

CSM330A

隔离 SPI/UART 转 CAN 芯片

UM01010101 V1.00 Date:2023/11/2

类别	内容
关键词	隔离SPI/UART转CAN芯片
摘要	CSM330A产品用户手册

CSM330A

隔离 SPI/UART 转 CAN 芯片

User Manual

修订历史

版本	日期	原因
V1.00	2023/10/01	创建文档

目 录

1. 功能简介	1
1.1 概述	1
1.2 产品特性	1
1.3 产品型号	1
1.4 应用场合	1
2. 硬件说明	2
2.1 产品外观	2
2.2 引脚定义	2
2.3 IO 说明	3
2.4 不带错误反馈硬件电路	4
2.4.1 SPI 转 CAN 硬件电路	4
2.4.2 UART 转 CAN 硬件电路	4
2.5 带错误反馈硬件电路	5
2.5.1 SPI 转 CAN 硬件电路	5
2.5.2 UART 转 CAN 硬件电路	5
2.6 外围保护电路	6
2.7 推荐组网方式	7
3. 产品应用	8
3.1 名词解释	8
3.2 工作模式	9
3.2.1 工作模式切换说明	9
3.2.2 SPI 转 CAN 模式	10
3.2.3 UART 转 CAN 模式	13
3.2.4 SPI 配置模式	14
3.2.5 UART 配置模式	14
3.3 数据转换方式	14
3.3.1 透明转换	14
3.3.2 透明带标识转换	23
3.3.3 自定义协议转换	29
3.3.4 自定义带校验协议转换	34
3.4 错误反馈机制	40
3.4.1 获取错误信息	41
3.4.2 错误信息回应帧	42
4. 产品配置	43
4.1 配置参数	43
4.1.1 转换参数	43
4.1.2 SPI 参数	44
4.1.3 UART 参数	44
4.1.4 CAN 参数	44
4.2 出厂默认配置	45
4.3 配置通信协议	46

4.3.1 写配置参数	47
4.3.2 验证产品硬件标识	51
4.3.3 读配置参数	51
4.4 配置方式	53
4.4.1 MCU 配置方式	53
4.4.2 上位机配置方式	55
5. 产品使用注意事项	56
6. 免责声明	57

1. 功能简介

1.1 概述

CSM330A 支持通过 SPI/UART 扩展出一路 CAN 接口，它内部采用全隔离设计，隔离耐压 3500VDC，内置 DC-DC 隔离电源、信号隔离和 CAN 接口电路隔离，当用户控制板上的 CAN 控制器资源不够时，可以通过 SPI 或 UART 接口扩展出更多的 CAN 总线接口。

该产品可以很方便地嵌入到具有 SPI 或 UART 接口的设备中，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 SPI 设备或 UART 设备和 CAN 总线网络之间的数据通讯。

1.2 产品特性

- ◆ 元器件 100%国产化
- ◆ 实现 SPI 或 UART 与 CAN 接口的双向数据通信；
- ◆ CAN 总线符合“ISO 11898-2”标准；
- ◆ 集成 1 路 SPI 接口，最高可达 2Mbps；
- ◆ 集成 1 路 UART 接口，支持多种速率，最高可达 2Mbps；
- ◆ 集成 1 路 CAN 通讯接口，支持多种速率，最高可达 1Mbps；
- ◆ 隔离耐压 3500VDC；
- ◆ 工作温度：-40℃~+85℃；
- ◆ 电磁辐射 EME 极低；
- ◆ 电磁抗干扰 EMS 极高；

1.3 产品型号

型号	供电	信号电平	SPI 速率	UART 速率	CAN 速率	封装
CSM330A	3.3V	3.3V	0-2Mbit/s	300-2Mbps	5k-1Mbps	DFN22

1.4 应用场合

- ◆ 储能
- ◆ 电池检测
- ◆ 充电桩
- ◆ 轨道交通
- ◆ 重型机械
- ◆ 煤矿
- ◆ 工业自动化
- ◆ 仪器仪表
- ◆

2. 硬件说明

2.1 产品外观

产品外观如图 2.1 所示。



图 2.1 产品外观图

2.2 引脚定义

CSM330A 具有 3 种接口。一种是 SPI 接口，一种是 UART 接口，另外一种是 CAN 接口。产品引脚排列如图 2.2。

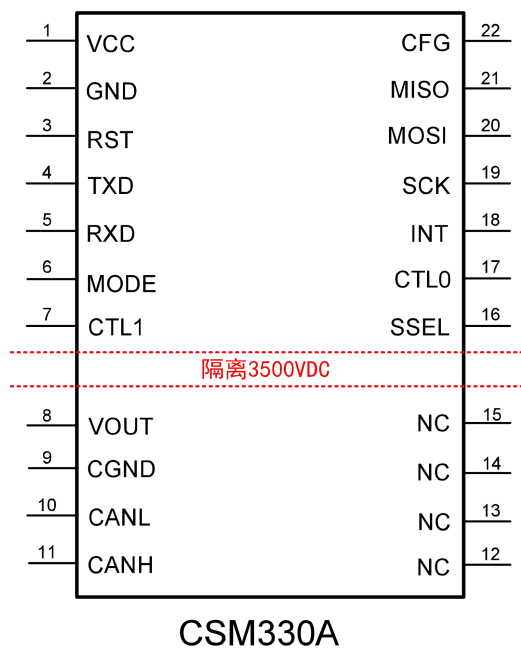


图 2.2 引脚排列

各引脚功能如表 2.1。

表 2.1 引脚功能描述

引脚	名称	功能	引脚	名称	功能
1	VCC	输入电源正	11	CANH	CANH 脚
2	GND	输入电源地	12~15	NC	内部悬空引脚
3	RST	复位脚	16	SSEL	SPI 片选引脚
4	TXD	UART 发送脚 (Error Txd 引脚)	17	CTL0	SPI 主机控制引脚 0
5	RXD	UART 接收脚 (Error Rxd 引脚)	18	INT	从机反馈引脚
6	MODE	模式控制脚	19	SCK	SPI SCK 脚
7	CTL1	SPI 主机控制引脚 1	20	MOSI	SPI MOSI 脚 (Error Rxd 引脚)
8	VOUT	隔离输出电源正	21	MISO	SPI MISO 脚 (Error Txd 引脚)
9	CGND	隔离输出电源地	22	CFG	配置引脚
10	CANL	CANL 脚			

2.3 IO 说明

表 2.2 产品 IO 引脚说明

引脚	名称	类型	说明	引脚	名称	类型	说明
1	VCC	输入	输入电源正	11	CANH	输入/输出	--
2	GND	输入	输入电源地	12~15	NC	内部悬空	--
3	RST	输入	低电平复位, 不支持 5V 容压	16	SSEL	输入	5V 容压
4	TXD	输出	5V 容压	17	CTL0	输入	5V 容压
5	RXD	输入	5V 容压	18	INT	输出	--
6	MODE	输入	5V 容压	19	SCK	输入	5V 容压
7	CTL1	输入	5V 容压	20	MOSI	输入	5V 容压
8	VOUT	输出	输出电源正	21	MISO	输出	--
9	CGND	输出	输出电源地	22	CFG	输入	5V 容压
10	CANL	输入/输出	--				

2.4 不带错误反馈硬件电路

2.4.1 SPI 转 CAN 硬件电路

为保证芯片 CSM330A 输入电源的稳定性，建议用户使用时在输入电源正与地之间并入 4.7μF 的陶瓷电容。使用 SPI 转 CAN 功能时，需要将 MODE 引脚接至高电平。MCU 的 SPI 接口与 CSM330A 的 SPI 接口连接，同时 MCU 需要提供 GPIO 与 RST、INT、CTL0、CTL1 引脚连接，实现对 CSM330A 的有效监测与控制。若需要通过 MCU 对 CSM330A 进行配置，则需要额外的 GPIO 与 CFG 引脚连接。另外，只要不超 VOUT 的最大带载范围该引脚接与不接均不影响产品正常使用。图 2.3 是 CSM330A 在 SPI 转 CAN 模式下不需要错误反馈机制的参考电路。

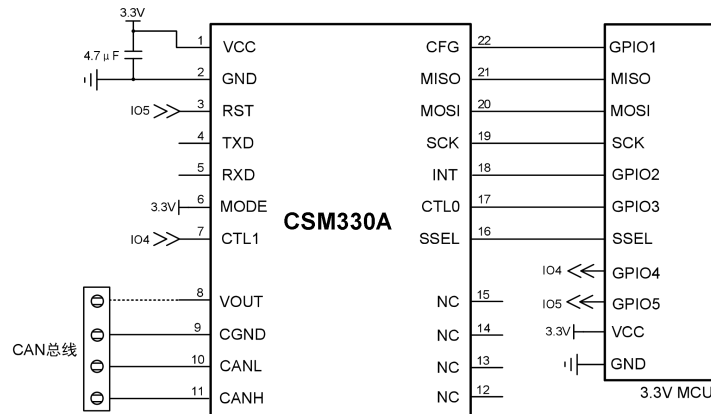


图 2.3 SPI 转 CAN 参考电路

2.4.2 UART 转 CAN 硬件电路

为保证芯片 CSM330A 输入电源的稳定性，建议用户使用时在输入电源正与地之间并入 4.7μF 的陶瓷电容。使用 UART 转 CAN 功能时，需要将 MODE 引脚接至低电平。MCU 的 UART 与 CSM330A 的 UART 接口连接，同时需要一个 GPIO 与 RST 引脚连接。若需要通过 MCU 对 CSM330A 进行配置，则需要额外 GPIO 的与 CFG 引脚连接。另外，只要不超 VOUT 的最大带载范围该引脚接与不接均不影响产品正常使用。图 2.4 是 CSM330A 在 UART 转 CAN 模式下不需要错误反馈机制的参考电路。

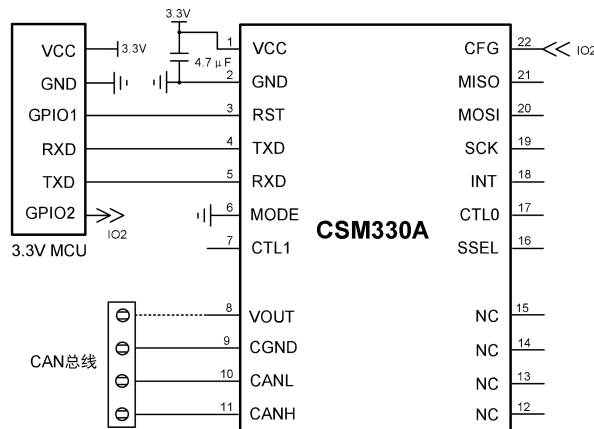


图 2.4 UART 转 CAN 参考电路

2.5 带错误反馈硬件电路

2.5.1 SPI 转 CAN 硬件电路

CSM330A 具备错误反馈机制，用于帮助用户了解通信过程中的错误信息及读取 CAN 错误计数值。如果用户使能此项功能，MCU 的 UART 应与 CSM330A 的 TXD（错误信息发送引脚）、RXD（接收错误命令引脚）连接，如果禁能此项功能应忽略不连接。另外，只要不超 VOUT 的最大带载范围该引脚接与不接均不影响产品正常使用。图 2.5 是 SPI 转 CAN 模式下需要错误反馈机制的连接电路。

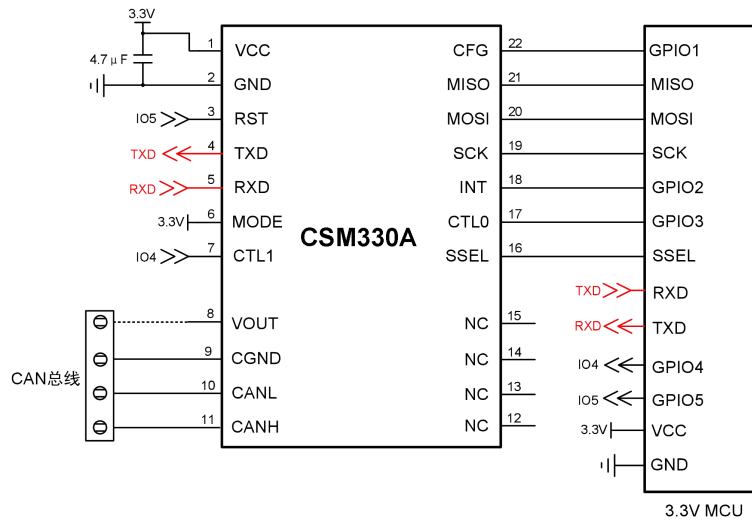


图 2.5 SPI 转 CAN 带错误反馈机制参考电路

2.5.2 UART 转 CAN 硬件电路

CSM330A 具备错误反馈机制，用于帮助用户了解通信过程中的错误信息及 CAN 错误计数值。如果用户使能此项功能，MCU 的 UART 应与 CSM330A 的 MOSI（接收错误命令引脚）、MISO（错误信息发送引脚）连接，如果禁能此项功能应忽略不连接。另外，只要不超 VOUT 的最大带载范围该引脚接与不接均不影响产品正常使用。图 2.6 是 UART 转 CAN 模式下需要错误反馈机制的连接电路。

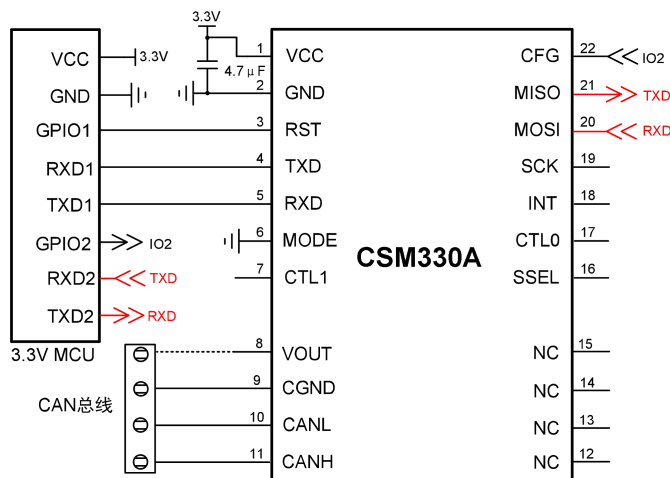


图 2.6 UART 转 CAN 带错误反馈机制参考电路

2.6 外围保护电路

CSM330A 可使用于各种需要使用到 CAN 总线的场合，如果应用环境比较恶劣（如高压电力、雷击等环境），强烈建议用户增加一定的外围保护电路。保护电路可以有效地吸收恶劣环境下引入到电源或总线上的浪涌，保护产品不被损坏。

图 2.7、图 2.8 提供了两个参考外围保护电路。图 2.7 是配合致远电子 SP00S12 信号浪涌抑制器使用的应用电路图。SP00S12 与 CSM330A 之间连接简单，使用方便，占板面积小，SP00S12 的详细参数请参考产品数据手册。而图 2.8 使用了 1 个 TVS 管与 6 个二极管实现端口差模和共模的保护。

表 2.3 的外围保护电路元器件的参数值仅作为参考，请根据实际情况来确定是否需要电路图中的器件，并选取适当的参数值。

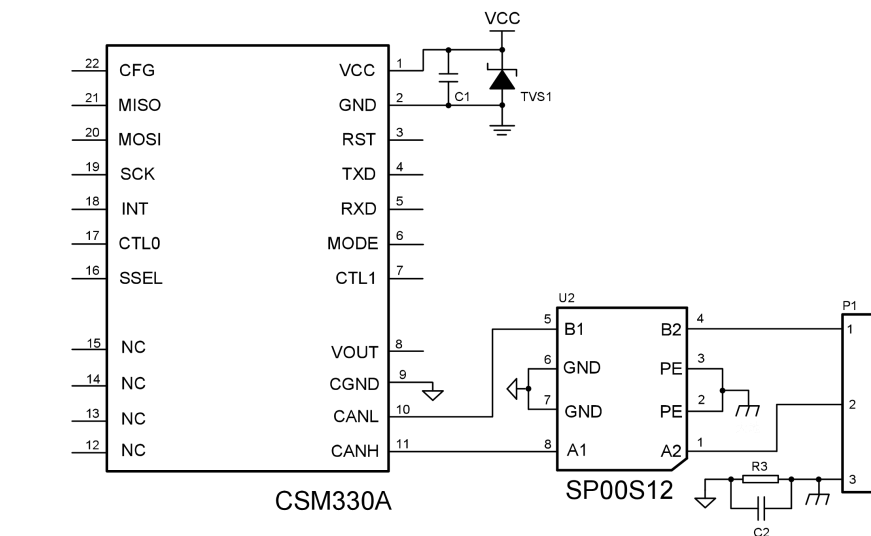


图 2.7 外围保护电路 1

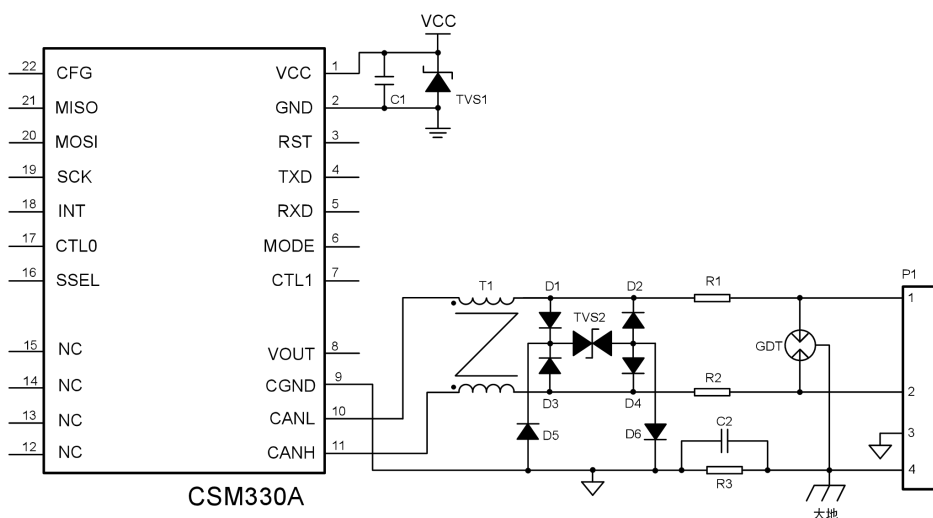


图 2.8 外围保护电路 2

表 2.3 外围保护电路 1 和电路 2 推荐参数表

标号	型号	标号	型号
C1	10μF, 25V	TVS1	SMBJ5.0A
R1, R2	2.7Ω, 2W	TVS2	P6KE15CA
R3	1MΩ, 1206	GDT	B3D090L
C2	102, 2kV	T1	B82793S0513N201
D1~D6	1N4007	U2	SP00S12

