



CAN 总线转换模块

BM42D7611-1

版本: V1.00 日期: 2024-11-27

www.bestmodulescorp.com

目录

特性	3
概述	3
应用领域	3
方框图	4
引脚图	4
引脚说明	4
技术规格	5
极限参数	5
建议工作范围	5
直流电气特性	5
交流电气特性	5
CAN 特性	7
功能描述	8
系统描述	8
工作模式	8
数据转换方式	11
配置说明	21
写配置参数	23
配置步骤	27
应用电路	27
硬件电路	27
端口保护电路	28
推荐组网方式	29
Layout 说明	29
尺寸图	29
参考信息	30
修订历史	30
在线购买	30

特性

- 工作电压 (V_{CC}): 3.15V~3.6V
- 工作电流: 13mA @ 3.3V
- SPI 或 UART 与 CAN 接口的双向转换
- 一个 SPI 接口, 非自定义模式 SPI 速度高达 500kHz, 自定义模式为 300kHz (所有模式下, 字节与字节之间至少间隔 30 μ s)
- 一个 UART 接口, 支持波特率 1200bps~128kbps
- 一个 CAN 通信接口, 支持波特率 5kbps~1Mbps
- 三种传输方式: 透明传输、带标识符透明传输、自定义转换
- 6 组 CAN 接收过滤器设定
- 工作温度: -40 $^{\circ}$ C~125 $^{\circ}$ C



外观图

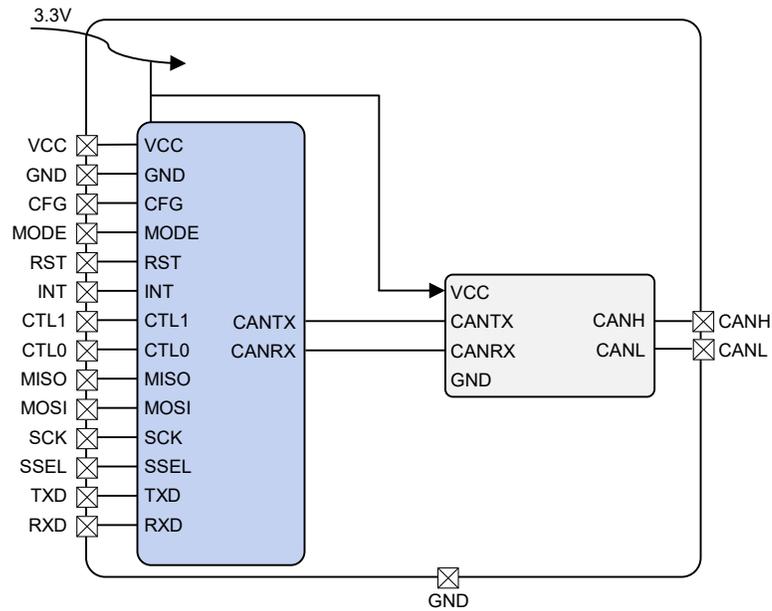
概述

BM42D7611-1 是一款非隔离 SPI/UART 转 CAN 模块, 集成微处理器、CAN 收发器于一体。该模块的工作原理是将 SPI/UART 接口转换为 CAN 接口, 在功能外设不足的情况下, 用户可使用该模块扩展 CAN 接口, 通过 SPI 或者 UART 将数据信息发送给另一个 CAN 接口, 从而实现 SPI 设备和 UART 设备与 CAN 总线网络之间的数据通信。

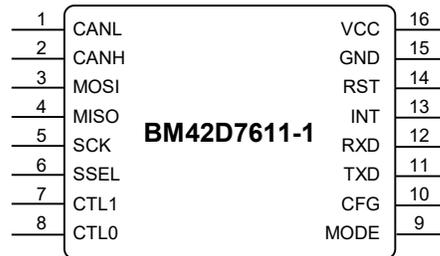
应用领域

- 工业自动化
- 家用电子产品
- 电力监控
- 农业机械
- 机器人

方框图



引脚图



引脚说明

引脚	功能	类型	说明
1	CANL	—	CANL 引脚
2	CANH	—	CANH 引脚
3	MOSI	I	SPI MOSI 引脚
4	MISO	O	SPI MISO 引脚
5	SCK	I	SPI SCK 引脚
6	SSEL	I	SPI 片选引脚
7	CTL1	I	SPI 主机控制引脚 1
8	CTL0	I	SPI 主机控制引脚 0
9	MODE	I	模式控制引脚
10	CFG	I	配置引脚
11	TXD	O	模块 UART 发送引脚
12	RXD	I	模块 UART 接收引脚

引脚	功能	类型	说明
13	INT	O	从机反馈引脚
14	RST	I	复位信号输入，RST 为低电平时模块复位
15	GND	PWR	逻辑负电源
16	VCC	PWR	逻辑正电源

注：PWR：电源；I：输入；O：输出。

技术规格

极限参数

参数	数值	单位
V _{CC}	-0.3~+3.6	V
储存温度	-40~125	°C
储存相对湿度	20%~60%	RH
工作温度	-40~125	°C
环境湿度	10%~95%	RH

建议工作范围

参数	数值	单位
V _{CC}	3.15~3.6	V
工作温度	-40~125	°C
环境湿度	10%~95%	RH

注：极限参数表示超过所规定范围将可能对驱动芯片造成损害。建议工作范围表示芯片可正常工作的条件，但不包含特定限制条件。

直流电气特性

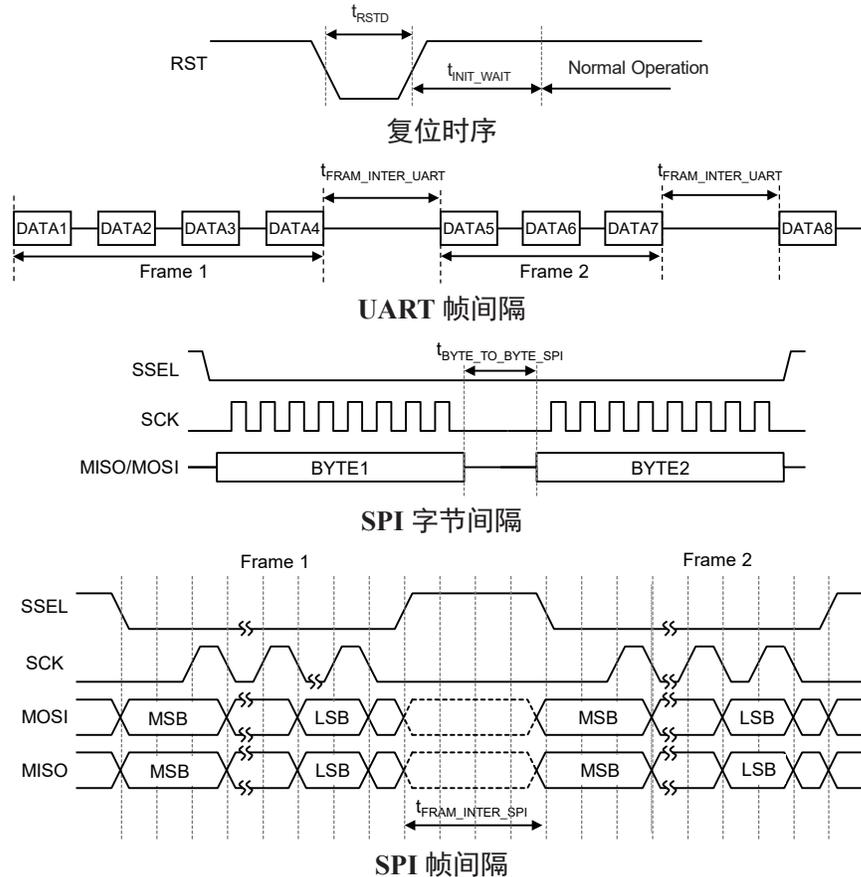
符号	参数	条件	最小	典型	最大	单位
V _{CC}	逻辑电源	—	3.15	—	3.6	V
V _{IL}	输入逻辑低电平电压	3.15≤V _{CC} ≤3.6V	-0.5	—	0.35 V _{CC}	V
V _{IH}	输入逻辑高电平电压	3.15≤V _{CC} ≤3.6V	0.65 V _{CC}	—	0.5 +V _{CC}	V
I _{DD}	工作电流	V _{CC} =3.3V，UART 波特率 128kbps， CAN 波特率 1Mbps	—	—	13	mA

