全保证该文档在任何时段的时效性与适用性。北京成石有权在没有通知的情况下对本手册上的内容进行更新, 恕不另行通知。为了得到最新版本的信息, 请尊敬的用户定时访问北京成石官方网站或者与北京成石工作人员联系。感谢您的包容与支持!