2.7 推荐组网方式

CAN 总线一般使用直线型布线方式，总线节点数可达 110 个。布线推荐使用屏蔽双绞线，CANH、CANL 与双绞线线芯连接，CGND 与屏蔽层连接，最后屏蔽层单点接地。无论总线长短，总线两端都需要连接终端电阻，电阻大小可根据实际布线进行调整，一般推荐值为 120Ω。图 2.9 给出了推荐组网示意图。

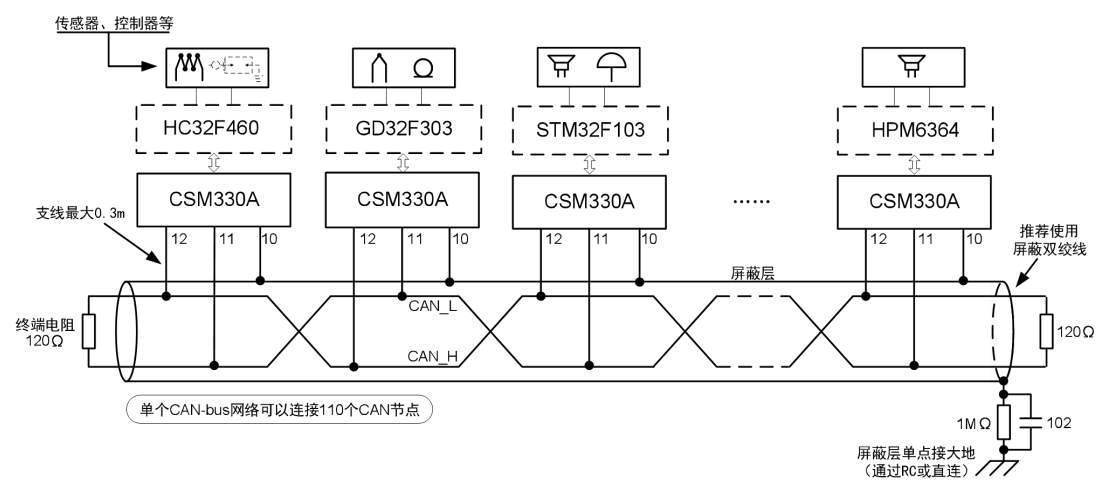


图 2.9 推荐组网示意图

3. 产品应用

3.1 名词解释

1. SPI

SPI 是串行外设接口（Serial Peripheral Interface）的缩写。SPI 是一种高速的、全双工、同步的通信总线。

2. UART

UART 是通用异步接收/发送装置（Universal Asynchronous Receiver/Transmitter）的缩写。UART 是一种通用串行数据总线，可实现全双工的串行异步通信。

3. CAN 总线

CAN 是控制器局域网（Controller Area Network）的缩写。CAN 总线属于现场总线的范畴，是一种有效支持分布式控制或实时控制的串行通信网络。

4. 串行帧

即串行总线帧，是 SPI 总线通信帧（下文简称 SPI 帧）、UART 总线通信帧（下文简称 UART 帧）的统称。

5. CAN 帧

即 CAN 总线帧，是 CAN 接口标准帧、扩展帧的统称。

6. 标准帧

CAN 帧的类型，标准帧的帧 ID 共 11 位，范围为：0x000-0x7ff。

7. 扩展帧

CAN 帧的类型，扩展帧的帧 ID 共 29 位，范围为：0x00000000-0x1fffffff。

8. 错误反馈机制

当 CSM330A 检测到 UART/SPI 的 CRC 校验错误，帧尾错误，CAN 错误（计数到 128 以上）时，会将 Error Txd（错误信息发送引脚）拉低保持在低电平通知 MCU，MCU 检测到低电平后，可向 Error Rxd（接收错误命令引脚）发送读错误命令，CSM330A 接收到该命令后，将通过 Error Txd 引脚发出错误信息。错误信息发出后错误寄存器值自动清零。

9. 透明转换

CSM330A 的一种数据传输方式，指串行总线（SPI/UART）、CAN 总线之间的数据不经过处理立即转换传输。

10. 透明带标识转换

CSM330A 的一种数据传输方式，在透明转换的基础上，增加总线标识（ID）的处理。串行总线向 CAN 总线转换时，串行帧的“标识”（ID）决定 CAN 帧的标识（ID），反之，CAN 总线向串行总线转换时，CAN 帧的标识（ID）决定串行帧的“标识”（ID）。

11. 自定义协议转换

CSM330A 的一种数据传输方式，自定义协议转换方式下，串行帧必须符合规定的帧格式。有效的串行帧由帧头、帧长度、帧类型、帧 ID、数据域、帧尾组成。

12. 自定义带校验协议转换

CSM330A 的一种数据传输方式，在自定义协议转换的基础上增加了 CRC 校验字节，用于判断 SPI/UART 端数据是否正确。串行帧必须符合规定的帧格式，有效的串行帧由帧头、帧长度、帧类型、帧 ID、数据域、CRC 校验域、帧尾组成。

3.2 工作模式

3.2.1 工作模式切换说明

CSM330A 上电后，MODE、CFG 引脚电平会决定产品处于 4 种不同的工作模式的其中一种：SPI 转 CAN 模式、UART 转 CAN 模式、SPI 配置模式、UART 配置模式。表 3.1 罗列了不同的引脚电平时，产品所处在的工作模式。

表 3.1 CSM330A 工作模式汇总表

引脚电平			工作模式
MODE	CFG	RST	
1	1	1	SPI 转 CAN
0	1	1	UART 转 CAN
1	0	1	SPI 配置
0	0	1	UART 配置
X	X	0	复位

若需要切换产品的工作模式，更改引脚电平后，必须对产品进行复位，才能使其进入设定的工作模式。需要注意的是，为保证产品成功复位，复位保持时间最少为 100us，复位后，产品初始化等待时间最少 200ms，待产品初始化完成后，才能进行正常操作如图 3.1 所示，图 3.2 是产品工作模式切换示意图。

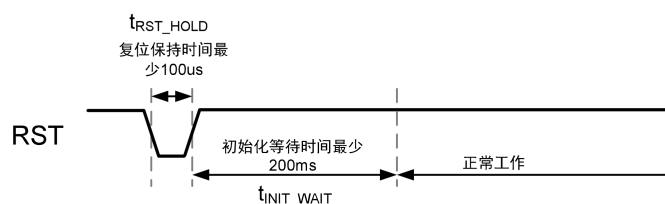


图 3.1 复位时序示意图

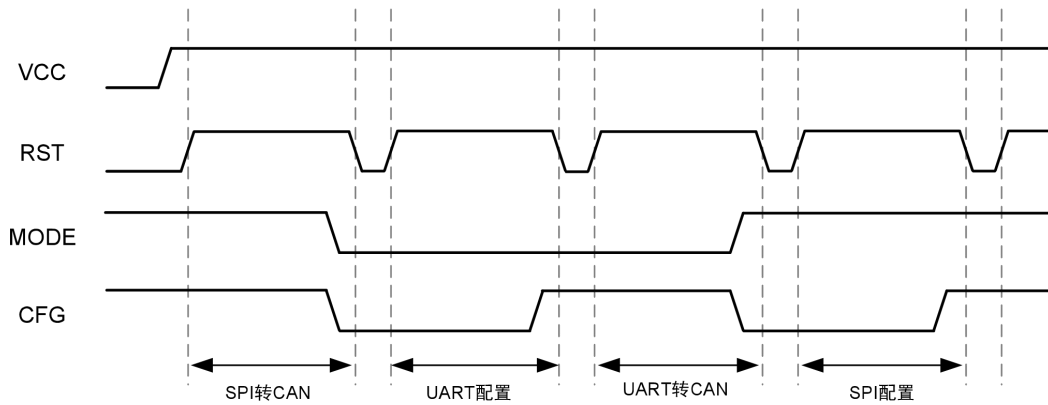


图 3.2 工作模式切换时序

3.2.2 SPI 转 CAN 模式

在此工作模式下，CSM330A 始终作为 SPI 从机，SPI 限定工作在模式 3（CPOL、CPHA 均为 1），数据长度限定为 8 位，MSB 高位先传输。SPI 的最高速率可达 2Mbps。

SPI 主机可以发送数据至 CAN 总线端，且可接收 CAN 总线端收到的数据。此时 UART 接口无效，若禁能错误反馈机制将不会处理任何出现在 UART 接口的数据，也不会返回 CAN 总线端接收到的数据至 UART。

1. SPI 帧

SPI 一次片选有效至片选无效之间的数据定义为一帧数据。读数据和写数据帧定义如图 3.3、图 3.4。帧与帧之间读写缓冲区数据应有 40us 的时间间隔，如图 3.5。

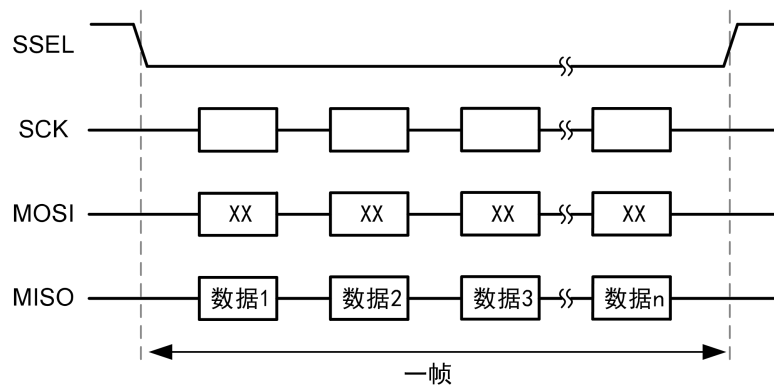


图 3.3 主机读数据帧示意

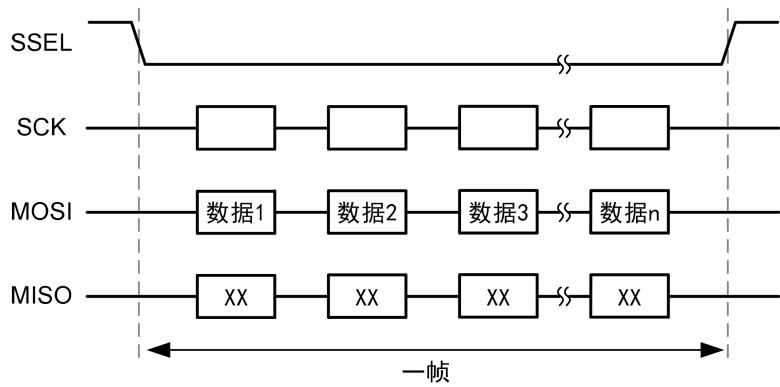


图 3.4 主机写数据帧示意

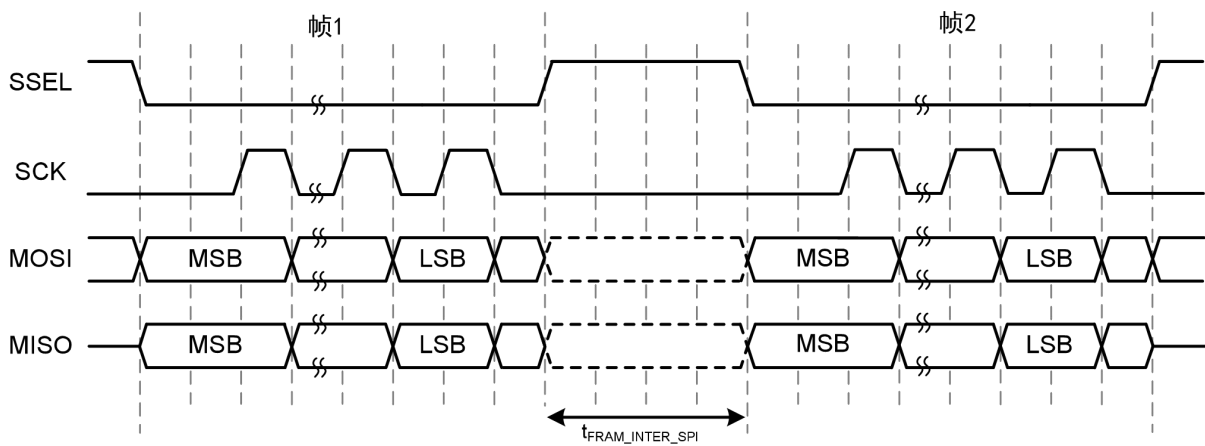


图 3.5 SPI 帧间隔示意图

2. 主机控制

CSM330A 有两个 SPI 主机控制引脚 CTL0, CTL1, 受主机端控制。主机通过控制 CTL0, CTL1 引脚, 使 CSM330A 进入不同的功能状态, 实现对 CSM330A 不同操作目的。主机端控制引脚电平不同对应功能如表 3.2 所示。

表 3.2 SPI 模式下主机控制功能

CTL0	CTL1	功能
0	0	空闲
0	1	主机读状态
1	0	主机读数据
1	1	主机写数据

主机可以通过读从机当前状态来获取产品当前可以读取的字节数以及可以写入的字节数。主机将功能选择为主机读状态, 然后通过 SPI 读出 4 个字节, 即为状态码。状态码由 32 个位构成, 具体定义如表 3.3 所示。

表 3.3 SPI 模式状态码构成

位	含义	符号	描述
0	可读标识位	read	CAN 接收缓冲区不为空时，该位为 1，反之为 0。
12:1	可读字节数	read_bytes	主机可以从 CSM330A 中读取的 CAN 数据字节数，可表示最大值为 0xFFF，若实际可读字节数超过表示的最大范围，则仍表示为 0xFFF。
13	可写标识位	write	CAN 发送缓冲区还未填满时，该位为 1，反之为 0。
25:14	可写字节数	write_bytes	主机可以写入 CSM330A 的串行字节数。
31:26	保留位	reserved	保留。

若定义 status[] 数组为 8 位整型，通过 SPI 读状态依次读出的数据为 status[0]、status[1]、status[2]、status[3]，则其数据结构如图 3.6。

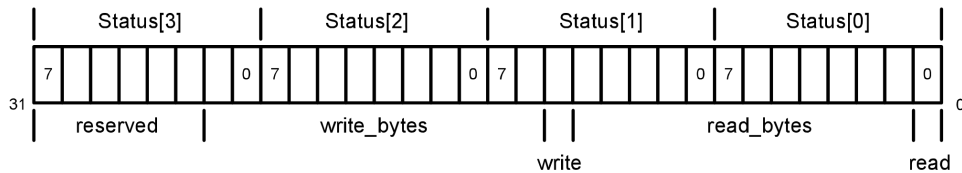


图 3.6 状态字节数据结构

用户在获得这 4 个字节后，应该将对应位分离出来，用作后续处理的判断基准。示例代码如下：

```

read = status [0] & 0x01; //分离出可读标识位
read_bytes = ((status [0] & 0xFE) >> 1) + ((status [1] & 0x1F) << 7); //分离出可读字节数
write = (status [1] & (1 << 5)) ? 1 : 0; //分离出可写标识位
write_bytes = ((status [1] & 0xC0) >> 6) + ((status [2] & 0xFF) << 2) + ((status [3] & 0x03) << 10); //分离出可写字节数

```

在实际使用中，CSM330A 的 CAN 发送缓冲区中的数据会很快发送出去，所以 write_bytes 一般都会恢复到最大值。

当主机不需要读写 CSM330A 时，应该将控制状态切换为空闲状态。主机通过 CTL0，CTL1 引脚切换状态后，必须经过至少 50us 才能片选使能 CSM330A 进行读写操作；特别的，进行写操作后，需要保持写状态至少 5us，才能保证数据正确转换至 CAN 总线。如图 3.7。

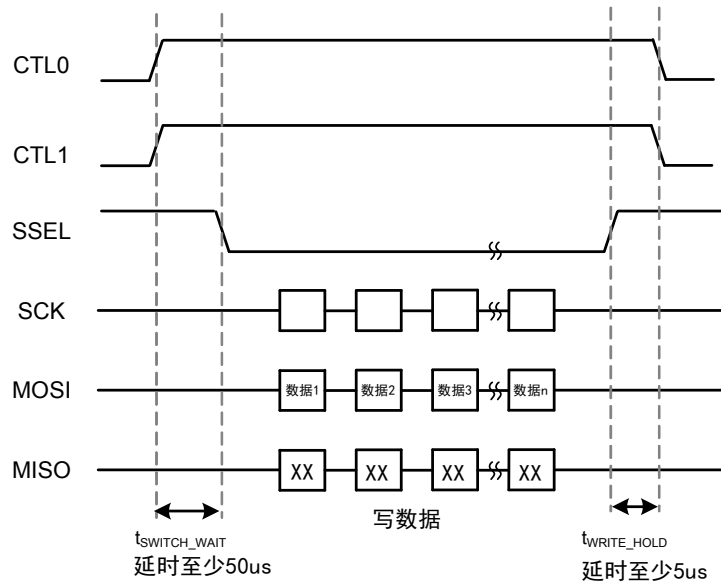


图 3.7 主机控制功能切换延时示意图

3. 反馈机制

CSM330A 只能作为 SPI 从机，不能主动地控制其他 SPI 总线设备，如果 CSM330A 每次收到 CAN 端发送过来的数据时，都需要主机来查询状态的话，会使得整个通讯过程的效率很低，因此我们为其添加了一个反馈机制。

CSM330A 硬件上有一个 INT 反馈引脚，此引脚与主机连接，出现以下两种情况时，INT 引脚会由高电平变成低电平，通知主机进行读数据操作（为避免数据丢失，建议主机使用低电平触发方式检测）：

(1) CAN 缓冲区 CAN 帧数达到设置的触发点时

当产品 CAN 总线端接收缓冲区接收到的 CAN 帧数达到触发点时，INT 引脚电平置低，直到缓冲区清空，INT 引脚才会恢复高电平。用户可以在获得 INT 信号之后查询 CSM330A 的状态，获取可读字节数，然后读取缓冲区 CAN 数据。

(2) CAN 缓冲区数据少于触发帧数，且在设定时间内主机未读取时

CAN 缓冲区有数据但少于触发帧数时，若总线长时间未有新增数据，且主机未进行读取操作时，CAN 接收缓冲区的数据将有可能长期得不到处理，这就导致数据的实时性不高。为了解决少量数据的实时性问题，CSM330A 内部设置了一个计时器，若 CAN 缓冲区的数据在一定时间内未被读取，将触发 INT 引脚置低，通知主机读取数据。CSM330A 在接收到最后一帧数据时，计时器启动，主机进行读取操作时复位计时器。

3.2.3 UART 转 CAN 模式

在此模式下，CSM330A 只能通过 UART 向 CAN 总线端发送或接收数据。UART 通信格式固定为：1 起始位，8 数据位，1 停止位，不可更改。UART 的通信速率范围为 300bps~2Mbps。此模式下，若禁能错误反馈机制 SPI 接口将无效，不会处理任何出现在 SPI 接口的数据，也不会返回 CAN 总线端接收到的数据至 SPI。