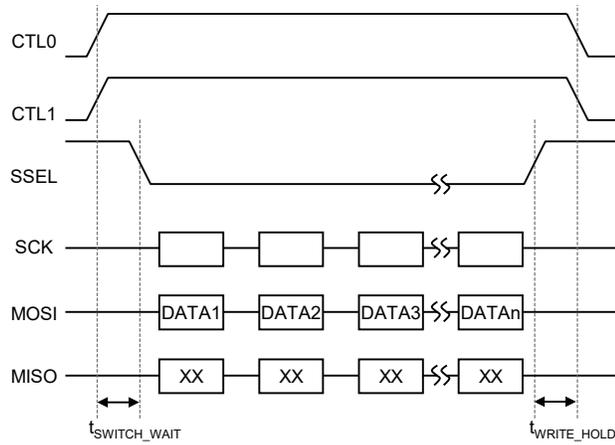
交流电气特性

符号	参数	条件	最小	典型	最大	单位
t _{RSTD}	系统复位 RST 引脚低电平保持时间	图：复位时序	100	—	—	μs
t _{INIT_WAIT}	初始化等待时间	图：复位时序	55	—	—	ms
f _{CLK_UART}	UART 波特率	—	1200	—	128k	bps

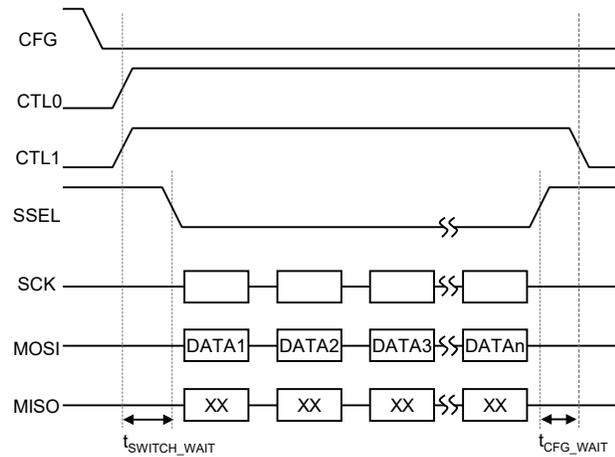
符号	参数	条件	最小	典型	最大	单位
f_{CLK_SPI}	SPI 时钟	非自定义协议转换	—	—	500k	Hz
		自定义协议转换	—	—	300k	
f_{CLK_CAN}	CAN 波特率	—	5k	—	1M	bps
$t_{FRAM_INTER_UART}$	UART 帧间隔 (注)	图: UART 帧间隔	$10 \times n / f_{CLK_UART}$	—	—	s
$t_{BYTE_TO_BYTE_SPI}$	SPI 字节间隔	图: SPI 字节间隔	30	—	—	μs
$t_{FRAM_INTER_SPI}$	SPI 帧间隔	图: SPI 帧间隔	40	—	—	μs
t_{SWITCH_WAIT}	SPI 状态切换操作延时	图: SPI 状态切换	50	—	—	μs
t_{WRITE_HOLD}	SPI 写操作后状态保持时间	图: SPI 状态切换	5	—	—	μs
t_{CFG_WAIT}	配置后等待时间	写配置	260	—	—	ms
		读配置	5	—	—	ms

注: n 为用户配置的 UART 帧间隔字符数。





SPI 状态切换



SPI 写配置

CAN 特性

符号	参数	条件	最小	典型	最大	单位
f_{CLK_CAN}	CAN 通信波特率	—	5	—	1000	kbps
$V_{O(D)}$	输出显性电平 (逻辑 0) 电压	$R_L=60\Omega$, CANH	2.45	—	V_{CC}	V
		$R_L=60\Omega$, CANL	0.5	—	1.25	
$V_{O(R)}$	输出隐性电平 (逻辑 1) 电压	$R_L=60\Omega$, CANH	—	2.3	—	V
		$R_L=60\Omega$, CANL	—	2.3	—	
$V_{DIFF(D)}$	差分输出电压显性	$R_L=60\Omega$	1.2	2	3	V
$V_{DIFF(R)}$	差分输出电压隐性	无负载	-0.5	—	0.05	
V_X	总线侧最大耐压	—	-36	—	36	V
CAN 总线接口		符合 ISO 11898-2 标准, 双绞线输出				

功能描述

系统描述

BM42D7611-1 是一款非隔离 SPI/UART 转 CAN 专业模块，用户可使用该模块将数据通过 UART 或者 SPI 转 CAN 发送。多种传输速度选择，CAN 速率最高可支持 1Mbps，UART 速度最高可支持 128kbps，SPI 最高速度可达 500kHz。三种传输方向选择，分别是双向、仅 SPI/UART 转 CAN、仅 CAN 转 SPI/UART。三种传输模式选择，分别是透明传输模式、透明带标识传输模式、自定义协议传输模式。

工作模式

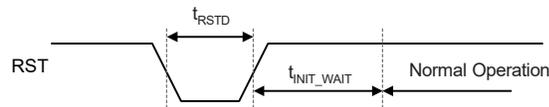
MODE 引脚为高电平时，模块为 SPI 转 CAN 模式，低电平时为 UART 转 CAN 模式。CFG 是控制模块处于配置模式还是数据发送模式。CTL0、CTL1 则是 SPI 转 CAN 模式下控制模块在空闲、主机读状态、主机读数据、主机写数据几种模式中切换。此外，INT 作为反馈引脚，SPI 从机通过拉低 INT 引脚通知主机读取配置回应帧、硬件标识回应帧、配置帧和数据帧，主机读取完相应信息，从机自动拉高 INT。主机每一次发送完配置命令帧后，通过拉低 RST 引脚复位模块，配置才得以完成。