1. UART 帧

UART 从接收到第一个数据开始，直到等待 n 个字符（此参数通过用户设置）时间后仍未收到新的数据为止，这段时间内的数据定义为一帧数据，这段时间定义为“帧间隔”。由于字符时间随波特率的变化而变化，故不同波特率下，相同字符个数的帧间隔时间也不同。例如，用户设置 UART 帧间隔为 2 个字符时间，波特率为 9600，由于每个字符由 10 位组成，则间隔时间为 $2 \times 10 / 9600 = 2.083\text{ms}$ 。

总线出现第一个数据，代表着一帧的起始，等待大于 n 个字符时间前的最后一个数据，为帧的最后一个数据；等待大于 n 个字符时间仍未有新数据出现，代表着一个帧的结束。UART 通信帧格式如图 3.8。

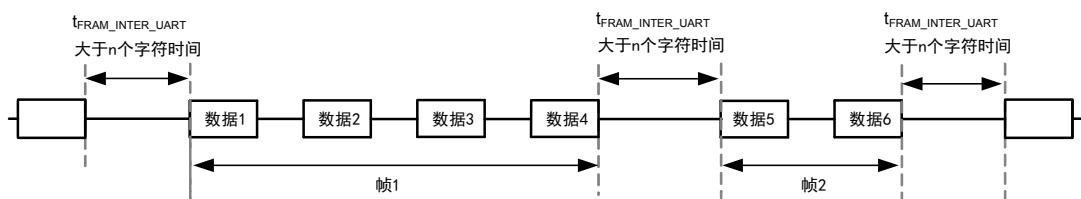


图 3.8 UART 通信帧格式

3.2.4 SPI 配置模式

在此模式下，CSM330A 处于等待配置状态，无法向 CAN 端发送或接收数据。此模式下仅能通过 SPI 接口进行配置，具体配置说明请参考第 4.4.1 节。

3.2.5 UART 配置模式

在此模式下，CSM330A 处于等待配置状态，无法向 CAN 端发送或接收数据。此模式下仅能通过 UART 接口进行配置，具体配置说明请参考第 4.4.1 节。

3.3 数据转换方式

CSM330A 的数据转换方式，指串行总线和 CAN 总线数据之间转换的基本规则。同一时间内，产品只能工作于一种数据转换方式，若需要改变数据转换方式，则需要更改配置。产品配置好一种转换方式时，即同时作用于 SPI 转 CAN 模式，及 UART 转 CAN 模式。

CSM330A 的数据转换方式有四种：透明转换、透明带标识转换、自定义协议转换、自定义带校验协议转换。

3.3.1 透明转换

透明转换，是指任何一侧总线只要接收到数据，即立刻发送至另一侧总线上，数据不做任何处理。

透明转换方式下，CSM330A 无需对数据进行额外处理，最大限度地提高了数据转换速度，也提高了缓冲区的利用率，因为在接收的同时 CSM330A 也在转换并发送，又空出了可以接收的缓冲区。透明转换方式下，数据转换的详细说明如下文。

1. 串行帧向 CAN 帧转换（SPI/UART→CAN）

串行帧的全部数据依次填充到 CAN 帧的数据域里。当产品检测到串行总线上有数据后就立即接收并转换。

由于 CAN 每帧的最大数据长度是 8 个字节，所以，当串行帧的数据长度小于等于 8 个字节时，数据通过一个 CAN 帧转发出去。若串行帧的数据长度大于 8 个字符，产品从串行帧首个字符开始，每一次取 8 个字符通过一个 CAN 帧转发出去，直到所有数据全部转发完成，此时一个串行帧数据转换为多个 CAN 帧数据。

数据转换示意如图 3.9，图 3.10 所示。CAN 帧只标示出以下有用信息：帧类型、帧 ID、数据长度以及数据域。其中 CAN 帧中的“帧类型”及“帧 ID”，由用户配置决定，且始终保持不变，除非用户重新对产品进行配置。CAN 帧中的“数据长度”根据实际分配到 CAN 帧的数据的字节数决定。

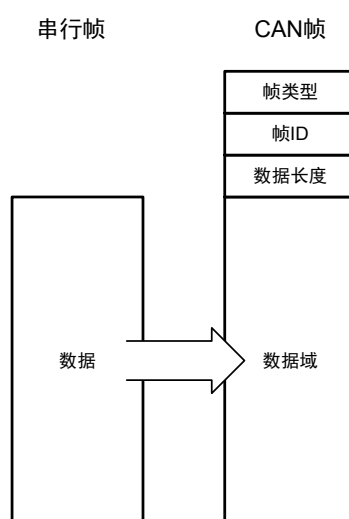


图 3.9 串行帧向 CAN 帧转换（透明转换，数据小于等于 8 个字节）

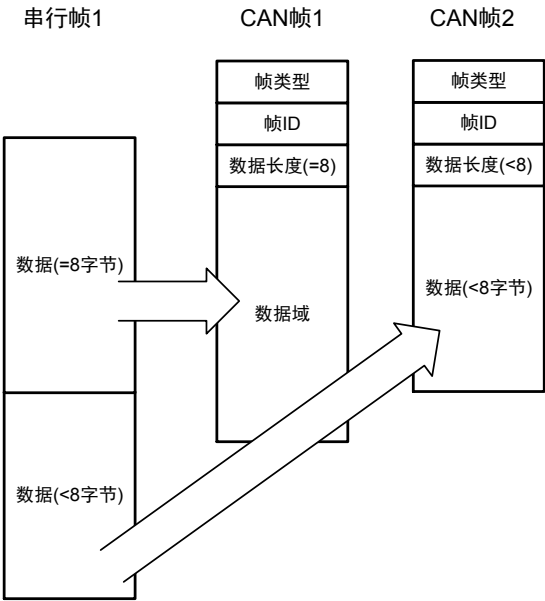


图 3.10 串行帧向 CAN 帧转换（透明转换，数据大于 8 个字节）

转换实例

例 1：假设用户配置转换成的 CAN 帧的帧类型为“标准帧”，帧 ID1, ID0 分别为“0x01, 0x23”，串行帧的数据为 0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07, 0x08 共 8 个字节，那么串行帧转 CAN 帧格式如图 3.11 所示。

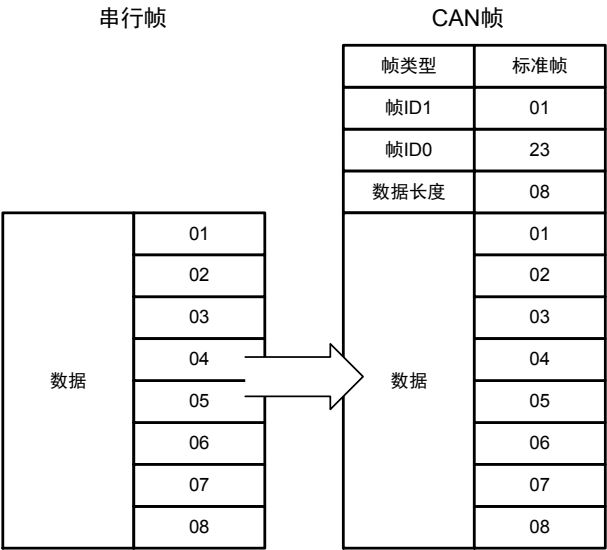


图 3.11 串行帧转 CAN 帧示例 1（透明转换）

例 2：假设用户配置转换成的 CAN 帧的帧类型为“扩展帧”，帧 ID1, ID0 分别为“0x01, 0x23”，串行帧的数据为 0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07, 0x08, 0x09, 0x0A, 0x0B, 0x0C, 0x0D 共 13 个字节，那么串行帧转 CAN 帧格式如图 3.12 所示。

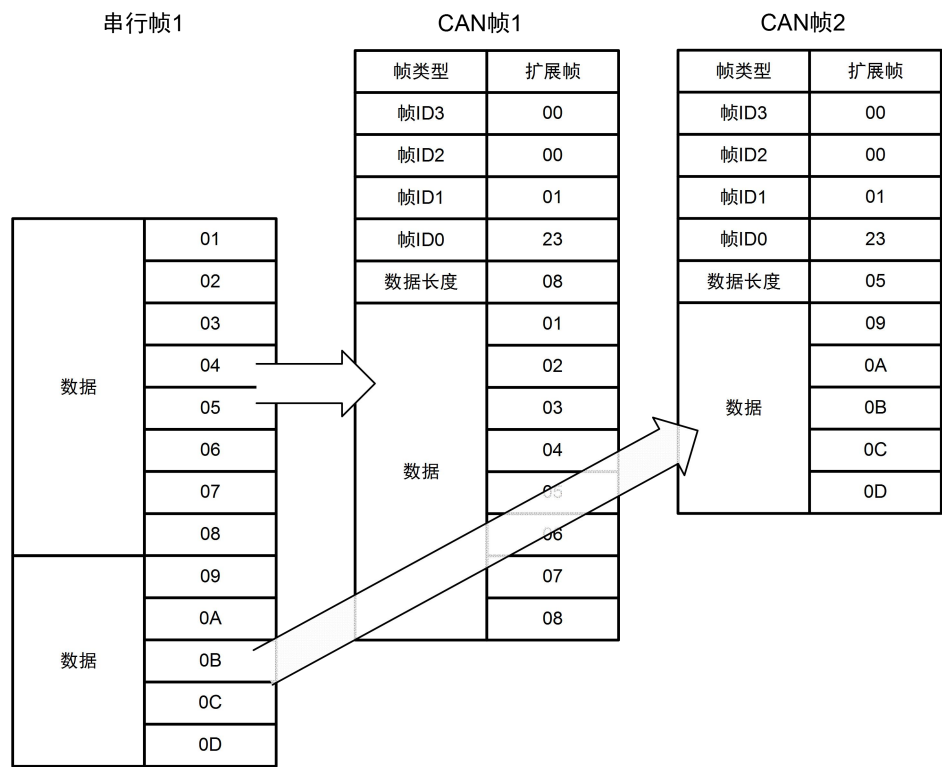


图 3.12 串行帧转 CAN 帧示例 2（透明转换）

2. CAN 帧转 UART 帧（CAN→UART）

CSM330A 从 CAN 总线收到一帧数据后就立即转发到 UART 接口。转换时，产品会将 CAN 帧数据域中的数据依序全部转换到 UART 帧中，每个 CAN 帧数据转换为一个 UART 帧数据，此时每一个串行帧的数据长度与所转发的 CAN 帧长度一致。数据格式对应如图 3.13 所示。

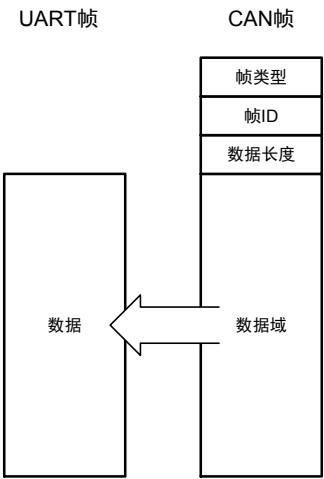


图 3.13 CAN 帧转 UART 帧（透明转换）

(1) 帧信息转换使能

帧信息，是指 CAN 帧的“帧类型”和“数据长度”信息，长度为一个字节。

帧信息转换使能，即 CAN 帧转换成 UART 帧时，同时将 CAN 帧的“帧类型”和“数据长度”信息，以一个特定意义的字节表示，将此字节作为 UART 帧的起始字节，转发至 UART 接口，数据紧随其后。

帧信息以一个字节表示，其中 7~4 位指代“帧类型”，“0000”代表“标准帧”，“1000”代表“扩展帧”；3~0 位指代“数据长度”，“0000~1000”分别代表“0~8”位数据。

例如，帧信息为 0x05，即表示 CAN 帧类型为标准帧，CAN 帧数据长度为 5 个字节；

帧信息为 0x86，即表示 CAN 帧类型为扩展帧，CAN 帧数据长度为 6 个字节。

若用户使能帧信息转换，转换的数据模式如图 3.14。

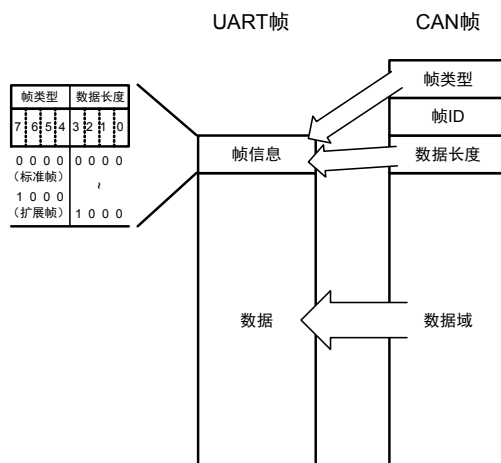


图 3.14 CAN 帧转 UART 帧（透明转换，帧信息使能）

(2) 帧 ID 转换使能

帧 ID，即 CAN 帧的“标识（ID）”。

帧 ID 转换使能，即 CAN 帧转换成 UART 帧时，同时将 CAN 帧的 ID 以 2 个字节（标准帧）或 4 个字节（扩展帧）转发至 UART 接口，ID 字节先于数据字节发送，其中 ID 字节高位在前。帧 ID 转换使能时，默认使能帧信息转换，先发送帧信息，再发送帧 ID，最后发送数据。

若用户使能帧 ID 转换，其转换数据格式如图 3.15。

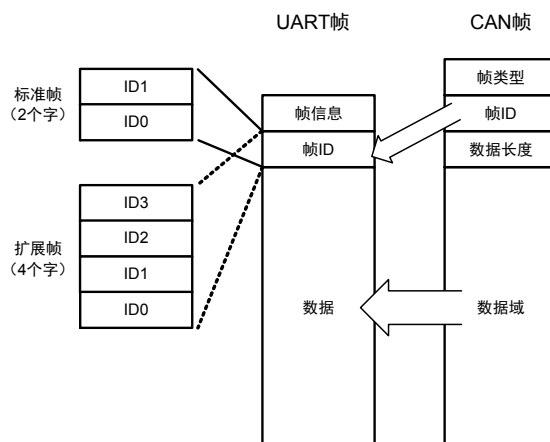


图 3.15 CAN 帧转 UART 帧（透明转换，帧 ID 转换使能）

假如 CAN 帧数据长度为 0，需要转换帧类型或帧 ID 时，会作对应转换，但是数据域为空；若不需要转换帧类型或帧 ID 时，则不做转换。

转换实例

例 1：假设用户配置如下：帧信息转换使能，帧 ID 转换使能。若 CAN 接口接收到的 CAN 帧为扩展帧，帧 ID 为 0x00000123，数据为 0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07，则 CAN 帧和转换后的 UART 帧如图 3.16 所示。

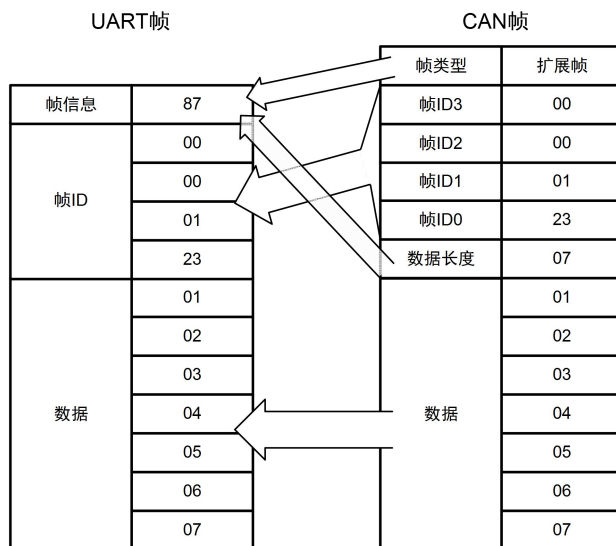


图 3.16 CAN 帧转 UART 帧示例（透明转换，帧信息、帧 ID 转换使能）

3. CAN 帧转 SPI 帧（CAN→SPI）

CSM330A 作为 SPI 从机，无法主动控制 SPI 外设，当产品从 CAN 总线收到一帧数据后，只能立即存储到 CAN 接收缓冲区。当 CAN 缓冲区的 CAN 帧数达到反馈触发帧数，或触发时间，INT 引脚输出低电平通知 SPI 主机进行数据读取。

当 SPI 主机从 CSM330A 获取 CAN 缓冲区的大小后，SPI 可以通过一帧读取出 CAN 缓冲区中包含的所有 CAN 帧数据如图 3.17。需要注意的是 CSM330A 可读缓冲区的大小我们用 12bit 来表示，因此能够表示的最大值仅为 0xFFF，若实际可读字节数超过表示的最大范围，用户获取到的可读字节数仍为 0xFFF，此种情况下用户可先读 0xFFF 个数据，再依次读出缓冲区剩余数据。

为避免缓存区数据量超过 0xFFF 后可能会发生读取最后一帧数据不完整的情况，建议用户可以先读取 0xFFF 个数据，再读取剩下的数据，读完后再将数据一起做处理；或者用户将第一次读取的数据先做处理，将最后一个不完整的帧做保留，将后续 CAN 数据读取完合并后再一起做处理。

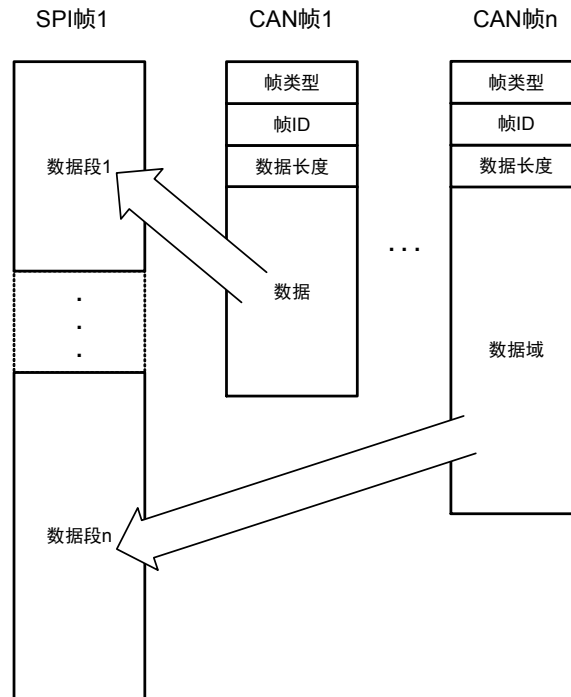


图 3.17 CAN 帧转 SPI 帧（透明转换）

(1) 帧信息转换使能

帧信息，是指 CAN 帧的“帧类型”和“数据长度”信息，长度为一个字节。

帧信息转换使能时，CAN 接收缓冲区的数据是帧信息和帧数据的集合。CAN 端接收到一个 CAN 数据帧时，产品将帧信息和帧数据同时放置到缓冲区。帧信息字节和帧数据字节组成一个“字节段”，每个字节段包括一个独立 CAN 帧的信息及数据，其中帧信息作为字节段的首个字节，数据紧随其后。SPI 主机获取 CAN 接收缓冲区大小后，可通过一个 SPI 帧一次读取出存储在缓冲区的 n 个字节段。

帧信息以一个字节表示，其中 7~4 位指代“帧类型”，“0000”代表“标准帧”，“1000”代表“扩展帧”；3~0 位指代“数据长度”，“0000~1000”分别代表“0~8”位数据。

例如，帧信息为 0x05，即表示 CAN 帧类型为标准帧，CAN 帧数据长度为 5 个字节；

帧信息为 0x86，即表示 CAN 帧类型为扩展帧，CAN 帧数据长度为 6 个字节。

若用户使能帧信息转换，转换的数据模式如图 3.18。

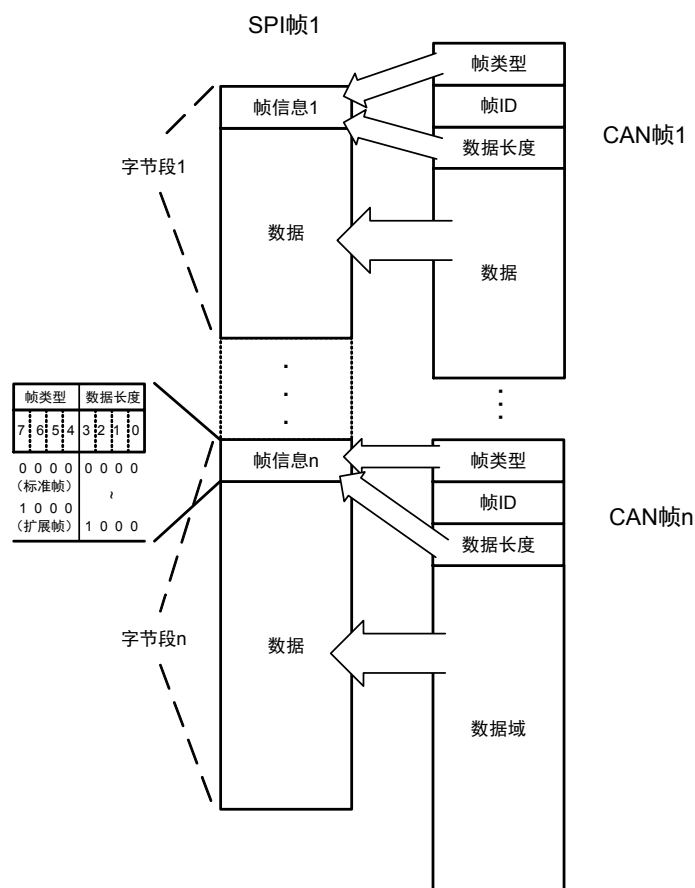


图 3.18 CAN 帧转 SPI 帧（透明转换，帧信息转换使能）

(2) 帧 ID 转换使能

帧 ID，即 CAN 帧的“标识（ID）”。

SPI 转 CAN 模式下，考虑一个 SPI 帧数据大部分时候均是多个 CAN 帧数据的集合，为了有效区分 CAN 帧 ID 与数据，帧 ID 转换使能后，默认开启帧信息转换使能，CAN 接收缓冲区是帧信息、帧 ID 和帧数据的集合。CAN 端接收到一个 CAN 数据帧时，产品将帧信息、帧 ID 和帧数据同时放置到缓冲区。CAN 帧的 ID 以 2 个字节（标准帧）或 4 个字节（扩展帧）进行存储，帧信息字节、帧 ID 字节和帧数据字节组成一个“字节段”，每个字节段包括一个独立 CAN 帧的信息、ID 及数据。其中帧信息最前，接着是帧 ID，帧 ID 高位在前，数据在帧 ID 字节之后。SPI 主机获取 CAN 接收缓冲区大小后，可通过一个 SPI 帧一次读取出存储在缓冲区的 n 个字节段。

若用户使能帧 ID 转换，其转换数据格式如图 3.19。由于默认使能帧信息转换，一个字节段按顺序包含帧信息字节、帧 ID 字节、帧数据字节。

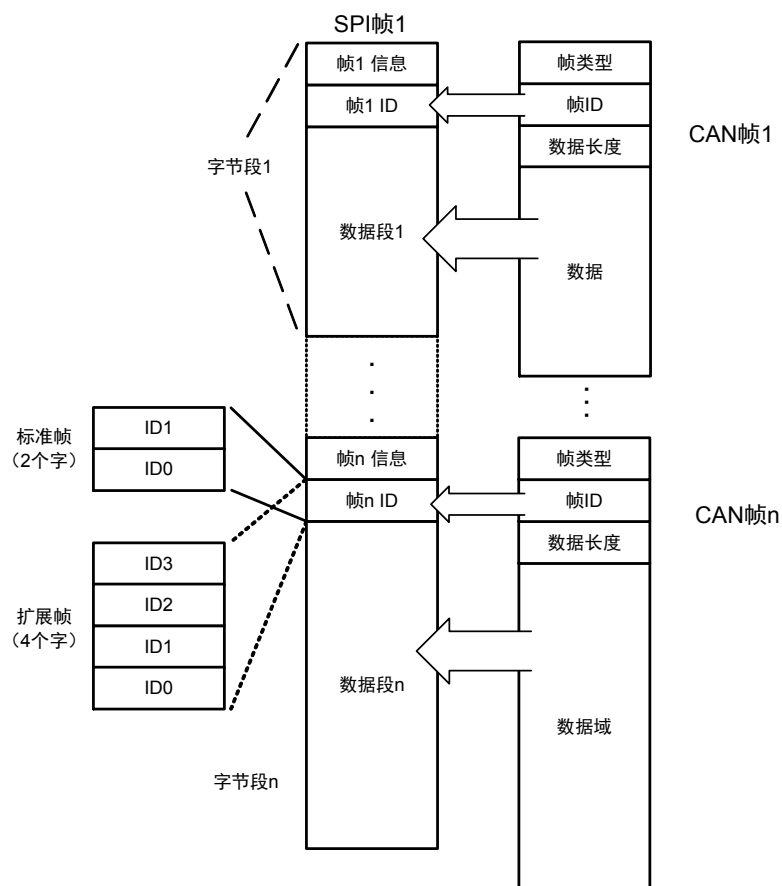


图 3.19 CAN 帧转 SPI 帧（透明转换，帧 ID 转换使能）

转换实例

例 1：假设用户配置如下：帧信息转换使能，帧 ID 转换使能。若 CAN 接口接收到的 CAN 帧为标准帧，首帧 ID 为 0x0123，数据为 0x01, 0x02, 0x03, 0x04, 0x05；第 n 帧 ID 为 0x0234 数据为 0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07；则 CAN 帧和转换后的 SPI 帧如图 3.20。

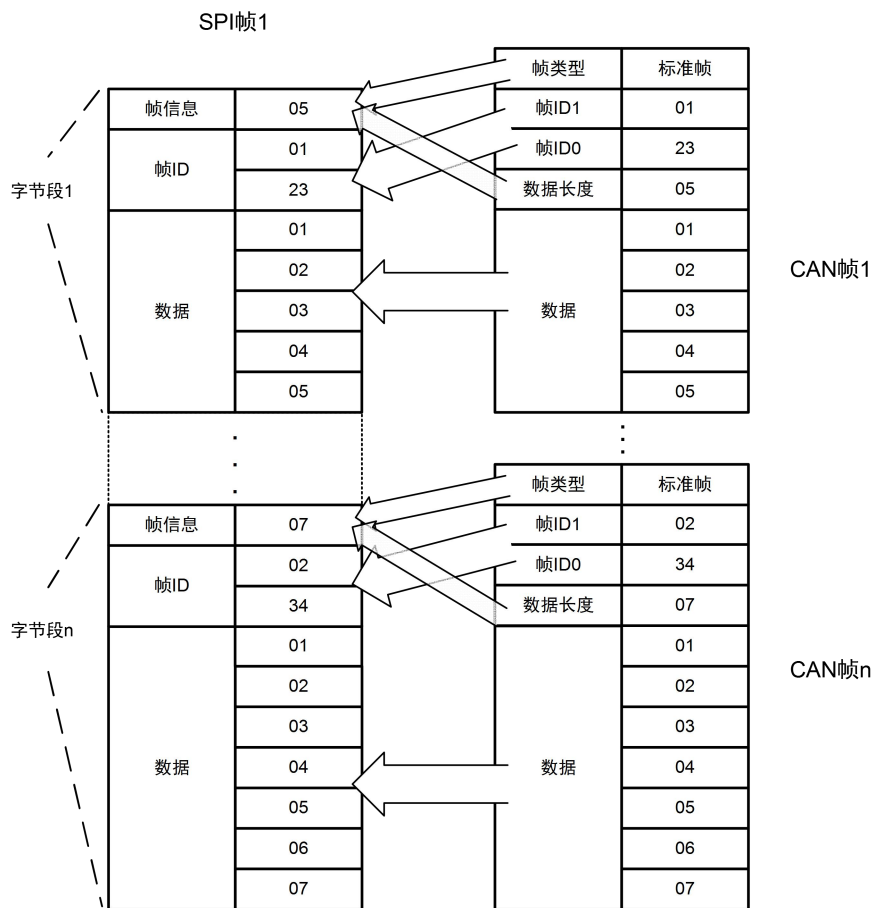


图 3.20 CAN 帧转 SPI 帧示例 1（透明转换，帧信息、帧 ID 使能）

3.3.2 透明带标识转换

透明带标识转换，是在透明转换的基础上衍生而来，是指发送或接收的串行帧中都包含了有效的 CAN 帧 ID 字节。

该方式把串行帧中的“帧 ID”自动转换成 CAN 帧中的帧 ID。只要告诉产品该“帧 ID”在串行帧的起始位置和长度，产品在转换时提取出这个“帧 ID”填充在 CAN 帧的帧 ID 域里，作为该串行帧转发时的 CAN 帧的帧 ID。在 CAN 帧转换成串行帧的时候，产品把 CAN 帧的帧 ID 转换在串行帧的相应位置，同时返回帧信息。

透明带标识转换方式下，用户可以控制串行帧所发数据的 CAN 帧 ID，方便用户在同一节点发送不同 ID 的 CAN 数据。用户也可以在接收到的串行帧中获取隐藏的 CAN 帧 ID。

1. 串行帧转 CAN 帧（SPI/UART→CAN）

串行帧中所带有的 CAN 帧的“帧 ID”在串行帧中的“起始地址”和“长度”可由用户配置设定。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）。若起始地址设置为 1，长度为 2，则串行帧中第 1，2 个字节（从 0 开始计算）作为 CAN 帧 ID。

转换时根据用户的配置信息将串行帧中的 CAN 帧“帧 ID”对应全部转换到 CAN 帧的帧 ID 域中（如果所带 ID 字节数少于 CAN 帧的帧 ID 字节数，那么在 CAN 帧的填充顺序是

帧 ID3~ID0，并将余下的 ID 填为 0），其它的数据依序转换，如图 3.21 所示。

由于 CAN 每帧的最大数据长度是 8 个字节，所以，当串行帧的数据长度小于等于 8 个字节时，数据通过一个 CAN 帧转发出去。如果一帧 CAN 帧未将串行帧数据转换完，则仍然用相同的 ID 作为 CAN 帧的帧 ID 继续转换直到将串行帧转换完成。

数据转换示意如图 3.21。CAN 帧只标示出以下有用信息：帧类型、帧 ID、数据长度以及数据域。其中 CAN 帧中的“帧类型”、串行帧中“帧 ID”的起始地址、长度，由用户配置决定，且始终保持不变，除非用户重新对产品进行配置。CAN 帧中的“数据长度”根据实际分配到 CAN 帧的数据的字节数决定。

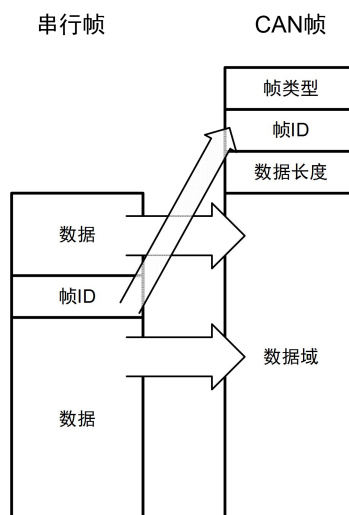


图 3.21 串行帧转 CAN 帧（透明带标识转换，数据小于等于 8 个字节）

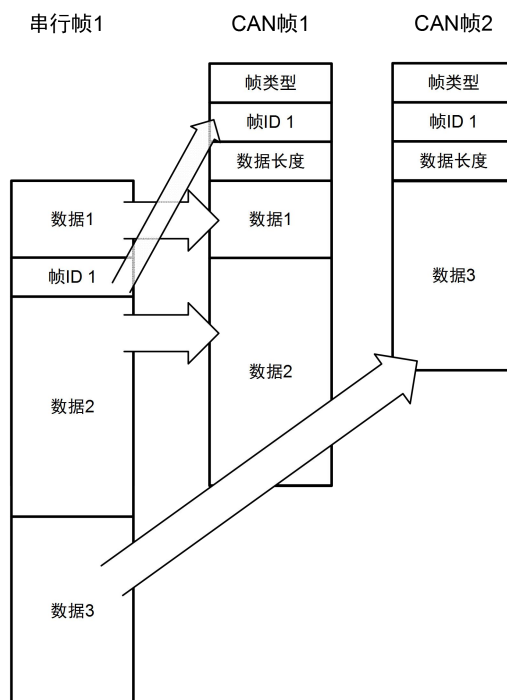


图 3.22 串行帧转 CAN 帧（透明带标识转换，数据大于 8 个字节）

转换实例

例 1：假设用户配置转换成的 CAN 帧的帧类型为“扩展帧”，“帧 ID”在串行帧中的起始地址是 2，长度是 3，串行帧发送的数据分别为 0x00, 0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07, 0x08, 0x09, 0x0A, 0x0B, 0x0C, 0x0D, 0x0E, 0x0F，那么串行帧转 CAN 帧格式如图 3.23 所示。

由于配置的 CAN 帧为扩展帧，所以在串行帧转 CAN 帧中的帧 ID 为 4 个字节。由于配置串行帧中的起始地址是 2，长度是 3，因此从串行帧中第 2 个字节开始，连续取 3 个字节，在本例中分别为 0x02, 0x03, 0x04，产品在转换时依次将这三个字节填充到 CAN 帧的帧 ID3、帧 ID2、帧 ID1，但帧 ID0 被填充为 00。串行帧中其余的数据将不作任何修改地转换到 CAN 帧的数据域。

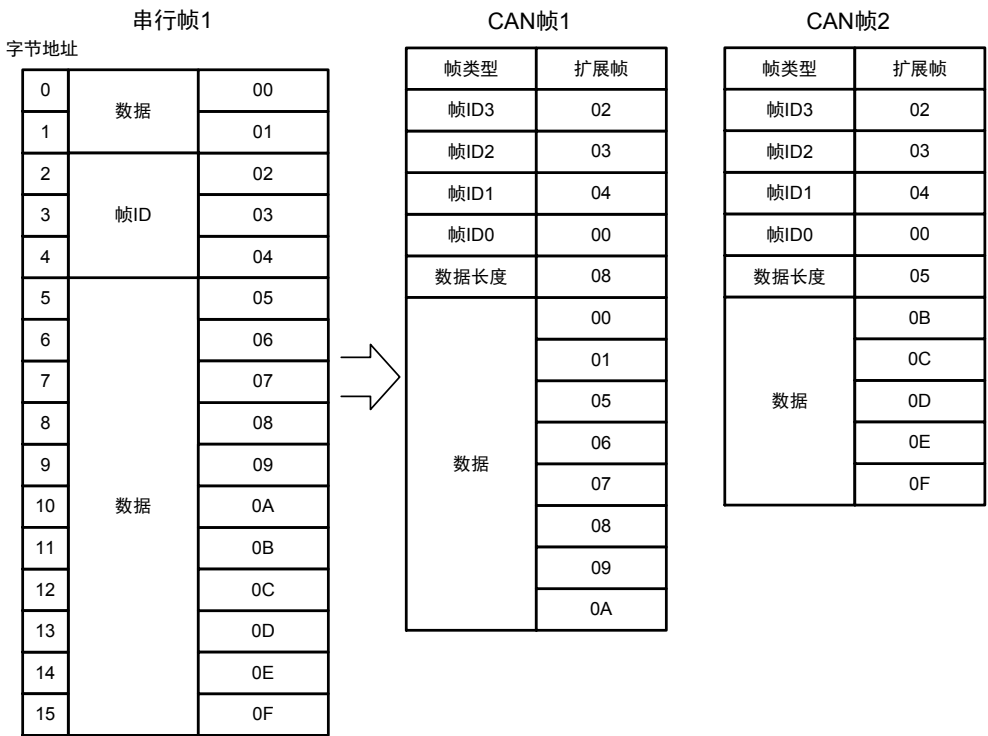


图 3.23 串行帧转 CAN 帧示例（透明带标识转换）

2. CAN 帧转 UART 帧（CAN→UART）

对于 CAN 帧，也是收到一帧就立即向 UART 接口转发一帧，每次转发的时候也根据用户配置的 CAN 帧 ID 在 UART 帧中的位置和长度，把接收到的 CAN 帧中的 ID 作相应的转换，其它数据依序转发。在这种模式下，CAN 帧的帧信息默认转换到 UART 帧的首字节。如图 3.24 所示。

帧信息以一个字节表示，其中 7~4 位指代“帧类型”，“0000”代表“标准帧”，“1000”代表“扩展帧”；3~0 位指代“数据长度”，“0000~1000”分别代表“0~8”位数据。

例如，帧信息为 0x05，即表示 CAN 帧类型为标准帧，CAN 帧数据长度为 5 个字节；
帧信息为 0x86，即表示 CAN 帧类型为扩展帧，CAN 帧数据长度为 6 个字节。

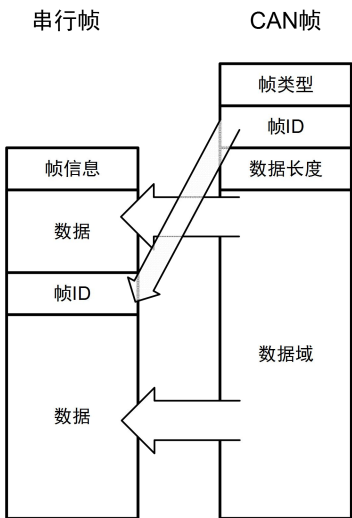


图 3.24 CAN 帧转 UART 帧（透明带标识）

转换实例

例 1：假设用户配置转换成的 CAN 帧的帧类型为“扩展帧”，“帧 ID”在 UART 帧中的起始地址是 2，长度是 3。接收到的 CAN 帧的帧 ID 为 0x00123456，数据为 0x01，0x02，0x03，0x04，0x05，0x06，0x07，0x08。转换格式如图 3.25 所示。