MODE	CFG	工作模式
0	0	UART 转 CAN 模式，UART 配置
0	1	UART 转 CAN 模式，数据传输
1	0	SPI 转 CAN 模式，SPI 配置
1	1	SPI 转 CAN 模式，数据传输

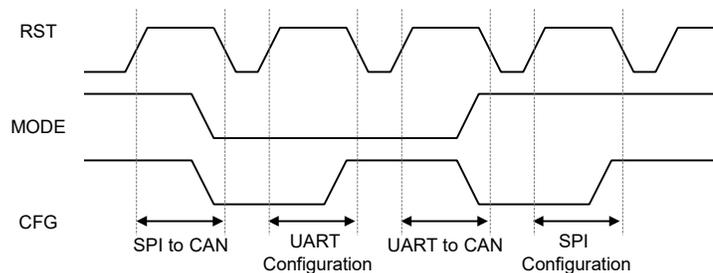
工作模式切换真值表

若要切换工作模式，更改引脚电平后，需要拉低 RST 引脚将产品复位，才能使其进入设定的工作模式。

为使其复位成功，拉低 RST 引脚至少保持 100 μ s，复位后至少等待 55ms，待产品初始化完成。RST 引脚外的其他功能控制引脚，在切换引脚状态后至少等待 50 μ s，预留时间让模块切换工作状态。



复位时序



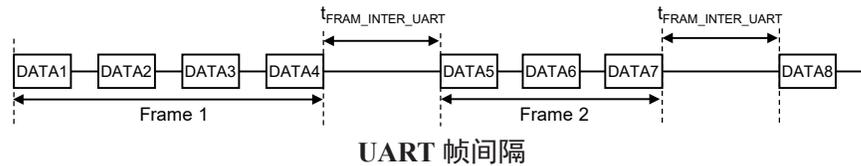
工作模式切换时序图

UART 通信模式

在 UART 模式下，SPI 传输功能禁止，SPI 接口无效。UART 通信格式固定为：1 位起始位，8 位数据位，无校验位，1 位停止位。UART 的通信速率范围为 1200bps~128kbps。

UART 帧

模块 UART 接收到第一个数据开始，直到一定的时间间隔后仍然未收新的数据，这段时间内的数据定义为一帧数据，这个时间间隔定义为 UART 的帧间隔。帧间隔代表 UART 发送 n 个字符数据的时间，这个 n 范围为 0x02~0x0A，由用户配置。不同波特率下，同样发送 n 个字符数据，时间也会不一样。当用户设置的 UART 帧间隔参数为 0x02，也就是 2 个字符时间，波特率为 9600bps，每个字符有 10 位组成（1 位起始位，8 位数据位，1 位停止位），则间隔时间为 $2 \times 10 / 9600 \approx 2\text{ms}$ 。UART 帧格式如下图：



SPI 通信模式

在 SPI 模式下，模块只能作为 SPI 从机，SCK 空闲时为低电平，上升沿读取数据，数据长度为 8，MSB 高位先传输。透明转换、透明带标识符转换下最高通信速率为 500kHz，自定义协议转换最高通信速率为 300kHz。

在 SPI 通信模式下，UART 传输功能禁止，模块不能通过 URAT 转 CAN 发送和接收数据。

主机控制

CFG	CTL0	CTL1	功能
0	0	0	SPI 配置，空闲
0	0	1	SPI 配置，主机读状态
0	1	0	SPI 配置，主机读回应帧、配置帧
0	1	1	SPI 配置，主机写命令
1	0	0	数据传输，空闲
1	0	1	数据传输，主机读状态
1	1	0	数据传输，主机读数据
1	1	1	数据传输，主机写数据

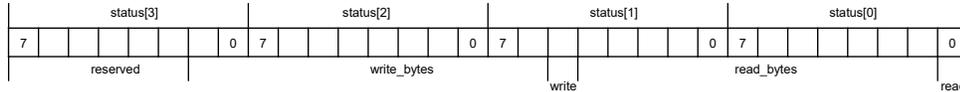
SPI 转 CAN 模式主机控制真值表

在 SPI 转 CAN 模式下，当主机功能选择为读状态，即 CTL0、CTL1 为 0、1 状态，主机可获取可以读的字节数以及可以写入的字节数。主机通过 SPI 读取 4 个字节即可获得模块当前状态。

位	含义	符号	描述
0	可读标识位	read	CAN 接收缓冲区不为空为 1，反之为 0
12:1	可读字节数	read_bytes	主机可以读的 CAN 数据字节数
13	可写标识位	write	CAN 发送缓冲区还未填满时为 1，反之为 0

位	含义	符号	描述
25:14	可写字节数	write_bytes	主机可以写入模块的字节数
31:26	保留位	reserved	保留

SPI 模式状态码构成



状态字节数据结构

用户在获取状态 4 个字节后，可提取出相应位用作相应判断和处理。示例代码如下：

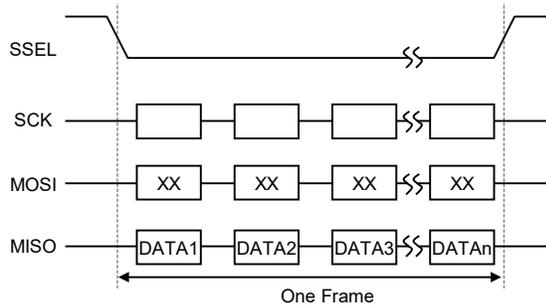
```

read = status[0]&0x01;           // 可读标识位
read_bytes = ((status[0] & 0xfe)>>1) + ((status[1] & 0x1f)<<7);
                                // 可读字节数
write = (status[1] & (1<<5)) ? 1:0; // 可写标识位
write_bytes = ((status[1]&0xc0)>>6)+((status[2]&0xff)<<2)+((status[3]&0x03)<<10);
                                // 可写字节数

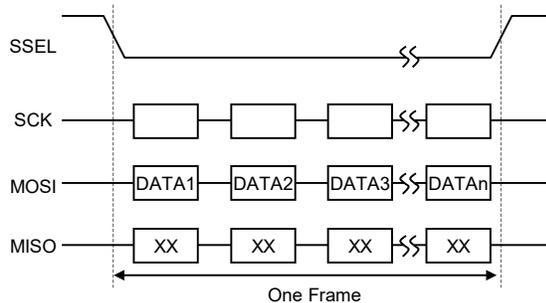
```