CAN 帧中的帧 ID3、帧 ID2、帧 ID1 依次转换到串行帧为第 2、3、4 字节，CAN 帧的数据域将不作任何修改地依次转换到 UART 帧中未被填充部分。

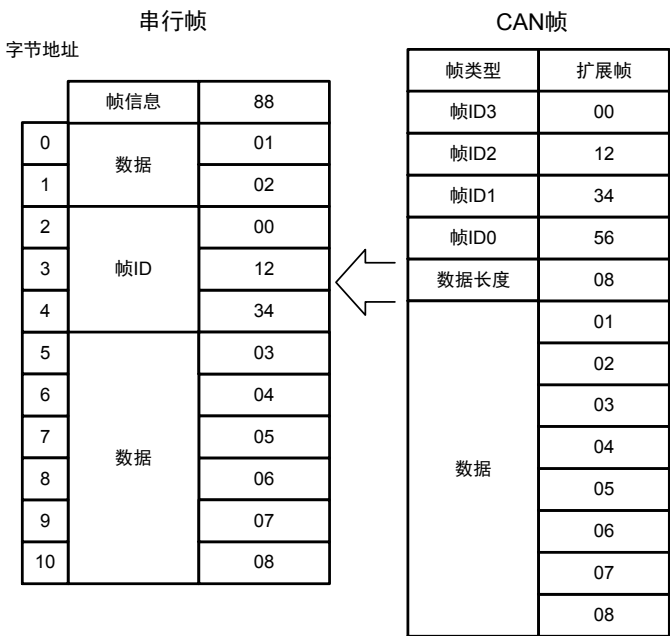


图 3.25 CAN 帧转 UART 帧示例（透明带标识方式）

3. CAN 帧转 SPI 帧 (CAN→SPI)

CSM330A 作为 SPI 从机，无法主动控制 SPI 外设，当产品从 CAN 总线收到一帧数据后，只能立即存储到 CAN 接收缓冲区。当 CAN 缓冲区的 CAN 帧数达到反馈触发帧数，或触发时间，INT 引脚输出低电平通知 SPI 主机进行数据读取。

当 SPI 主机从 CSM330A 获取 CAN 缓冲区的大小后，SPI 可以通过一帧读取出 CAN 缓冲区中包含的所有 CAN 帧数据。需要注意的是 CSM330A 可读缓冲区的大小我们用 12bit 来表示，因此能够表示的最大值仅为 0xFFF，若实际可读字节数超过表示的最大范围，用户获取到的可读字节数仍为 0xFFF，此种情况下用户可先读取 0xFFF 个数据，再依次读出缓冲区剩余数据。为避免缓存区数据量超过 0xFFF 后可能会发生读取最后一帧数据不完整的情况，建议用户可以先读取 0xFFF 个数据，再读取剩下的数据，读完后再将数据一起做处理；或者用户将第一次读取的数据先做处理，将最后一个不完整的帧做保留，将后续 CAN 数据读取完合并后再一起处理。

SPI 转 CAN 模式下，考虑一个 SPI 帧数据大部分时候均是多个 CAN 帧数据的集合，为了有效区分 CAN 帧 ID 与数据，CAN 接收缓冲区是帧信息、帧 ID 和帧数据的集合。透明带标识转换方式下，CAN 接收缓冲区的是帧信息、帧 ID 和帧数据的集合。CAN 端接收到一个 CAN 数据帧时，产品将帧信息、帧 ID 和帧数据同时放置到缓冲区。CAN 帧信息作为首个字节进行存储，帧 ID 按用户配置的起始地址及长度在指定的位置进行存储。帧信息、帧 ID 字节和帧数据字节组成一个“字节段”，每个字节段包括一个独立 CAN 帧的帧信息、帧 ID 及数据，其中帧 ID 高位在前。

帧信息以一个字节表示，其中 7~4 位指代“帧类型”，“0000”代表“标准帧”，“1000”代表“扩展帧”；3~0 位指代“数据长度”，“0000~1000”分别代表“0~8”位数据。

例如，帧信息为 0x05，即表示 CAN 帧类型为标准帧，CAN 帧数据长度为 5 个字节；

帧信息为 0x86，即表示 CAN 帧类型为扩展帧，CAN 帧数据长度为 6 个字节。

SPI 主机获取 CAN 接收缓冲区大小后，可通过一个 SPI 帧一次读取出存储在缓冲区的 n 个字节段。其转换数据格式如图 3.26 所示。

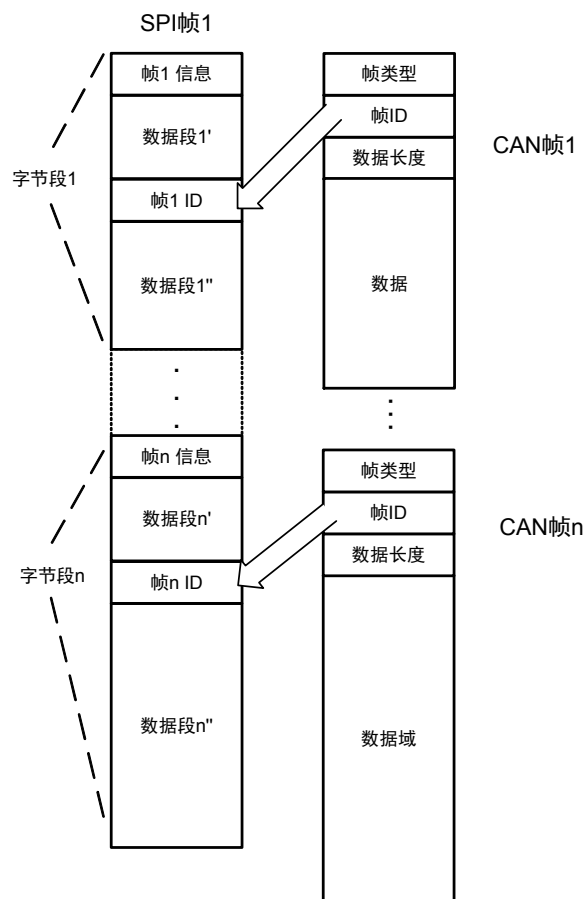


图 3.26 CAN 帧转 SPI 帧（透明带标识）

转换实例

例 1：假设用户配置转换成的 CAN 帧的帧类型为“标准帧”，“帧 ID”在 SPI 帧中的起始地址是 1，长度是 2。若 CAN 接口接收到的 CAN 帧为标准帧，首帧 ID 为 0x0123，数据为 0x01, 0x02, 0x03, 0x04, 0x05；第 n 帧 ID 为 0x0234 数据为 0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07；则 CAN 帧和转换后的 SPI 帧如图 3.27。

CAN 帧中的帧 ID1、帧 ID0 依次转换到每个字节段的第 2、3 字节，CAN 帧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。

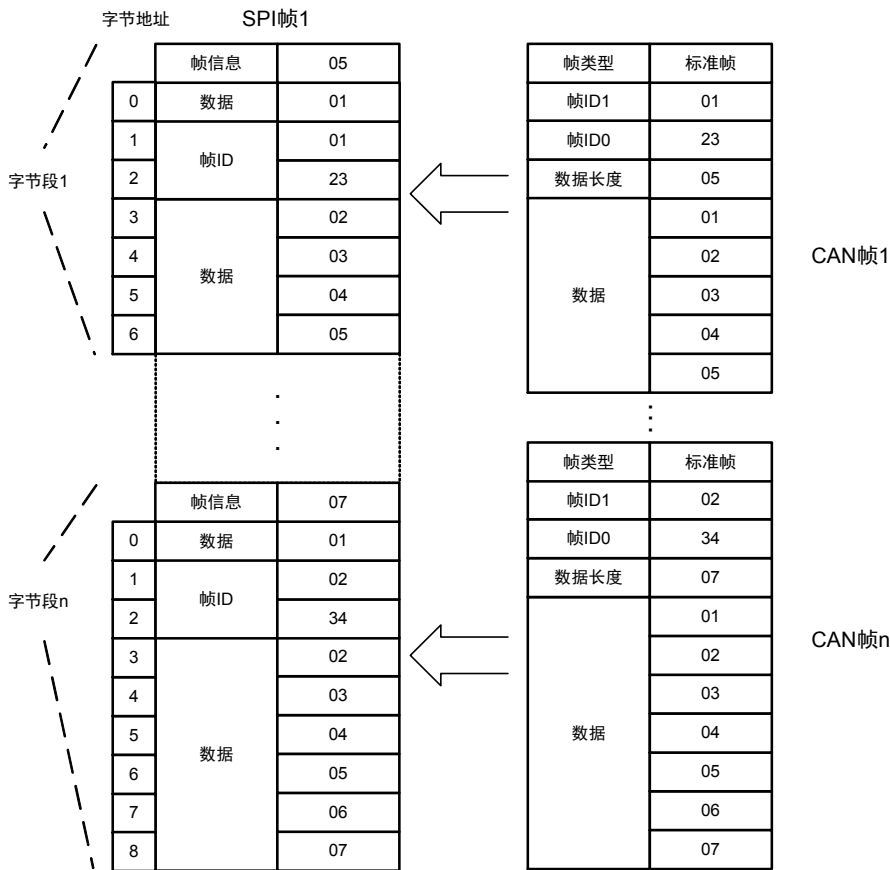


图 3.27 CAN 帧转 SPI 帧示例（透明带标识）

3.3.3 自定义协议转换

自定义协议转换方式下，串行帧必须符合规定的帧格式。有效的串行帧由帧头、帧长度、帧类型、帧 ID、数据域、帧尾。

帧头，是一个串行帧的开始字节，范围为 0x00~0xFF，由用户配置。

帧长度，指帧类型、帧 ID、数据域总共包含的字节数。

帧类型，指发送 CAN 帧的帧类型。0x00 代表标准帧，0x08 代表扩展帧。

帧 ID，指发送 CAN 帧的帧 ID。帧类型为 0x00 时，帧 ID 由 2 个字节组成，高位在前；帧类型为 0x08 时，帧 ID 由 4 个字节组成，高位在前。

数据域，是需要发送的数据，转换至 CAN 帧的数据域。

帧尾，是一个串行帧的结束字节，范围为 0x00~0xFF，由用户配置。

当用户发送的串行帧完全符合定义的格式时，CSM330A 才会接收串行帧的数据并进行转发，否则不作任何处理直接丢弃。

1. 串行帧转 CAN 帧（SPI/UART→CAN）

串行接口接收到有效的串行帧后，“帧类型”确定将要发送 CAN 帧的帧类型，“帧 ID”作为 CAN 帧 ID，“数据域”填充至 CAN 帧数据域。如图 3.28、图 3.29 为串行帧转 CAN 帧示意图。其中图 3.28 串行帧数据域字节数小于等于 8，数据全部通过一个 CAN 帧转发；

图 3.29 串行帧数据域字节数大于 8，数据通过多个 CAN 帧转发。

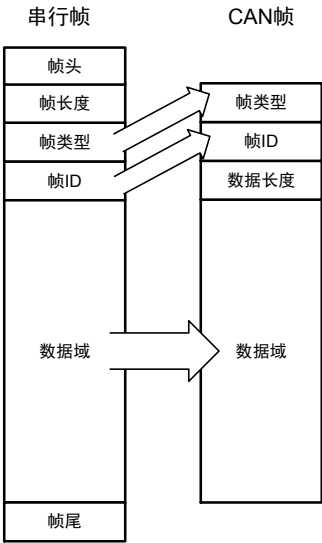


图 3.28 串行帧转 CAN 帧 1（自定义协议转换）

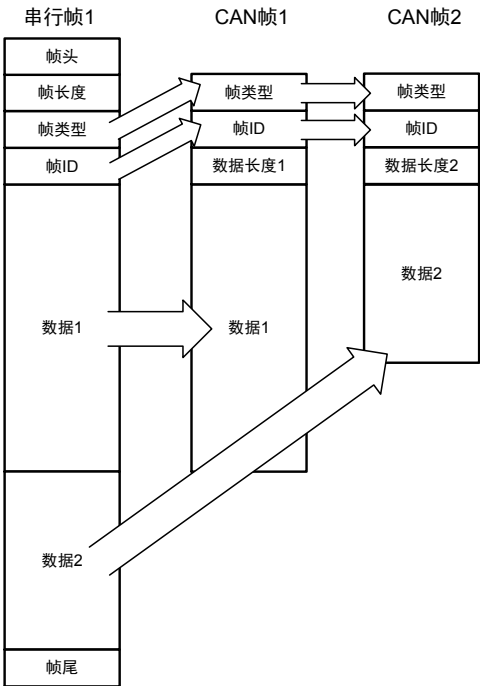


图 3.29 串行帧转 CAN 帧 2（自定义协议转换）

转换实例

例 1: 假设用户配置的串行帧头为 0x40, 帧尾为 0x1A。用户发送帧类型为标准帧(0x00), 帧 ID 为 0123, 数据为 0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07, 0x08, 0x09, 0x0A, 0x0B, 则帧长度为 0x0E。转换示例如图 3.30。

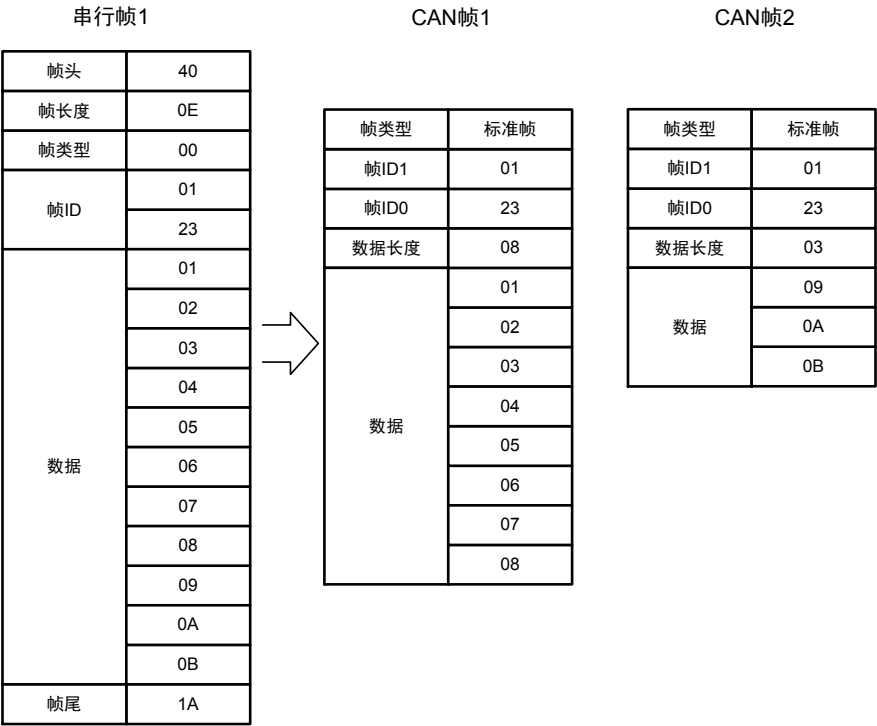


图 3.30 串行帧转 CAN 帧示例（自定义协议转换）

2. CAN 帧转 UART 帧

UART 转 CAN 模式下，CAN 总线收到一帧 CAN 帧即立即转发一帧。数据格式对应如图 3.31 所示，转换时，CAN 帧类型转换到 UART 帧帧类型字节，帧 ID 转换到 UART 帧帧 ID 字节，CAN 帧数据域中的数据依序全部转换到串行帧数据域中。

为了保证串行帧的完整性，在 CAN 帧转换为串行帧时，帧头、帧尾与用户设置一致，这时的串行帧完全与串行帧转 CAN 帧时的帧格式一致。

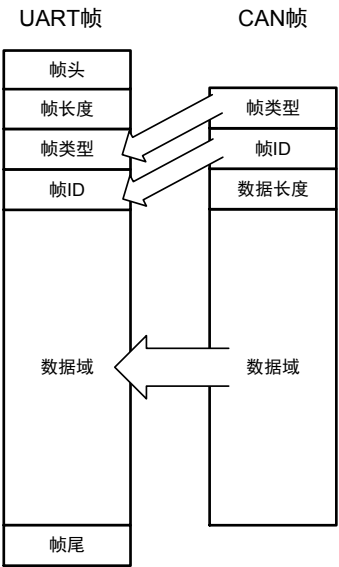


图 3.31 CAN 帧转 UART 帧（自定义协议转换）

转换实例

例 1：假设用户配置的串行帧头为 0x40，帧尾为 0x1A。转换示例如图 3.32。

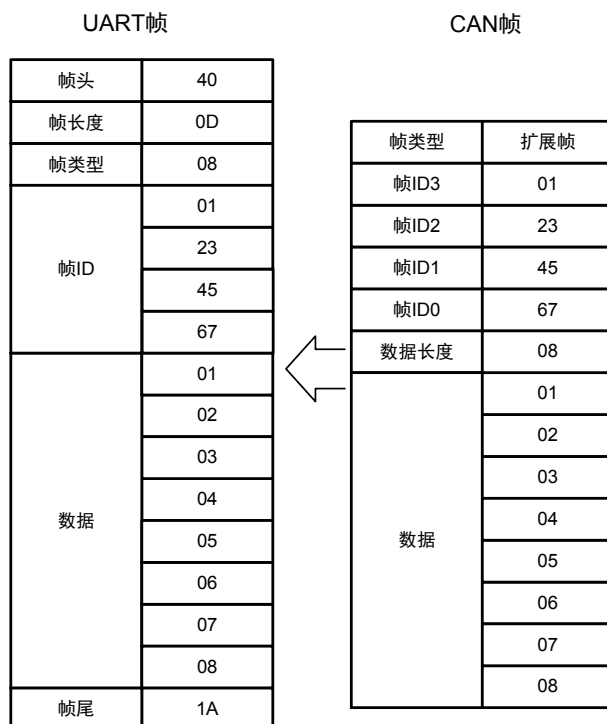


图 3.32 CAN 帧转 UART 帧示例（自定义协议转换）

3. CAN 帧转 SPI 帧（CAN→SPI）

CSM330A 作为 SPI 从机，无法主动控制 SPI 外设，当产品从 CAN 总线收到一帧数据后，只能立即存储到 CAN 接收缓冲区。当 CAN 缓冲区的 CAN 帧数达到反馈触发帧数，或触发时间，INT 引脚输出低电平通知 SPI 主机进行数据读取。