SPI 帧

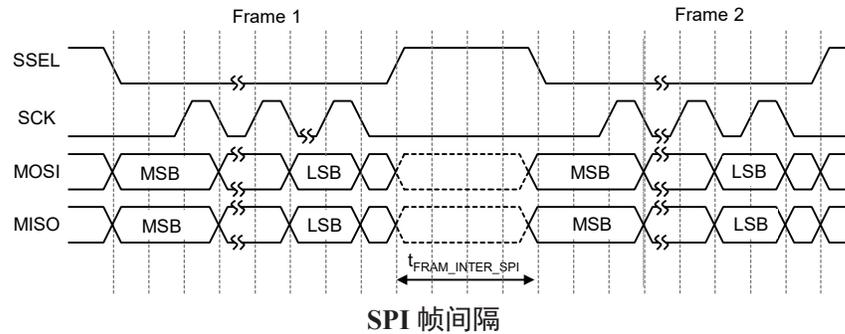
SPI 一次片选有效至片选无效之间的数据定义为一帧数据。模块 SPI 接收缓存区最多能存放 124 个字节，主机 SPI 发数据给模块的一帧数据最多不能超过 124 个字节，当模块缓存区的数据满了之后，不再接收数据，模块将 SPI 接收缓冲区的数据通过 CAN 发送完成后，SPI 接收缓存区方可接收新的数据。



SPI 读取数据



SPI 写数据



INT 反馈引脚

模块只能做 SPI 从机，不能主动控制片选引脚从而发送数据给主机。因此当模块 CAN 接收到数据后且满足一定条件才拉低 INT 引脚通知主机读取数据，当主机读取完数据，INT 引脚恢复高电平。有以下两种情况 INT 引脚才会拉低：

- CAN 接收缓冲区 CAN 帧数达到设置的触发帧数时（这个触发帧数是在配置模式下由主机发送的配置帧设定）
当模块 CAN 接收的帧数达到配置的触发帧数时，INT 引脚拉低，直至主机读取完 CAN 接收缓冲区的数据后，INT 引脚恢复高电平。
- CAN 缓冲有接收到数据但少于触发帧数，在设定时间内主机未读取时（这个时间是在配置模式下由主机发送的配置帧设定）
CAN 接收缓冲区有数据但收到的数据未达到触发帧数，且主机在模块 CAN 接收到数据后超过一定时间未读取数据，INT 引脚也会拉低从而通知主机读取数据，直至主机读取完 CAN 接收缓冲区的数据后，INT 引脚恢复高电平。

数据转换方式

数据转换方式指的是串行总线和 CAN 总线之间数据转换的基本准则，同一时间内模块只能工作于其中一种数据转换方式，如果要改变转换方式则需要重新配置模块。模块支持透明转换、透明带标识符转换和自定义协议转换三种方式。

透明转换

透明转换是指任何一侧总线接收到数据，不做任何处理就将数据发送到另一侧总线上。

串行帧向 CAN 帧转换

一帧 CAN 帧的数据长度最多有 8 个字节，串行帧的数据长度小于或等于 8 个字节时，数据通过一个 CAN 帧发送出去。串行帧的数据长度大于 8 个字节时，模块每次取串行帧的 8 个字节填充到 CAN 帧中通过 CAN 接口转发出去，直到所有数据全部转发完成。

CAN 帧中的数据长度由实际分配到 CAN 帧的数据的字节数决定。此时的帧 ID、帧类型由用户配置决定，若要重新更改帧 ID、帧类型，需要用户重新配置，重启后配置命令生效。数据转换示意如下：



串行帧转 CAN 帧 (透明转换, 数据 ≤ 8 个字节)

串行帧	数据 (= 8 字节)				数据 (≤ 8 字节)				
CAN 帧 1	帧类型	帧 ID	数据长度	数据域					
CAN 帧 2	帧类型	帧 ID	数据长度	数据域					

串行帧转 CAN 帧 (透明转换, 数据 > 8 个字节)

● 转换范例

范例 1: 用户配置的发送 CAN 帧类型为标准帧, 帧 ID 为 0x0123 (标准帧 ID 长度共 11 位, 范围为: 0x000~0x7FF), 串行帧的数据为 0x01~0x08, 共 8 个字节, 串行帧转换至 CAN 帧格式如下:

串行帧	数据										
	01	02	03	04	05	06	07	08			
CAN 帧	帧类型	帧 ID1	帧 ID0	数据长度	数据域						
	标准帧	01	23	08	01	02	03	04	05	06	07

串行帧转 CAN 帧范例 (数据 ≤ 8 个字节)

范例 2: 用户配置的发送 CAN 帧类型为扩展帧, 帧 ID 为 0x00012345 (扩展帧 ID 长度共 29 位, 范围为: 0x00000000~0x1FFFFFFF), 串行帧的数据为 0x01~0x0C 共 12 个字节, 串行帧转换至 CAN 帧格式如下:

串行帧	数据								数据				
	01	02	03	04	05	06	07	08	09	0A	0B	0C	
CAN 帧 1	帧类型	帧 ID3	帧 ID2	帧 ID1	帧 ID0	数据长度	数据域						
	扩展帧	00	01	23	45	08	01	02	03	04	05	06	07
CAN 帧 2	帧类型	帧 ID3	帧 ID2	帧 ID1	帧 ID0	数据长度	数据域						
	扩展帧	00	01	23	45	04	09	0A	0B	0C			

串行帧转 CAN 帧范例 (数据 > 8 个字节)

CAN 帧向 UART 帧转换

模块 CAN 收到一帧数据之后就立即转发到 UART 接口。所以每个 CAN 帧数据转为一个 UART 帧数据。

● 帧 ID 和帧信息转换除能

CAN 帧转 UART 帧时, UART 帧中只有 CAN 帧中的数据, 无帧信息和帧 ID 字节信息, 转换示意图如下:

CAN 帧	帧类型	帧 ID	数据长度	数据域
UART 帧	数据			

CAN 转 UART 帧 (透明转换)

◆ 转换范例

若 CAN 接口接收的第一帧 CAN 帧为标准帧，ID 为 0x0123，数据为 0x01~0x06 共 6 个字节；则 CAN 帧和转换后的 UART 帧如下：

CAN 帧	帧类型	帧 ID1	帧 ID0	数据长度	数据域							
	标准帧	01	23	06	01	02	03	04	05	06		
UART 帧	数据											
	01	02	03	04	05	06						

CAN 转 UART 帧范例 (透明转换)

● 帧信息转换使能

帧信息转换使能后，UART 帧也包含了 CAN 帧的帧类型、数据长度和数据。转换后，UART 帧以一个帧信息字节为开头，其后是数据，帧信息以一个字节表示，包括帧类型，数据长度。高四位代表帧类型，低四位代表长度。高四位“0b1000”代表扩展帧，“0b0000”代表标准帧。低四位代表数据长度，范围为“0b0000~0b1000”分别代表“0~8”字节数据。

例如：帧信息为 0x06，即表示接收的 CAN 帧类型为标准帧，CAN 数据长度为 6 个字节；

帧信息为 0x87，即表示接收的 CAN 帧类型为扩展帧，CAN 数据长度为 7 个字节。

CAN 帧转换成串行帧后，每一个 CAN 帧转换成一个 UART 帧，转换示意图如下：



CAN 转 UART 帧 (透明转换, 帧信息转换使能)

◆ 转换范例

若 CAN 接口接收的第一帧 CAN 帧为标准帧，ID 为 0x0123，数据为 0x01~0x06 共 6 个字节；则 CAN 帧和转换后的 UART 帧如下：

CAN 帧	帧类型	帧 ID1	帧 ID0	数据长度	数据域								
	标准帧	01	23	06	01	02	03	04	05	06			
UART 帧	帧信息	数据											
	06	01	02	03	04	05	06						

CAN 转 UART 帧范例 (透明转换, 帧信息转换使能)

说明：UART 帧信息 (06) = 标准 CAN 帧 (0) 和数据长度 (6) 的组合。

● 帧 ID 转换使能

用户配置帧 ID 转换使能后，默认使能帧信息转换，所以当开启帧 ID 转换后，CAN 帧转成 UART 帧，UART 帧的字节按顺序包括一字节的帧信息 (帧信息包括帧类型、数据长度)、帧 ID (标准帧 2 个字节，扩展帧 4 个字节)、帧数据。CAN 帧转 UART 帧的帧 ID 高位在前，低位在后，转换示意如下：



CAN 转 UART 帧 (透明转换, 帧 ID 转换使能)

◆ 转换范例

若用户配置帧 ID 转换使能，默认使能帧信息转换。CAN 接口收到的 CAN 帧为标准帧，ID 为 0x0123，数据为 0x01~0x07 共 7 个字节；则 CAN 帧和转换后的 UART 帧如下：

CAN 帧	帧类型	帧 ID1	帧 ID0	数据长度	数据域						
	标准帧	01	23	07	01	02	03	04	05	06	07
UART 帧	帧信息	帧 ID1	帧 ID0	数据							
	07	01	23	01	02	03	04	05	06	07	

CAN 转 UART 帧范例 (透明转换, 帧 ID 转换使能)

CAN 帧向 SPI 帧转换

SPI 始终作为从机，当模块从 CAN 总线收到一帧数据后立即存储到 CAN 缓冲区。接收的帧数到达反馈帧数或者触发时间到达，INT 引脚拉低通知主机读数据。主机在检测到 INT 引脚为低电平时，可以通过一帧 SPI 帧取出 CAN 缓冲区中的所有数据。

CAN 帧转 SPI 帧时，有三种数据转换格式。第一种，CAN 端接收到一个 CAN 帧数据时，只有 CAN 数据存入缓冲区。第二种，CAN 端接收到一个 CAN 帧数据时，CAN 帧的帧信息和数据一起存入缓冲区中，帧信息是指 CAN 帧的帧类型和数据长度，占一个字节，数据在其后，这种转换方式叫帧信息转换使能。第三种，CAN 端接收到一个 CAN 帧数据时，CAN 帧的帧信息、帧 ID、帧数据一起存入缓冲区中，帧信息为一个字节，包括数据长度和帧类型，帧 ID 为 2 个字节 (标准帧) 或者 4 个字节 (扩展帧)，按帧信息、帧 ID、数据的顺序，这种转换方式叫做帧 ID 转换使能。以下一一介绍说明。

● 帧 ID 和帧信息转换除能

CAN 帧转 SPI 帧时，SPI 帧中只有 CAN 帧中的数据，无帧信息和帧 ID 字节信息，转换示意如下：

CAN 帧 1	帧类型	帧 ID	数据长度	数据域
CAN 帧 n	帧类型	帧 ID	数据长度	数据域
SPI 帧	数据段 1	...	数据段 n	

CAN 转 SPI 帧 (透明转换)

◆ 转换范例

若 CAN 接口接收的第一帧 CAN 帧为标准帧，ID 为 0x0123，数据为 0x01~0x06，共 6 个字节；接收到的第 n 帧为扩展帧，ID 为 0x00012345，数据为 0x00、0x11~0x16，共 7 个字节；则 CAN 帧和转换后的 SPI 帧如下：

CAN 帧 1	帧类型	帧 ID	数据长度	数据域									
	标准帧	0123	06	01	02	03	04	05	06				
CAN 帧 n	帧类型	帧 ID	数据长度	数据域									
	扩展帧	00012345	07	00	11	12	13	14	15	16			
SPI 帧	数据段 1			...	数据段 n								
	01	02	03	04	05	06	...	00	11	12	13	14	15

CAN 转 SPI 帧范例 (透明转换)

● 帧信息转换使能

帧信息转换使能后，SPI 帧包含了 CAN 帧的帧类型、数据长度和数据。SPI 帧以一个帧信息字节为开头，其后是数据。帧信息以一个字节表示，包括帧类型，数据长度。高四位代表帧类型，低四位代表长度。高四位“0b1000”代表扩展帧，“0b0000”代表标准帧。低四位代表数据长度，范围为“0b0000~0b1000”分别代表“0~8”字节数据。

例如：帧信息为 0x06，即表示接收的 CAN 帧类型为标准帧，CAN 数据长度为 6 个字节；

帧信息为 0x87，即表示接收的 CAN 帧类型为扩展帧，CAN 数据长度为 7 个字节。

帧信息转换使能后 CAN 帧转 SPI 帧的帧格式数据如下：



◆ 转换范例

若 CAN 接口接收到的第一个 CAN 帧为标准帧，ID 为 0x0123，数据为 0x01~0x06，共 6 个字节；第 n 帧为扩展帧，ID 为 0x00012345，数据为 0x00，0x11~0x16，共 7 个字节；则 CAN 帧转换后的 SPI 帧如下：



说明：SPI 帧信息 1 (06) = 标准 CAN 帧 1(0) 和数据长度 (6) 的组合，
SPI 帧信息 n (87) = 扩展 CAN 帧 n (8) 和数据长度 (7) 的组合。

● 帧 ID 转换使能

用户配置帧 ID 转换使能后，默认开启帧信息转换，所以当开启帧 ID 转换使能后，CAN 帧转成 SPI 帧，SPI 帧的字节段按顺序包括一字节的帧信息 (帧信息包括帧类型、数据长度)、帧 ID (标准帧 2 个字节，扩展帧 4 个字节)、帧数据。CAN 帧转 SPI 帧的帧 ID 高位在前，低位在后，转换示意如下：



◆ 转换范例

若用户配置帧 ID 转换使能，因此默认开启帧信息转换。CAN 接口收到的第一个 CAN 帧为标准帧，ID 为 0x0123，数据为 0x01~0x06，共 6 个字节；第 n 帧为扩展帧，ID 为 0x00012345，数据为 0x00, 0x11~0x16，共 7 个字节；则 CAN 帧和转换后的 SPI 帧如下：

CAN 帧 1	帧类型	帧 ID	数据长度	数据域															
	标准帧	0123	06	01	02	03	04	05	06										
CAN 帧 n	帧类型	帧 ID	数据长度	数据域															
	扩展帧	00012345	07	00	11	12	13	14	15	16									
SPI 帧	帧信息 1	帧 ID	数据段 1				...	帧信息 n	帧 ID	数据段 n									
	06	01 23	01	02	03	04	05	06	87	00	01	23	45	00	11	12	13	14	15

CAN 转 SPI 帧范例 (透明转换, 帧 ID 转换使能)