当 SPI 主机从 CSM330A 获取 CAN 缓冲区的大小后，SPI 可以通过一帧读取出 CAN 缓冲区中包含的所有 CAN 帧数据。需要注意的是 CSM330A 可读缓冲区的大小我们用 12bit 来表示，因此能够表示的最大值仅为 0xFFF，若实际可读字节数超过表示的最大范围，用户获取到的可读字节数仍为 0xFFF，此种情况下用户可先读取 0xFFF 个数据，再依次读出缓冲区剩余数据。

自定义转换方式下，CAN 接收缓冲区存储的是多个有效格式字节段的集合。CAN 端接收到一个 CAN 数据帧时，将其转换为符合用户定义格式的字节段放置到缓冲区。为避免缓存区数据量超过 0xFFF 后可能会发生读取最后一帧数据不完整的情况，建议用户可以先读取 0xFFF 个数据，再读取剩下的数据，读完后再将数据一起做处理；或者用户将第一次读取的数据先做处理，将最后一个不完整的帧做保留，将后续 CAN 数据读取完合并后再一起做处理。

SPI 主机获取 CAN 接收缓冲区大小后，可通过一个 SPI 帧一次读取出存储在缓冲区的 n 个字节段。其转换数据格式如图 3.33。

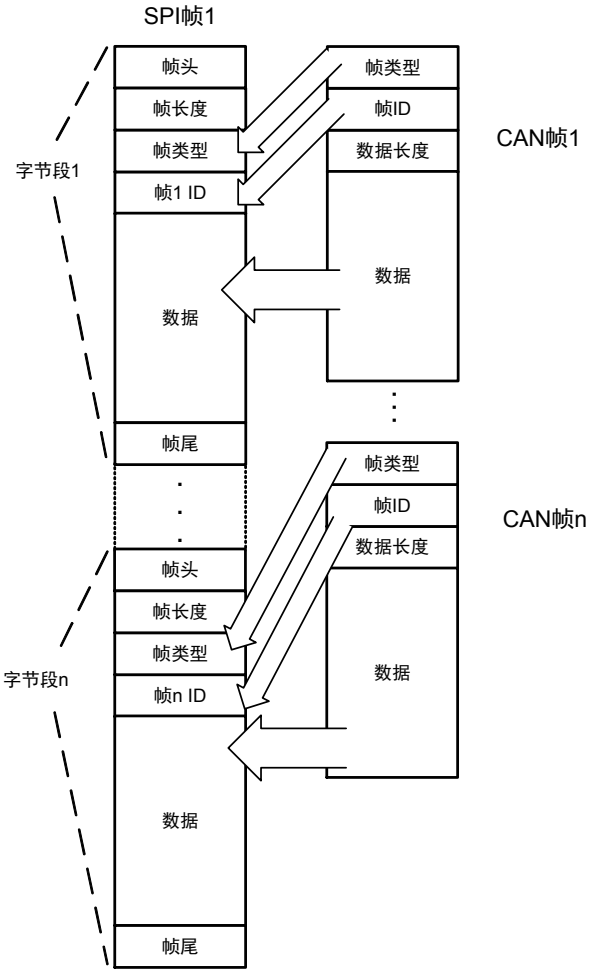


图 3.33 CAN 帧转 SPI 帧（自定义协议转换）

转换实例

例 1：假设用户配置的串行帧头为 0x40，帧尾为 0x1A。转换示例如图 3.34。

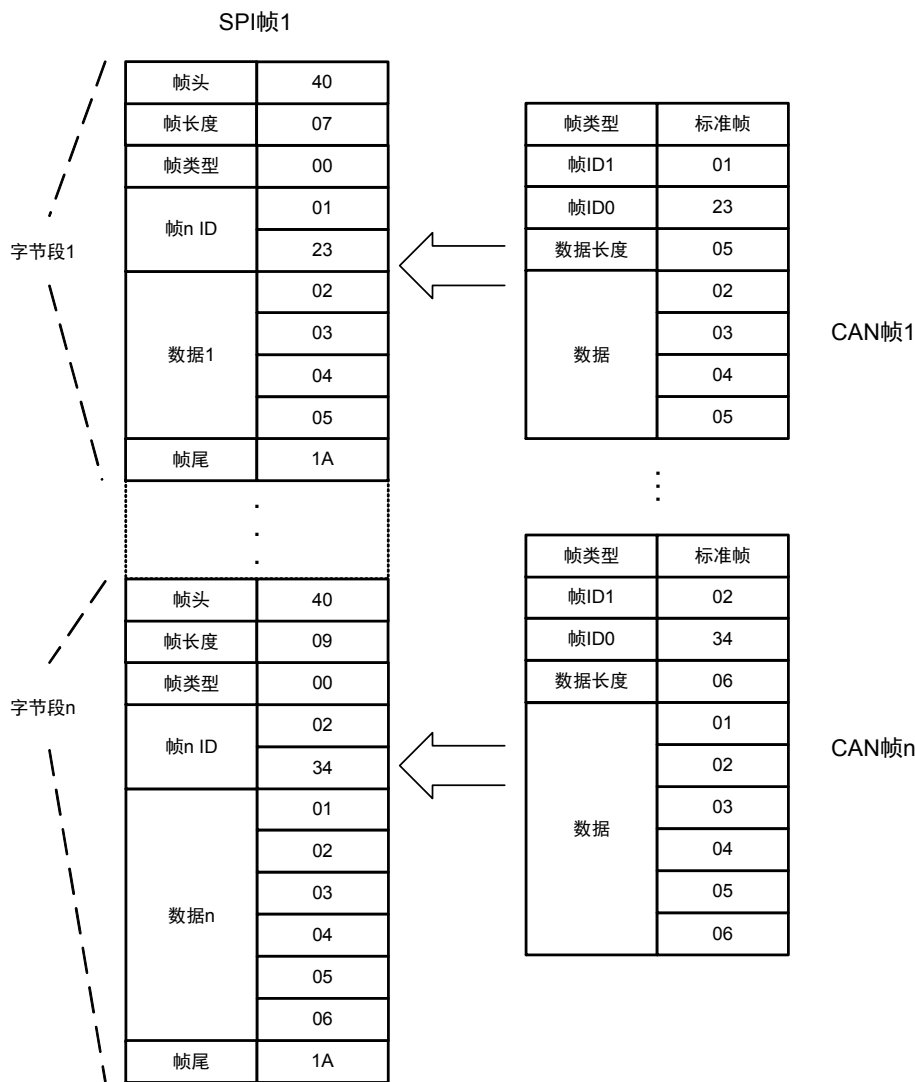


图 3.34 CAN 帧转 SPI 帧示例（自定义协议转换）

3.3.4 自定义带校验协议转换

自定义带校验协议转换方式下，串行帧必须符合规定的帧格式。有效的串行帧由帧头、帧长度、帧类型、帧 ID、数据域、CRC 校验、帧尾。

帧头，是一个串行帧的开始字节，范围为 0x00~0xFF，由用户配置。

帧长度，指帧类型、帧 ID、数据域总共包含的字节数。

帧类型，指发送 CAN 帧的帧类型。0x00 代表标准帧，0x08 代表扩展帧。

帧 ID，指发送 CAN 帧的帧 ID。帧类型为 0x00 时，帧 ID 由 2 个字节组成，高位在前；帧类型为 0x08 时，帧 ID 由 4 个字节组成，高位在前。

数据域，是需要发送的数据，转换至 CAN 帧的数据域。

CRC 校验，指对帧类型、帧 ID、数据域进行 CRC 校验的 1 个字节。CRC 采用的算法参考表 3.4。在该 CRC 算法下将数据 {0x00, 0x01, 0x23, 0x01, 0x02, 0x03, 0x04, 0x05,



0x06, 0x07, 0x08, 0x09, 0x0A, 0x0B}进行校验结果为 0x6A。

表 3.4 CRC 校验说明

参数模型	宽度位数	多项式	初始值	结果异或值	输入输出反转性
X8+X5+X4+1	8	0x31	0xFF	00	均无反转

帧尾，是一个串行帧的结束字节，范围为 0x00~0xFF，由用户配置。

当用户发送的串行帧完全符合定义的格式时，CSM330A 才会接收串行帧的数据并进行转发，否则不作处理直接丢弃。

1. 串行帧转 CAN 帧（SPI/UART→CAN）

串行接口接收到有效的串行帧后，“帧类型”确定将要发送 CAN 帧的帧类型，“帧 ID”作为 CAN 帧 ID，“数据域”填充至 CAN 帧数据域。如图 3.35、图 3.36 为串行帧转 CAN 帧示意图。其中图 3.35 串行帧数据域字节数小于等于 8，数据全部通过一个 CAN 帧转发；图 3.36 串行帧数据域字节数大于 8，数据通过多个 CAN 帧转发。

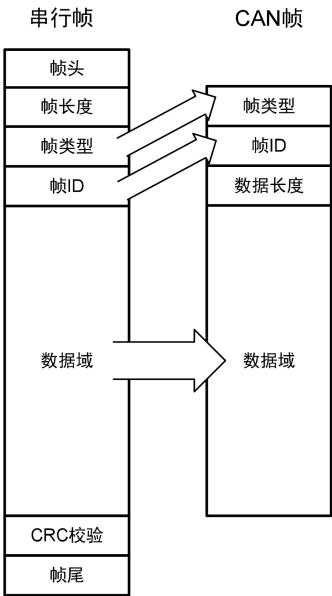


图 3.35 串行帧转 CAN 帧 1（自定义带校验协议转换）

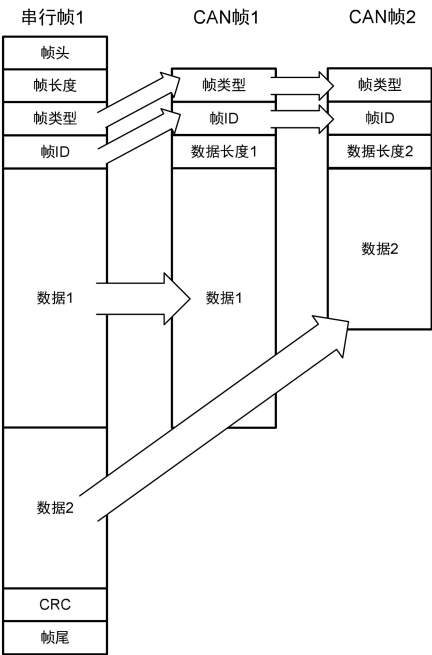


图 3.36 串行帧转 CAN 帧 2（自定义带校验协议转换）

转换实例

例 1: 假设用户配置的串行帧头为 0x40, 帧尾为 0x1A。用户发送帧类型为标准帧(0x00), 帧 ID 为 0123, 数据为 0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07, 0x08, 0x09, 0x0A, 0x0B, 则帧长度为 0x0E。转换示例如图 3.37。

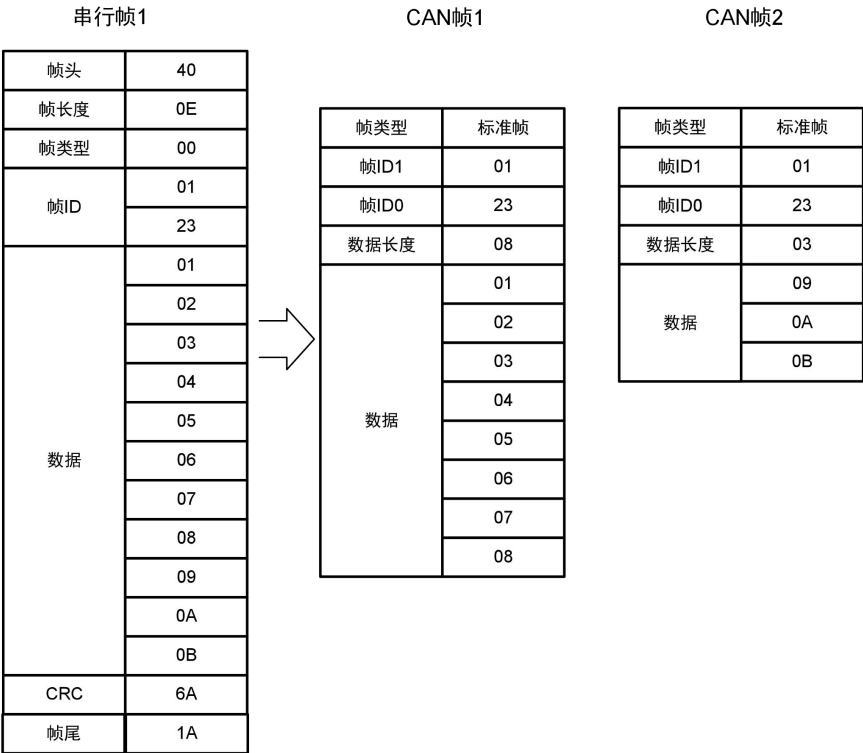


图 3.37 串行帧转 CAN 帧示例（自定义带校验协议转换）

2. CAN 帧转 UART 帧

CAN 转 UART 模式下，CAN 总线收到一帧 CAN 帧即立即转发一帧。数据格式对应如图 3.38 所示，转换时，CAN 帧类型转换到 UART 帧帧类型字节，帧 ID 转换到 UART 帧帧 ID 字节，CAN 帧数据域中的数据依序全部转换到串行帧数据域中，CRC 校验域将自动计算后填充。

为了保证串行帧的完整性，在 CAN 帧转换为串行帧时，帧头、帧尾与用户设置一致，这时的串行帧完全与串行帧转 CAN 帧时的帧格式一致。

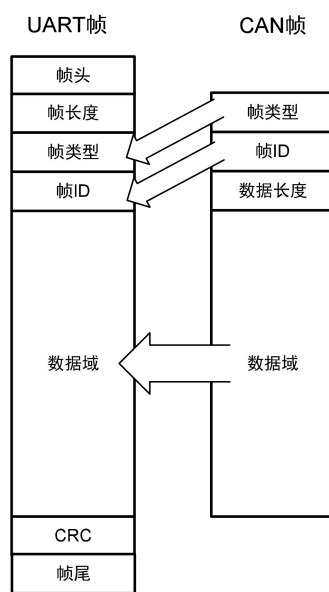


图 3.38 CAN 帧转 UART 帧（自定义带校验协议转换）

转换实例

例 1：假设用户配置的串行帧头为 0x40，帧尾为 0x1A。转换示例如图 3.39。

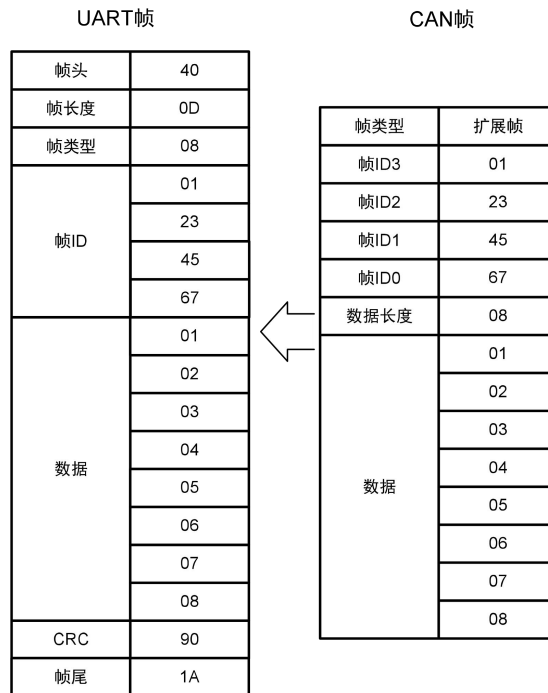


图 3.39 CAN 帧转 UART 帧示例（自定义带校验协议转换）

3. CAN 帧转 SPI 帧（CAN→SPI）

CSM330A 作为 SPI 从机，无法主动控制 SPI 外设，当产品从 CAN 总线收到一帧数据后，只能立即存储到 CAN 接收缓冲区。当 CAN 缓冲区的 CAN 帧数达到反馈触发帧数，或触发时间，INT 引脚输出低电平通知 SPI 主机进行数据读取。

当 SPI 主机从 CSM330A 获取 CAN 缓冲区的大小后，SPI 可通过一帧读取出 CAN 缓冲区中包含的所有 CAN 帧数据。需要注意的是 CSM330A 可读缓冲区的大小我们用 12bit 来表示，因此能够表示的最大值仅为 0xFFF，若实际可读字节数超过表示的最大范围，用户获取到的可读字节数仍为 0xFFF，此种情况下用户可先读取 0xFFF 个数据，再依次读出缓冲区剩余数据。

自定义带校验转换方式下，CAN 接收缓冲区存储的是多个有效格式字节段的集合。CAN 端接收到一个 CAN 数据帧时，将其转换为符合用户定义格式的字节段放置到缓冲区。为避免缓存区数据量超过 0xFFF 后可能会发生读取最后一帧数据不完整的情况，建议用户可以先读取 0xFFF 个数据，再读取剩下的数据，读完后再将数据一起做处理；或者用户将第一次读取的数据先做处理，将最后一个不完整的帧做保留，将后续 CAN 数据读取完合并后再一起处理。