透明带标识符转换

透明带标识是指发送或者接收的串行帧中都包含了有效的 CAN 帧 ID 字节。透明带标识符转换模式下，在串行帧转 CAN 帧之前，用户已经配置好了帧 ID 的起始地址和长度，模块在转换时根据配置好的帧 ID 起始地址和长度，从串行帧中提取帧 ID 填充在 CAN 帧的帧 ID 域里，数据则填充在 CAN 帧的数据域里。同样的当 CAN 帧转串行帧时，模块从 CAN 帧中提取出帧 ID 转换在串行帧的相应位置，同时返回帧信息。在透明带标识符转换方式下，用户可以在同一节点发送不同 ID 的 CAN 数据。

串行帧向 CAN 帧转换

串行帧中的帧 ID 起始地址和长度由用户配置设定，起始地址范围是 0~7，长度范围分别是 1~2 (标准帧) 或者 1~4 (扩展帧)。若要重新更改串行帧 ID 的起始地址和长度，需要用户通过发送配置命令帧重新配置，重启后配置命令生效。

当串行帧数据长度小于或等于 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。

在透明带标识符转换模式下，CAN 帧中的帧类型、串行帧中的帧 ID 的起始地址、长度由用户配置且始终保持不变，除非用户重新对模块进行配置，CAN 中的数据长度根据实际分配到 CAN 的数据字节数决定。

串行帧	数据	帧 ID	数据
-----	----	------	----

CAN 帧	帧类型	帧 ID	数据长度	数据域
-------	-----	------	------	-----

串行帧转 CAN 帧 (透明带标识符转换, 数据 ≤ 8 个字节)

串行帧	数据 1	帧 ID	数据 2	数据 3
-----	------	------	------	------

CAN 帧 1	帧类型	帧 ID	数据长度	数据 1	数据 2
---------	-----	------	------	------	------

CAN 帧 2	帧类型	帧 ID	数据长度	数据 3
---------	-----	------	------	------

串行帧转 CAN 帧 (透明带标识符转换, 数据 > 8 个字节)

● 转换范例

若用户配置 CAN 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，串行帧发送的数据是 0x00~0x0F，共 16 个字节。由于配置的 CAN 帧为扩展帧，所以 CAN 帧的帧 ID 为 4 字节。但配置的 ID 在串行帧的起始地址是 2，长度是 3，所以从串行帧提取出来填充到 CAN 帧 ID 域里的 ID 为 0x020304，不足部分在低字节补 0，故 CAN 帧 ID 为 0x02030400，串行帧转 CAN 示意如下：

字节地址	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
串行帧	数据		ID			数据										
	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
CAN 帧 1	帧类型	帧 ID			数据长度	数据										
	扩展帧	02	03	04	00	08	00	01	05	06	07	08	09	0A		
CAN 帧 2	帧类型	帧 ID			数据长度	数据										
	扩展帧	02	03	04	00	05	0B	0C	0D	0E	0F					

串行帧转 CAN 帧范例 (透明带标识符转换)

CAN 帧向 UART 帧转换

模块接收到一帧 CAN 帧就立即向 UART 接口转发一帧，所以在透明带标识符转换模式下，CAN 转 UART，UART 帧是 CAN 帧信息、帧 ID、帧数据的集合。帧 ID 起始地址和长度由用户配置设定，起始地址范围是 0~7，长度范围分别是 1~2 (标准帧) 或者 1~4 (扩展帧)。若要重新更改串行帧 ID 的起始地址和长度，需要用户通过发送配置命令帧重新配置，重启后配置命令生效。

帧信息以一个字节表示，包括帧类型，数据长度。高四位代表帧类型，低四位代表长度。高四位“0b1000”代表扩展帧，“0b0000”代表标准帧。低四位代表数据长度，范围为“0b0000~0b1000”分别代表“0~8”字节数据。

例如：帧信息为 0x06，即表示接收的 CAN 帧类型为标准帧，CAN 数据长度为 6 个字节；

帧信息为 0x85，即表示接收的 CAN 帧类型为扩展帧，CAN 数据长度为 5 个字节。

CAN 帧	帧类型	帧 ID	数据长度	数据域
UART 帧	帧信息	数据	帧 ID	数据

CAN 帧转 UART 帧 (透明带标识符转换)

● 转换范例

假设用户配置接收 CAN 帧类型为扩展帧，帧 ID 的起始地址配置为 2，长度是 3，帧 ID 为 0x10123456，数据为 0x01~0x08 共 8 个字节，转换示意如下：

CAN 帧	帧类型	帧 ID			数据长度	数据							
	扩展帧	10	12	34	56	08	01	02	03	04	05	06	07
字节地址		0	1	2	3	4	5	6	7	8	9	10	
UART 帧	帧信息	数据		ID		数据							
	88	01	02	10	12	34	03	04	05	06	07	08	

CAN 转 UART 帧范例 (透明带标识符转换)

CAN 帧向 SPI 帧转换

在透明带标识符转换模式下，转换后的 SPI 帧根据用户配置的帧 ID 起始地址和长度将 CAN 帧的 ID 和数据填入 SPI 帧中，CAN 帧信息作为首字节先存储到 SPI 帧中。所以在透明带标识符转换模式下，CAN 转 SPI，SPI 帧包含了 CAN 帧信息、帧 ID (高位在前) 和帧数据。

帧 ID 起始地址和长度由用户配置设定，起始地址范围是 0~7，长度范围分别是 1~2 (标准帧) 或者 1~4 (扩展帧)。若要重新更改串行帧 ID 的起始地址和长度，需要用户通过发送配置命令帧重新配置，重启后配置命令生效。

帧信息以一个字节表示，包括帧类型，数据长度。高四位代表帧类型，低四位代表长度。高四位“0b1000”代表扩展帧，“0b0000”代表标准帧。低四位代表数据长度，范围为“0b0000~0b1000”分别代表“0~8”字节数据。

例如：帧信息为 0x06，即表示接收的 CAN 帧类型为标准帧，CAN 数据长度为 6 个字节；

帧信息为 0x87，即表示接收的 CAN 帧类型为扩展帧，CAN 数据长度为 7 个字节。



CAN 转 SPI 帧 (透明带标识符转换)

● 转换范例

假设用户配置接收 CAN 帧类型为标准帧，帧 ID 的起始地址配置为 2，长度是 2，第一帧的帧 ID 为 0x0123，数据为 0x01~0x06 共 6 个字节；第 n 帧 ID 为 0x0345，数据为 0x07~0x0A 共 4 个字节；转换示意如下：



CAN 转 SPI 帧范例 (透明带标识符转换)

自定义协议转换

自定义协议转换下串行帧由帧头、帧长度、帧类型、帧 ID、数据、帧尾组成。

帧头：串行帧开始字节，范围为 0x00~0xFF，由用户配置。

帧长度：帧类型、帧 ID、数据总共的字节数。

帧类型：CAN 帧的帧类型，0x00 代表标准帧，0x08 代表扩展帧。

帧 ID：CAN 帧的 ID，标准帧 ID 由 2 个字节组成，扩展帧 ID 由 4 个字节组成。

数据：CAN 帧数据。

帧尾：串行帧的结束字节，范围为 0x00~0xFF，由用户配置。

用户发送的串行帧符合定义的格式，模块才会接收串行帧数据并将数据进行转换，否则不做任何处理直接丢弃。

在自定义协议转换方式下，用户可以在同一节点发送不同 ID 的 CAN 帧数据，也可以在接收到的串行帧中获取 CAN 接口接收帧的帧 ID。

串行帧向 CAN 帧转换

当串行帧数据长度小于或等于 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。转换示意如下：



● 转换范例

假设用户配置的串行帧头为 0x10，帧尾为 0x1B。发送的帧类型为标准帧 (0x00)，帧 ID 为 0x0123，数据为 0x01~0x0A 共 10 个字节，则帧长度为帧类型 (1 字节)、帧 ID (2 字节)、帧数据 (10 字节) 的字节个数和，也就是 0x0D。转换示意如下所示。



CAN 帧向 UART 帧转换

当 CAN 接口收到一帧数据之后就立即转换成一帧 UART 帧通过 UART 接口转发出去。转换示意如下所示。



● 转换范例

假设用户配置的串行帧头为 0x10，帧尾为 0x1B，CAN 接口接收到的 CAN 帧类型为扩展帧，帧 ID 为 0x01234567，数据为 0x01~0x06 共 6 个字节，转换示意如下所示。

CAN 帧	帧类型	帧 ID				数据长度	数据							
	扩展帧	01	23	45	67	06	01	02	03	04	05	06		
UART 帧	帧头	帧长度	帧类型	帧 ID				数据						帧尾
	10	0B	08	01	23	45	67	01	02	03	04	05	06	1B

CAN 帧转 UART 帧范例 (自定义转换)

说明：串行帧长度 (0x0B) = 帧类型 (1 个字节) + 帧 ID (4 个字节) + 数据 (6 个字节)。

CAN 帧向 SPI 帧转换

当 CAN 接口接收到数据时，模块将收到的 CAN 数据转换为用户定义格式的串行帧存入缓冲区中，模块收到数据后若满足触发帧数或者反馈触发时间，会主动拉低 INT 脚通知 SPI 主机读取数据。主机 SPI 可以通过一帧读取存有将 CAN 接收到的数据转换成用户定义格式的 SPI 帧数据。转换示意如下：

CAN 帧 1	帧类型	帧 ID	数据长度	数据域									
CAN 帧 n	帧类型	帧 ID	数据长度	数据域									
SPI 帧	帧头	帧长度	帧类型	帧 1 ID	数据	帧尾	...	帧头	帧长度	帧类型	帧 n ID	数据	帧尾

CAN 帧转 SPI 帧 (自定义转换)

● 转换范例

假设用户配置的串行帧帧头为 0x10，帧尾为 0x0B，CAN 接口接收到的帧类型为标准帧，第一帧帧 ID 为 0x0123，数据为 0x01~0x06 共 6 个字节；第 n 帧帧 ID 为 0x0234，数据为 0x09~0x0C 共 4 个字节，转换示意如下所示。

CAN 帧 1	帧类型	帧 ID	数据长度	数据									
	标准帧	01	23	06	01	02	03	04	05	06			
CAN 帧 n	帧类型	帧 ID	数据长度	数据									
	标准帧	02	34	04	09	0A	0B	0C					
SPI 帧	数据段 1												
	帧头	帧长度	帧类型	帧 1 ID	数据						帧尾		
	10	09	00	01	23	01	02	03	04	05	06	0B	
	数据段 n												
...	帧头	帧长度	帧类型	帧 n ID	数据				帧尾				
...	10	07	00	02	34	09	0A	0B	0C	0B			

CAN 帧转 SPI 帧范例 (自定义转换)

说明：数据段 1 的帧长度 (0x09) = 帧类型 (1 个字节) + 帧 1 ID (2 个字节) + 数据 (6 个字节)。

配置说明

转换参数

数据转换方式

数据转换方式是指串行总线与 CAN 总线之间的数据转换规则，模块工作时只能工作于一种数据转换方式。模块支持透明转换、透明带标识符转换、自定义协议转换三种方式，各转换方式详细说明见透明转换、透明带标识符转换、自定义协议转换章节。

转换方向

是指数据允许转换的方向，模块支持三种转换方向：双向 SPI/UART 与 CAN 互转、单向 SPI/UART 转 CAN、单向 CAN 转 SPI/UART。

CAN 帧信息转换到串行帧

仅在透明转换方式下有效，使能后，CAN 转 SPI/UART 时，CAN 帧的帧类型和数据长度同时转换至 SPI/UART 帧，详细说明参考透明转换章节的帧信息转换使能小节。

CAN 帧 ID 转换到串行帧

仅在透明转换方式下有效，使能后，会同时使能帧信息转换，CAN 转 SPI/UART 时，CAN 帧的帧类型、数据长度和帧 ID 同时转换至 SPI/UART，详细说明参考透明转换章节的帧 ID 转换使能小节。

CAN 帧 ID 在串行帧中的位置

仅在透明带标识符转换方式下有效，包括 CAN 帧 ID 的起始地址和长度，详细说明参考透明带标识符转换章节。

帧头和帧尾

仅在自定义协议转换方式下有效，用于设置串行帧的帧头和帧尾，详细说明参考自定义协议转换章节。

SPI 参数

反馈触发帧数

仅在 SPI 转 CAN 模式下有效。模块作为 SPI 从机无法主动发送数据，故当模块接收到一定数量的 CAN 帧数据后，通过拉低 INT 引脚通知主机读取数据。反馈触发帧数以接收的 CAN 帧为单位，当接收的 CAN 帧达到设定数量后，拉低 INT 引脚触发反馈。

反馈触发时间

仅在 SPI 转 CAN 模式下有效。模块作为 SPI 从机无法主动发送数据，当接收的 CAN 帧未达到设定数量，并超过反馈时间未被读取数据时，模块也会拉低 INT 引脚触发反馈。

UART 参数

波特率

指 UART 发送数据的波特率。有效串口波特率见 UART 和 CAN 波特率代码表。

帧间隔

指 UART 通信帧之间的时间间隔，详细说明参考 UART 通信模式章节。

CAN 参数

波特率

指 CAN 的工作波特率。有效 CAN 波特率见 UART 和 CAN 波特率代码表。

发送帧类型

仅在透明转换、透明带标识符转换方式下有效。指发送的 CAN 帧类型，包括标准帧和扩展帧。

发送标识符

仅在透明转换方式下有效。CAN 发送帧的 ID 以设定 ID 为准。发送 CAN 帧为标准帧，则发送 CAN 帧 ID 范围为 0x000~0x7FF；发送 CAN 帧为扩展帧，则发送 CAN 帧 ID 范围为 0x00000000~0x1FFFFFFF；帧 ID 从左到右依次为 ID3、ID2、ID1、ID0，帧 ID3 为最高字节。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 0x00000123。