SPI 主机获取 CAN 接收缓冲区大小后，可通过一个 SPI 帧一次读取出存储在缓冲区的 n 个字节段。其转换数据格式如图 3.40。

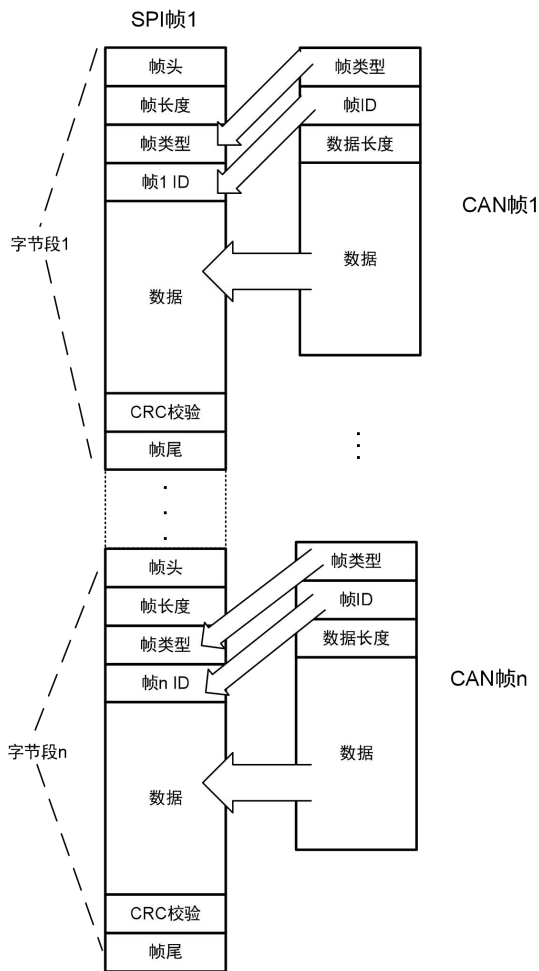


图 3.40 CAN 帧转 SPI 帧（自定义带校验协议转换）

转换实例

例 1：假设用户配置的串行帧头为 0x40，帧尾为 0x1A。转换示例如图 3.41。

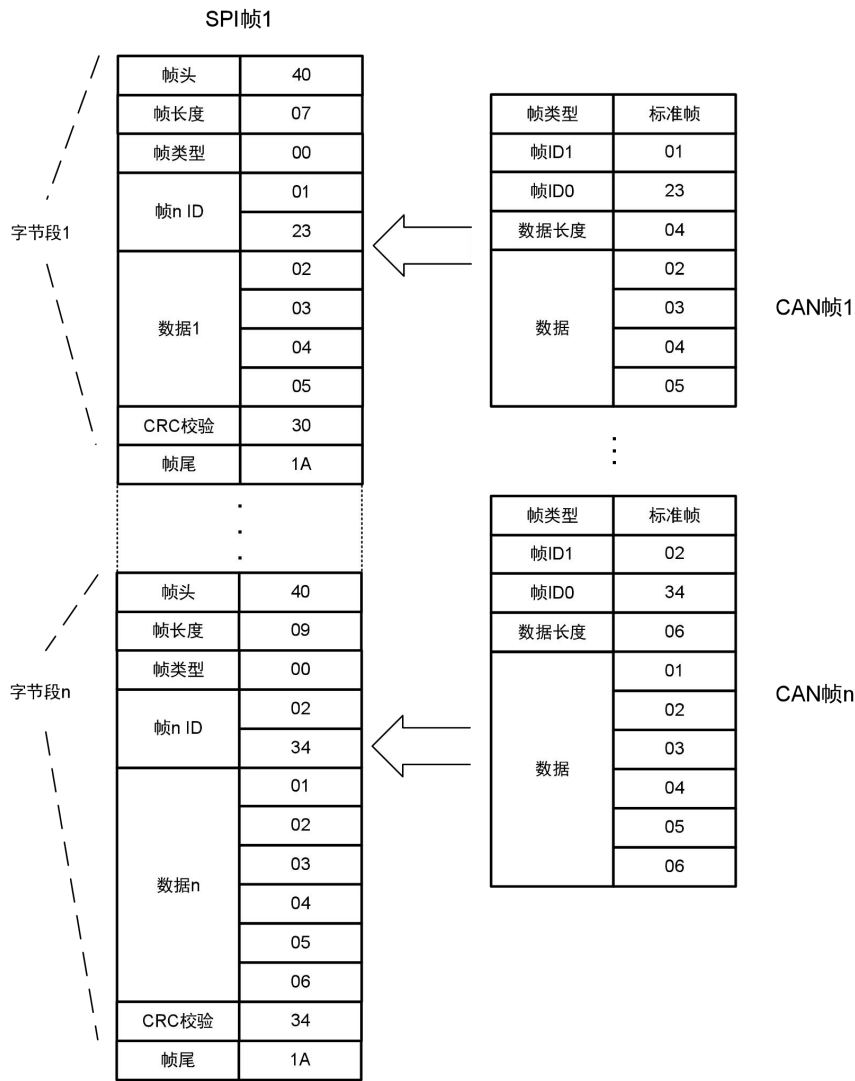


图 3.41 CAN 帧转 SPI 帧示例（自定义带校验协议转换）

3.4 错误反馈机制

CSM330A 具备错误反馈机制，用于帮助用户了解通信过程中的错误信息及读取 CAN 错误计数值。该功能默认是禁能状态，在产品配置时使能该功能才可使用。

错误检测范围为串行端的帧尾错误、串行端的 CRC 校验错误、CAN 计数错误。当 CSM330A 接收的串行数据发生前 2 项错误或 CAN 错误计数值>128, CSM330A 的 Error Txd 引脚由高电平转为低电平通知用户。除此之外，用户也可无需检测该引脚，主动发送读错误命令也可读取到错误寄存器中的信息。

在透明转换、透明带标识转换模式下 UART/SPI 端没有帧尾及 CRC 校验字节，因此在这两种模式下，错误检测范围仅为 CAN 计数错误；同理，在自定义模式下，错误检测范围为帧尾、CAN 计数错误；在自定义带校验模式下，检测范围为帧尾、CRC 校验、CAN 计数错误。

3.4.1 获取错误信息

1. UART 转 CAN 模式

在该模式下，我们复用 MOSI 和 MISO 分别为错误命令接收（Error Rxd）和错误信息发送（Error Txd）引脚，硬件连接图参考图 2.6。错误反馈机制串口速率与通信串口速率相同，错误反馈机制时序可参考图 3.42 所示。

（1）当 CSM330A 检测的错误信息满足触发条件时，延时约 15us 后 Error Txd 由高电平转为低电平并保持；

（2）主机检测到 Error Txd 为低电平时，可延时约 10us 后，向 Error Rxd 引脚发送读错误信息命令(F7, F8, 04, 00, 0B)；

（3）CSM330A 检测到读错误命令等待约 20us 后，Error Txd 转为串口模式的 TXD 功能，将错误信息共 9 个字节发出，详见 3.4.2 节；该串口格式与串口转 CAN 模式下的 UART 格式一致，参考表 4.2 中的 UART 参数；

（4）值得注意的是若 Error Txd 引脚转为低电平的 1s 内仍未收到读错误命令帧，该引脚将自动转为高电平，等待下一次的错误触发，该动作不会清除错误寄存器中的值。若 Error Txd 在低电平状态下 CSM330A 又产生了新的错误触发，低电平保持时间依次后移，直至 1s 内没有新的触发，该引脚才转为高电平。

2. SPI 转 CAN 模式

在该模式下，我们复用串口的 TXD 和 RXD 分别为错误信息发送（Error Txd）和错误命令接收（Error Rxd）引脚，硬件连接图参考图 2.5，错误反馈机制速率默认为 115200，可通过写配置命令改变错误反馈机制串口速率。

（1）当 CSM330A 检测的错误信息满足触发条件时，延时约 15us 后 Error Txd 由高电平转为低电平并保持；

（2）主机检测到 Error Txd 为低电平时，可延时约 10us 后，向 Error Rxd 引脚发送读错误信息命令(F7, F8, 04, 00, 0B)；

（3）CSM330A 检测到读错误命令等待约 20us 后，Error Txd 转为串口模式的 TXD 功能，将错误信息共 9 个字节发出，详见 3.4.2 节；该串口格式与串口转 CAN 模式下的 UART 格式一致，参考表 4.2 中的 UART 参数；

（4）值得注意的是若 Error Txd 引脚转为低电平的 1s 内仍未收到读错误命令帧，该引脚将自动转为高电平，等待下一次的错误触发，该动作不会清除错误寄存器中的值。若 Error Txd 在低电平状态下 CSM330A 又产生了新的错误触发，低电平保持时间依次后移，直至 1s 内没有新的触发，该引脚才转为高电平。

3. 读错误命令帧格式

CSM330A 读错误命令帧格式如表 3.5。

表 3.5 读错误命令帧格式

帧起始	命令字	保留字	校验字
2 字节	1 字节	1 字节	1 字节
0xF7, 0xF8	0x04	0x00	0x0B

说明：

- 帧起始：2 字节，依次为 0xF7，0xF8。
- 命令字：读错误命令字 0x04。
- 保留字：1 字节为 0x00。
- 校验字：1 字节，为前面所有字节的异或。

3.4.2 错误信息回应帧

CSM330A 收到主机发送的读错误命令(F7, F8, 04, 00, 0B)后，将通过 Error Txd 发出共 9 个字节的通信错误信息，错误信息格式见表 3.6。

表 3.6 错误信息回应帧格式

帧起始	保留字节	CRC/帧尾 错误计数	CAN 总线错误计数				校验字
			接收错误 计数	发送错误 计数	被动错误 计数	BUSOFF 计数	
2 字节	1 字节	1 字节	1 字节	1 字节	1 字节	1 字节	1 字节
0xF7,0xF8	0x00	0x00	0x00	0x00	0x00	0x00	0x0C

说明：

- 帧起始：2 字节，依次为 0xF7，0xF8。
- 保留字：1 字节，固定为 0x00。
- CRC/帧尾错误数：1 字节，此字节存放串行端错误信息计数值，最大累加至 FF，用户读取后清 0。
- 接收错误计数：1 字节，此字节存放 CAN 接收错误计数值，最大累加至 FF，用户读取后清 0。
- 发送错误计数：1 字节，此字节存放 CAN 发送错误计数值，最大累加至 FF，用户读取后清 0。
- 被动错误计数：当发送错误计数值或接收错误计数值大于 128 时，被动错误计数将加 1，最大累加至 FF，用户读取后清 0。
- BUSOFF 计数：当发送错误计数值或接收错误计数值大于 255 时，BUSOFF 错误计数加 1，最大累加至 FF，用户读取后清 0。
- 校验字节：前面所有字节的异或。

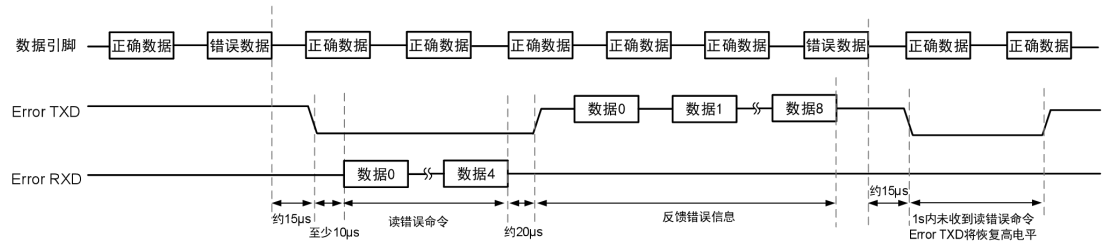


图 3.42 错误反馈机制时序图

4. 产品配置

4.1 配置参数

4.1.1 转换参数

1. 转换方式

CSM330A 的数据转换方式，指串行总线和 CAN 总线数据之间转换的基本规则。同一时间内，产品只能工作于一种数据转换方式，若需要改变数据转换方式，则需要更改配置。产品配置好一种转换方式时，将同时作用于 SPI 转 CAN 模式，及 UART 转 CAN 模式。

数据转换方式有四种：透明转换、透明带标识转换、自定义协议转换、自定义带校验协议转换。

透明转换，是指任何一侧总线只要接收到数据，即立刻发送至另一侧总线上，数据不做任何处理。此方式的详细说明见 3.3.1 小节。

透明带标识转换，是在透明转换的基础上衍生而来，是指发送或接收的串行帧中都包含了有效的 CAN 帧 ID 字节。此方式的详细说明见 3.3.2 小节。

自定义协议转换，串行帧必须符合规定的帧格式。有效的串行帧由帧头、帧长度、帧类型、帧 ID、数据域、帧尾组成。此方式的详细说明见 3.3.3 小节。

自定义带校验协议转换，串行帧必须符合规定的帧格式。有效的串行帧由帧头、帧长度、帧类型、帧 ID、数据域、CRC 校验域、帧尾组成。此方式的详细说明见 3.3.4 小节。

2. 转换方向

转换方向，指数据的允许转换方向。CSM330A 有三种转换方向：双向、仅 SPI/UART 转 CAN、仅 CAN 转 SPI/UART。

3. 允许 CAN 帧信息转换到串行帧中

允许 CAN 帧 ID 转换到串行帧中此配置参数仅在透明转换方式下有效，若使能，CAN 转 SPI/UART 时，CAN 帧的帧信息同时转换至 SPI/UART，详见 3.3.1 小节。

4. 允许 CAN 帧 ID 转换到串行帧中

此配置参数仅在透明转换方式下有效，若使能，默认同时使能帧信息转换，CAN 转 SPI/UART 时，CAN 帧的帧 ID、帧信息同时转换至 SPI/UART，详见 3.3.1 小节。

5. CAN 标识符在串行帧中的位置

此配置参数仅在透明带标识转换方式下有效，包括 CAN 标识符的起始地址和长度设置，详见 3.3.2 小节。

6. 帧头、帧尾

此配置参数仅在自定义协议、自定义带校验协议转换方式下有效，用于设置串行帧的帧开始、帧结束。

7. CRC 校验

此配置参数仅在自定义带校验模式下有效，该校验字节校验范围为帧类型+帧 ID+帧数据。

8. 错误反馈机制

此功能默认是禁能状态，需用户配置使能才可使用，该功能描述详见 3.4 节。

4.1.2 SPI 参数

1. 反馈触发帧数

此配置参数仅在 SPI 转 CAN 模式下有效。由于 CSM330A 作为 SPI 从机，无法主动向主机发送数据，故当 CSM330A 接收到一定数量的 CAN 帧数据后，需要通过 INT 引脚通知主机获取数据。反馈触发帧数以接收到的 CAN 帧为单位，当 CAN 缓冲区接收到设定数量的 CAN 帧后，触发反馈。

2. 反馈触发时间

此配置参数仅在 SPI 转 CAN 模式下有效。由于 CSM330A 作为 SPI 从机，无法主动向主机发送数据。当 CAN 缓冲区接收到的 CAN 帧数未达到反馈触发帧数，并在反馈触发时间内未被读取时，通过 INT 引脚通知主机获取数据。

反馈触发时间以 100ms 为单位，当 CAN 反馈触发时间到达设定值后，触发反馈。

4.1.3 UART 参数

1. 波特率

指通信串口和错误反馈机制串口的工作波特率。有效串口波特率如表 4.6。

2. 帧间隔

指 UART 通信帧之间的时间间隔，详见 3.2.3 小节 UART 帧的定义。

4.1.4 CAN 参数

1. 波特率

指 CAN 的工作波特率。有效 CAN 波特率如表 4.6。

2. 发送帧类型

此配置参数仅在透明转换、透明带标识转换方式下有效。指发送的 CAN 帧类型，包括两种：标准帧、扩展帧。

3. 发送标识符

此配置参数仅在透明转换方式下有效。透明转换方式下，CAN 发送帧的 ID 以设定 ID 为准。

帧 ID 从左到右依次为 ID3、ID2、ID1、ID0。帧 ID3 为最高字节。如果为标准帧，其范围为 0x000~0x7FF，扩展帧范围为 0x00000000~1FFFFFFF。如在透明转换时发送帧 ID 为 0x0123 的标准帧，其发送标识符应设为 00 00 01 23。

4. 滤波器使能

只有选择了该项，接收过滤模式及相应的屏蔽、验收码才会开放。如果不想使用滤波功能，则不选择该项以接收所有 CAN 帧。

5. 接收过滤模式

该项选择分为扩展帧滤波和标准帧滤波，如果仅想接收扩展帧格式的 CAN 帧，则应该选择扩展帧滤波；如果仅想接收标准帧格式的 CAN 帧，则应该选择标准帧滤波。

6. 屏蔽码

屏蔽码用来管理“验收码”，按照位管理。当屏蔽码某位值为 1 时，则该位对应的验收码会被“使能”，被“使能”的“验收码”和产品要接收的 CAN 帧的“帧 ID”相同，该帧 CAN 帧才会被接收到接收缓冲区。当“屏蔽码”的位值为 0 时，验收码不起作用，相应位的帧标识为任何值都可被接收。

填充数据格式为 16 进制，每个 8 位的字节间用“空格符”隔开。

7. 验收码

验收码有验收码 0~验收码 5，共 6 组。

接受 CAN “帧 ID”时的比较值，和“屏蔽码”按照位的关系相对应。在“屏蔽码”设定为 1 时，只有接收“帧 ID”和“验收码”相同时才会将该帧数据收到接收缓冲区中，否则不接收。

填充数据格式为 16 进制，每个 8 位的字节间用“空格符”隔开。表 4.1 给出了屏蔽位、验收位过滤帧 ID 的真值关系。

表 4.1 滤波、屏蔽码真值表

屏蔽位	验收位	帧 ID	接收或拒绝位
0	X	X	接收
1	0	0	接收
1	0	1	拒绝
1	1	0	拒绝
1	1	1	接收

4.2 出厂默认配置

表 4.2 出厂默认配置参数

参数	默认值	说明	备注
UART 波特率	115200 bps	通信串口波特率 错误反馈机制串口波特率	--
UART 数据位	8	串口数据位，固定为 8	无法更改
UART 停止位	1	串口停止位，固定为 1	
UART 校验位	0	串口校验位，固定为 0	
CAN 波特率	125 kbps	CAN 工作波特率	--
错误反馈	禁能	禁能错误反馈机制	--
滤波器使能	禁能	CAN 验收滤波器功能选择	--
接收过滤模式	扩展帧滤波	CAN 验收滤波方式选择	--
屏蔽码	FF FF FF FF	CAN 验收滤波器屏蔽码	--
验收码 0	00 00 00 00	CAN 验收滤波器仲裁码 0	--
验收码 1	00 00 00 00	CAN 验收滤波器仲裁码 1	--
验收码 2	00 00 00 00	CAN 验收滤波器仲裁码 2	--
验收码 3	00 00 00 00	CAN 验收滤波器仲裁码 3	--

续上表

参数	默认值	说明	备注
验收码 4	00 00 00 00	CAN 验收滤波器仲裁码 4	--
验收码 5	00 00 00 00	CAN 验收滤波器仲裁码 5	--
转换方式	透明带标识转换	数据转换方式	--
转换方向	双向	数据转换方向	--
UART 帧间隔	2 个字符	UART 帧时间间隔字符数	--
CAN 帧信息转发到串行帧中	禁能	向串行帧转换时的 CAN 帧信息的转发使能	只适用于透明转换
CAN 帧 ID 的转发到串行帧中	禁能	向串行帧转换时帧 ID 的转发使能	只适用于透明转换
发送帧类型	扩展帧	向 CAN 转换时配置的 CAN 帧类型	适用于透明转换、透明带标识转换
发送标识符	00 00 00 00	SPI/UART 转 CAN 的 ID3~ID0	只适用于透明转换
串行帧中 CAN 标识的长度	4 字节	串行帧中 CAN 标识的长度	只适用于透明带标识转换
串行帧中 CAN 标识起始地址	0 字节	串行帧中 CAN 标识起始地址	
帧头	0x40	串行帧帧头	只适用于自定义协议转换、自定义带校验协议转换
帧尾	0x1A	串行帧帧尾	
反馈触发帧数	15 帧	CAN 缓冲区反馈触发帧数	只适用于 SPI 转 CAN 模式
反馈触发时间	500ms	CAN 缓冲区反馈触发时间	
数据位长度	8 位	SPI 数据长度固定为 8 位	无法更改
位传输方式	MSB	位传输按高位优先	
CPOL	1	SPI 工作模式参数 CPOL (固定为 1)	
CPHA	1	SPI 工作模式参数 CPHA (固定为 1)	

4.3 配置通信协议

对 CSM330A 进行配置，首先需要将产品复位并进入“SPI 配置模式”或“UART 配置模式”。用户向 CSM330A 的 SPI 接口或 UART 接口发送相应命令帧，产品接收到配置命令后，对自身进行操作，并返回回应帧。

需要注意的是，在“UART 配置模式”下，产品接收到数据，处理完后会主动向 UART 返回回应帧。但产品处于“SPI 配置模式”时，CSM330A 作为从机无法主动返回回应帧，因此，SPI 主机发送命令帧，CSM330A 在正确接收并处理完成后，会将 INT 引脚置低，通知主机处理完成，可以读取回应帧。

若用户不想对 INT 引脚进行检测，则应在主机发送命令帧后，等待一定时间，待 CSM330A 处理完成后，SPI 主机才能读取回应帧数据或更改产品的工作状态。命令帧处理等待时间如表 4.3。

表 4.3 命令帧处理等待时间

命令帧类型	等待处理时间
写配置	200ms
读配置	3ms
验证产品标识	3ms

1. 命令帧

命令帧由主控制端（上位机、MCU 等）向被控制端 CSM330A 发送，CSM330A 接收到命令帧后进行相应操作。

2. 回应帧

回应帧指 CSM330A 接收到命令帧后，由 CSM330A 返回的，对主控制端的回应信息。

4.3.1 写配置参数

1. 写配置命令帧

写配置命令帧，用于对 CSM330A 的参数进行配置，其包含配置 CSM330A 所需要的所有数据。写配置命令帧的格式如表 4.4。

表 4.4 写配置命令帧格式

帧起始	命令字	数据长度	数据域	校验字
2 字节	1 字节	1 字节	60 字节	1 字节
0xF7, 0xF8	0x01	0x3C	定义如表 4.5	前面所有字节的异或

说明：

帧起始：2 字节，依次为 0xF7，0xF8。

命令字：1 字节。固定为 0x01。

数据长度：1 字节，帧内数据域中数据字节数，固定为 60（即 0x3C）。

数据域：60 字节，配置信息。

校验字：1 字节，为前面所有字节的异或。

表 4.5 写配置命令帧数据域定义

字节位置	定义	数据范围	说明
0	通信 UART 波特率， 错误反馈机制 UART 波特率	0x00~0x13	对应关系如表 4.6。
1	UART 数据位	0x08	固定为 8。
2	UART 停止位	0x01	固定为 1。
3	UART 校验位	0x00	固定为 0。
4	CAN 波特率	0x00~0x0F	对应关系如表 4.6。
5	错误反馈机制	0x00 / 0x01	0x00：禁能； 0x01：使能。

续上表

字节位置	定义	数据范围	说明
6	(保留字节)	--	未使用, 无效。
7	(保留字节)	--	未使用, 无效。
8	滤波器使能	0x00 / 0x01	0x00: 禁能; 0x01: 使能。
9	接收过滤模式	0x08 / 0x00	0x08: 仅扩展帧滤波; 0x00: 仅标准帧滤波。
10~13	屏蔽码	00 00 00 00 ~ FF FF FF FF	--
14~17	验收码 0	00 00 00 00 ~ FF FF FF FF	--
18~21	验收码 1	00 00 00 00 ~ FF FF FF FF	--
22~25	(保留字节)	--	未使用, 无效。
26~29	验收码 2	00 00 00 00 ~ FF FF FF FF	--
30~33	验收码 3	00 00 00 00 ~ FF FF FF FF	--
34~37	验收码 4	00 00 00 00 ~ FF FF FF FF	--
38~41	验收码 5	00 00 00 00 ~ FF FF FF FF	--
42	转换模式	0x01~0x04	0x01: 透明转换; 0x02: 透明带标识转换; 0x03: 自定义协议转换; 0x04: 自定义带校验协议转换;
43	转换方向	0x00~0x02	0x00: 双向; 0x01: 仅 SPI/UART 转 CAN; 0x02: 仅 CAN 转 SPI/UART。
44	UART 帧间隔	0x02~0x0F	--
45	CAN 帧信息转发到串行帧中	0x00 / 0x01	0x00: 禁能; 0x01: 使能。 该项只用于透明转换。

续上表

字节位置	定义	数据范围	说明
46	CAN 帧 ID 转发到串行帧中	0x00 / 0x01	0x00: 禁能; 0x01: 使能。 该项只用于透明转换。
47	发送帧类型	0x08 / 0x00	0x08: 扩展帧; 0x00: 标准帧。 该项用于透明转换、透明带标识转换。
48~51	发送标识符	00 00 00 00 ~ 1F FF FF FF	该项只用于透明转换, 标识 ID3~ID0。 标准帧: 000~7FF 有效; 扩展帧: 00000000~1FFFFFFF 有效。
52	串行帧中 CAN 标识的长度	0x01~0x04	以字节为单位, 该项只用于透明带标识转换。
53	串行帧中 CAN 标识起始地址	0x00~0x07	以字节为单位, 该项只用于透明带标识转换。
54	帧头	0x00~0xFF	一个字节, 用户自定义。
55	帧尾	0x00~0xFF	一个字节, 用户自定义。
56	反馈触发帧数	0x08~0xFF	以一个 CAN 帧为一个单位, 该项只用于 SPI 转 CAN 模式下。
57	反馈触发时间	0x01~0xFF	以 100ms 为一个时间单位, 该项只用于 SPI 转 CAN 模式下。
58	CPOL	0x00~0x01	暂仅支持 0x01。
59	CPHA	0x00~0x01	暂仅支持 0x01。

表 4.6 串口、CAN 波特率代码表

代码	对应 16 进制	串口波特率 (bps)	CAN 波特率 (bps)
0	0x00	--	--
1	0x01	115200	5k
2	0x02	57600	10k
3	0x03	38400	20k
4	0x04	19200	40k
5	0x05	14400	50k
6	0x06	9600	80k
7	0x07	4800	100k
8	0x08	2400	125k
9	0x09	1200	200k
10	0x0A	600	250k
11	0x0B	300	400k
12	0x0C	128000	500k
13	0x0D	230400	666k
14	0x0E	256000	800k
15	0x0F	460800	1M
16	0x10	921600	25k
17	0x11	1M	--
18	0x12	1.5M	--
19	0x13	2M	--

2. 写配置回应帧

CSM330A 在收到写配置参数命令帧后，会依据收到的数据对当前配置进行更新，操作完成后，无论配置更新成功与否，都会对主控制端发送回应帧进行应答。写配置回应帧的格式如表 4.7。

表 4.7 写配置参数命令帧格式

帧起始	命令字	状态字	校验字
2 字节	1 字节	1 字节	1 字节
0xF7, 0xF8	0x01	0x13 或 0x07	前面所有字节的异或

说明：

帧起始：2 字节，依次为 0xF7，0xF8。

命令字：回应帧中的命令字和命令帧中的相同，即 0x01。

状态字：1 字节，0x13 表示命令执行成功，0x07 表示命令执行失败。

校验字：1 字节，为前面所有字节的异或。

4.3.2 验证产品硬件标识

1. 验证产品硬件标识命令帧

验证产品硬件标识，用于在配置产品前，确认所配置产品的具体硬件信息是否正确。验证产品硬件标识命令帧的格式如表 4.8。

表 4.8 验证产品硬件标识命令帧格式

帧起始	命令字	数据长度	数据域	校验字
2 字节	1 字节	1 字节	4 字节	1 字节
0xF7, 0xF8	0x02	0x04	0x0A, 0x15, 0x12, 0x03	0x07

说明：

帧起始：2 字节，依次为 0xF7，0xF8。

命令字：1 字节。固定为 0x02。

数据长度：1 字节，帧内数据域中数据字节数，固定为 4（即 0x04）。

数据域：4 字节，产品硬件标识信息。

校验字：1 字节，为前面所有字节的异或。

2. 验证产品硬件标识回应帧

CSM330A 在收到验证产品硬件标识命令帧后，会将命令帧中的硬件标识与自身硬件标识进行匹配，并将比对结果通过回应帧返回到主控制端。验证产品硬件标识回应帧的格式如表 4.9。

表 4.9 验证产品硬件标识回应帧格式

帧起始	命令字	状态字	校验字
2 字节	1 字节	1 字节	1 字节
0xF7, 0xF8	0x02	0x13 或 0x07	前面所有字节的异或

说明：

帧起始：2 字节，依次为 0xF7，0xF8。

命令字：回应帧中的命令字和命令帧中的相同，即 0x02。

状态字：1 字节，0x13 表示标识匹配成功，0x07 表示标识匹配失败。

校验字：1 字节，为前面所有字节的异或。

4.3.3 读配置参数

1. 读配置参数命令帧

读配置参数命令帧，可用于获取 CSM330A 当前配置参数。读配置参数命令帧的格式如表 4.10。

表 4.10 读配置参数命令帧格式

帧起始	命令字	数据长度	数据域	校验字
2 字节	1 字节	1 字节	0 字节	1 字节
0xF7, 0xF8	0x03	0x00	无	0x0C

说明：

帧起始：2 字节，依次为 0xF7，0xF8。

命令字：1 字节。固定为 0x03。

数据长度：1 字节，此命令帧无数据，固定为 0x00。

数据域：0 字节，无数据。

校验字：1 字节，为前面所有字节的异或，即 0x0C。

2. 读配置参数回应帧

CSM330A 在收到读配置参数命令帧后，会获取自身当前的配置参数信息，并通过回应帧返回到主控制端。读配置参数回应帧的格式如表 4.11。

表 4.11 读配置参数回应帧格式

帧起始	命令字	数据长度	数据域	校验字
2 字节	1 字节	1 字节	60 字节	1 字节
0xF7, 0xF8	0x03	0x3C	定义如表 4.5	前面所有字节的异或

说明：

帧起始：2 字节，依次为 0xF7，0xF8。

命令字：回应帧中的命令字和命令帧中的相同，即 0x03。

数据长度：1 字节，帧内数据域中数据字节数，固定为 60（即 0x3C）。

数据域：60 字节，当时配置参数信息。

校验字：1 字节，为前面所有字节的异或。

4.4 配置方式

4.4.1 MCU 配置方式

1. 通过 SPI 配置

当用户使用 MCU 的 SPI 接口与 CSM330A 进行连接时，可以通过 SPI 接口对产品参数进行配置，硬件连接图可参考图 2.3。

配置方法如下：

- (1) 上电后各引脚电平可以保持稳定，此时 CFG 置低至少 50us，控制 RST 复位至少 100us，等待至少 200ms，产品进入 SPI 配置模式；
- (2) CTL0、CTL1 引脚置高，控制 CSM330A 进入 SPI 主机写状态，等待至少 50us；
- (3) SSEL 置低，MCU 发送写配置命令帧，然后 SSEL 置高，等待至少 200ms，或检测到 INT 引脚至低；
- (4) CTL0 置高，CTL1 置低，控制 CSM330A 进入 SPI 主机读状态，等待至少 50us；
- (5) SSEL 置低，主机发送 5 个字节无效数据，接收产品写配置回应帧，然后 SSEL 置高，延时 50us；
- (6) MCU 确认写配置回应帧内容，判断配置是否成功；
- (7) 配置完成后，CTL0 置低等待至少 50us，CFG 引脚置高等待至少 50us，RST 复位，等待至少 200ms，产品进入 SPI 转 CAN 模式。

通过 SPI 写配置命令的时序图如图 4.1，此时序同样适用于读配置命令、验证产品标识等操作，只需要更改 SPI 读写数据，以及写配置命令后等待时间。

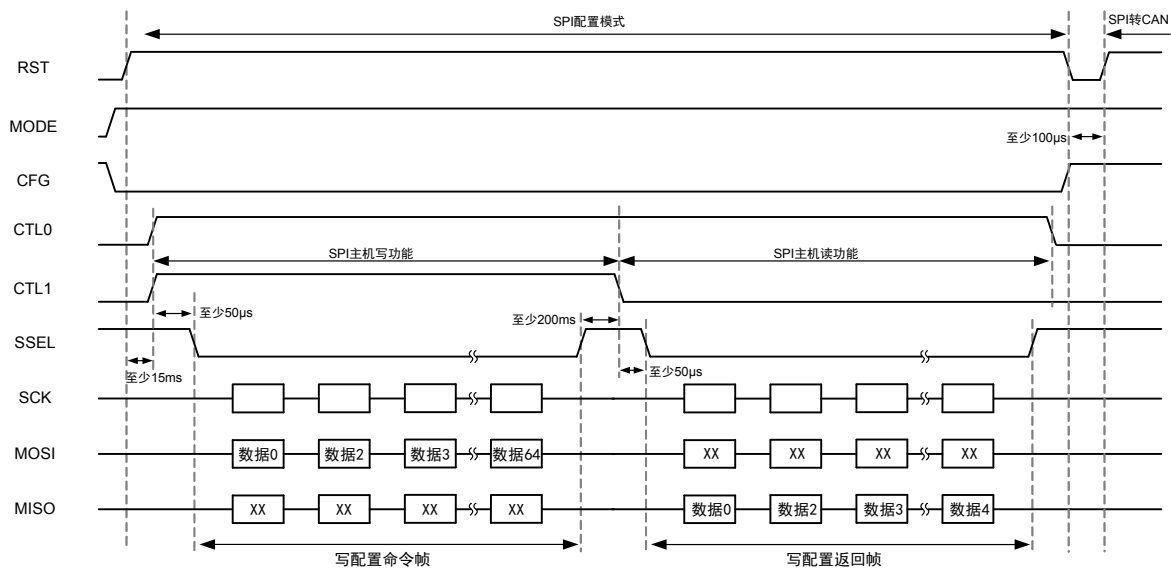


图 4.1 通过 SPI 写配置命令的时序图

2. 通过 UART 配置

当用户使用 MCU 的 UART 接口与 CSM330A 进行连接时,可以通过 UART 接口对产品参数进行配置,硬件连接图可参考图 2.4。其中,配置时的串口波特率固定为 9600bps,无法更改。若使用波特率不正确,则无法配置成功。

配置方法如下:

- (1) 上电后各引脚电平可以保持稳定,此时 CFG 置低,等待至少 50us 后, RST 低电平保持最少 100us 后,等待至少 200ms,产品进入 UART 配置模式;
- (2) MCU 发送写配置命令帧(产品 RXD 接收到数据);
- (3) 产品收到配置命令帧后,主动发送写配置回应帧至 MCU(产品 TXD 发送数据);
- (4) MCU 确认写配置回应帧内容,判断配置是否成功;
- (5) 配置完成至少等待 50us 后,CFG 引脚置高至少等待 50us, RST 复位,等待至少 200ms,产品进入 UART 转 CAN 模式。

通过 UART 写配置命令的时序图如图 4.2,此时序同样适用于读配置命令、验证产品标识等操作,只需要更改 MCU 发送的命令帧。

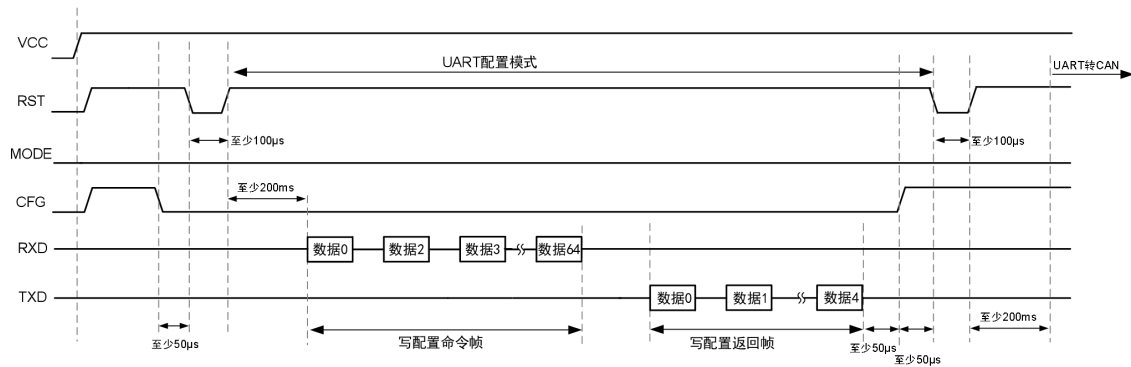


图 4.2 通过 UART 写配置命令时序图

4.4.2 上位机配置方式

除了用 MCU 对 CSM330A 进行配置的方式外，用户也可以先使用上位机配置方式对产品进行，配置完成后再使用。通过上位机进行配置，需要使用到 CSMCfgTools 配置软件、CSM330A-Eval、CSM330A-Eval-01 评估板，详细配置使用说明请参考 CSM330A-Eval 评估板说明文档《CSM330A 辅助开发工具手册》。

5. 产品使用注意事项

- ◆ 不支持热插拔；
- ◆ 未使用引脚请悬空处理；
- ◆ 产品为 ESD 敏感器件，请做好防静电措施；
- ◆ 产品供电电压切勿超过允许范围，以免损坏产品。CSM330A 标准 3.3V 供电，非 CAN 总线信号接口均为 3.3V 电平标准；
- ◆ 若需要更深入了解 CSM330A 产品的电气参数，请参考《CSM330A 数据手册》；
- ◆ 若需要更深入了解辅助开发工具，请参考《CSM330A 辅助开发工具手册》。
- ◆ CSM330A 具备 BOOT 检测，用于产品后续固件的维护。在上电或复位 70ms 内 CSM330A 的 CAN 端收到波特率为 1Mbps，ID 为 0x33A，数据为 0x75，0x70，0x67，0x72，0x61，0x64，0x65，当这 4 个条件同时满足后会进入到 BOOT 模式，一旦触发此模式芯片将无法再工作。用户使用过程中，应避免同时触发这 4 个条件。
- ◆ 生产注意事项：产品经来料检后，需放入干燥柜进行存储；产品上机贴片前应检查包装的完整性；产品在车间使用的时间长短需按 MSL3 等级管控，在车间使用寿命内，拆封后未使用完产品，须重新放入新的湿度指示卡和干燥剂进行真空包装，然后放入干燥柜存储；对于超出车间使用寿命未使用完产品，需烘烤后再使用；产品回流焊最高温度需 $\leq 245^{\circ}\text{C}$ 。以上生产注意事项详细内容与回流曲线请查阅《SiP 产品使用说明》。

6. 免责声明

隔离 SPI/UART 转 CAN 收发器芯片 CSM330A 版权属广州致远电子股份有限公司所有，其产权受国家法律绝对保护，未经本公司授权，其它公司、单位、代理商及个人不得非法使用和拷贝，否则将会承担相关的法律责任。

广州致远电子股份有限公司保留随时修改上述内容的权利，产品用户手册更新时恕不另行通知，如需查看最新版本的信息，请访问我司官方网站或联系我司人员获取。

诚信共赢，持续学习，客户为先，专业专注，只做第一

广州致远电子股份有限公司

更多详情请访问
www.zlg.cn

欢迎拨打全国服务热线
400-888-4005