接收过滤使能

当接收过滤使能后，模块 CAN 接口会根据过滤模式及屏蔽码、验收码筛选接收 CAN 帧。当接收过滤除能后，模块 CAN 接口会接收总线上的所有 CAN 帧。

接收过滤模式

接收过滤分为扩展帧过滤和标准帧过滤。

屏蔽码和验收码

屏蔽码和验收码配合使用从而实现过滤模式。

当屏蔽码某位为 1，则在相应的位上，接收的 CAN 帧 ID 和验收码相同才会接收，否则不接收。

当屏蔽码某位为 0，则在相应的位上，接收的 CAN 帧 ID 和验收码不论相同与否都会接收。

屏蔽位	验收位	帧 ID	接收或拒绝
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

屏蔽码与验收码真值表

● 举例 1

假设模块被设置为扩展帧过滤，屏蔽码为 0x1FFFFFFF，验收码为 0x00002345。

屏蔽码：0b 0001 1111 1111 1111 1111 1111 1111

验收码：0b 0000 0000 0000 0000 0010 0011 0100 0101

因为屏蔽码所有位都为 1，所以总线上帧 ID 与验收码一致的扩展帧才能被接收，即只有帧 ID 为 0x00002345 的扩展帧才能被模块接收。

- 举例 2
假设模块被设置为标准帧过滤，屏蔽码为 0x7F0，验收码为 0x345。
屏蔽码：0b 0111 1111 0000
验收码：0b 0011 0100 0101
总线上 ID 为 0b 0011 0100 XXXX 的标准帧都能接收，即 ID 为 0x340~0x34F 的标准帧都能接收，其中 X 代表 0 或 1。
- 举例 3
假设模块被设置为标准帧过滤，屏蔽码为 0x1312，验收码为 0x0301。
屏蔽码：0b 0001 0011 0001 0010
验收码：0b 0000 0011 0000 0001
因为模块被设置为标准帧，而标准帧 ID 范围为 0x000~0x7FF，超出范围的位不会参与和验收码比较。因此，总线上只要 ID 为 0b 0X11 XXX0 XX0X 的标准帧都能接收，其中 X 代表 0 或 1。
- 举例 4
假设模块被设置为标准帧过滤，屏蔽码为 0x00 00 00 00，验收码为 0x0301，此时因为屏蔽码所有位都为 0，因此总线上的标准帧 ID 不管是什么都能接收，即模块能接收总线上所有的标准帧，也就是和接收过滤除能一样。

写配置参数

当用户需要配置相应模式转换功能时，需要发送以下固定帧格式、多个字节组成的命令帧，配置完成后掉电也会保存数据，如要更换转换功能，重新配置即可。配置命令如下表。

写配置命令格式

帧起始	命令码	数据长度	数据域	校验码
2 字节	1 字节	1 字节	60 字节	1 字节
0xF7, 0xF8	0x01	0x3C	定义如下表	前面所有字节的异或

帧起始：2 字节，依次为 0xF7、0xF8。

命令码：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。

校验码：1 字节，为前面所有字节的异或。

序号	参数	数据范围	默认值	说明
0	UART 波特率	0x00~0x10	0x0C (128kbps)	更改 UART 波特率，详细见 UART 和 CAN 波特率代码表
1	UART 数据位	0x08	0x08	固定为 8 位数据
2	UART 停止位	0x01	0x01	固定为 1 位停止位
3	UART 校验位	0x00	0x00	固定为无校验
4	CAN 波特率	0x00~0x0F	0x08 (125kbps)	更改 CAN 波特率，详细见 UART 和 CAN 波特率代码表
5~7	保留字节	—	0x00 00 00	未使用，无效
8	CAN 接收过滤使能	0x00/0x01	0x00	0x00：除能 0x01：使能

序号	参数	数据范围	默认值	说明
9	接收过滤模式	0x08/0x00	0x08	0x00: 仅标准帧过滤 0x08: 仅扩展帧过滤
10~13	屏蔽码	0x00 00 00 00 ~ 0xFF FF FF FF	0xFF FF FF FF	
14~17	验收码 0	0x00 00 00 00 ~ 0xFF FF FF FF	0x00 00 00 00	
18~21	验收码 1	0x00 00 00 00 ~ 0xFF FF FF FF	0x00 00 00 00	
22~25	保留字节	—	0x00 00 00 00	未使用, 无效
26~29	验收码 2	0x00 00 00 00 ~ 0xFF FF FF FF	0x00 00 00 00	
30~33	验收码 3	0x00 00 00 00 ~ 0xFF FF FF FF	0x00 00 00 00	
34~37	验收码 4	0x00 00 00 00 ~ 0xFF FF FF FF	0x00 00 00 00	
38~41	验收码 5	0x00 00 00 00 ~ 0xFF FF FF FF	0x00 00 00 00	
42	转换模式	0x01~0x03	0x02	0x01: 透明转换 0x02: 透明带标识符转换 0x03: 自定义协议转换
43	转换方向	0x00~0x02	0x00	0x00: 双向 0x01: 仅 SPI/UART 转 CAN 0x02: 仅 CAN 转 SPI/UART
44	UART 帧间隔	0x02~0x0A	0x02 (2 个字符)	UART 帧间隔 = $10 \times n / f_{CLK_UART}$, $f_{CLK_UART} = 1200\text{bps} \sim 128\text{kbps}$
45	CAN 帧信息转发到串行帧中	0x00/0x01	0x00	0x00: 除能 0x01: 使能 该项只用于透明转换。
46	CAN 帧 ID 转发到串行帧中	0x00/0x01	0x00	0x00: 除能 0x01: 使能 该项只用于透明转换。
47	CAN 发送帧类型	0x00/0x08	0x08	0x00: 标准帧 0x08: 扩展帧 该项用于透明转换、透明带标识符转换。
48~51	CAN 发送标识符	0x00 00 00 00 ~ 0x1F FF FF FF	0x00 00 00 00	该项只用于透明转换, 标识 ID3~ID0。 标准帧: 0x000~0x7FF 有效; 扩展帧: 0x00000000~0x1FFFFFFF 有效。
52	串行帧中 CAN 标识的长度	0x01~0x04	0x04	以字节为单位, 该项只用于透明带标识符转换。
53	串行帧中 CAN 标识起始地址	0x00~0x07	0x00	以字节为单位, 该项只用于透明带标识符转换。
54	帧头	0x00~0xFF	0x40	一个字节, 用户自定义
55	帧尾	0x00~0xFF	0x1A	一个字节, 用户自定义
56	反馈触发帧数	0x04~0x0B	0x07	以一个 CAN 帧为一个单位, 该项只用于 SPI 转 CAN 模式下。

序号	参数	数据范围	默认值	说明
57	反馈触发时间	0x01~0xFF	0x05	以 100ms 为一个时间单位，该项只用于 SPI 转 CAN 模式下。
58	CPOL	0x00	0x00	固定
59	CPHA	0x00	0x00	固定

写配置命令帧数据域定义表

序号	代码	串口波特率 (bps)	CAN 波特率 (bps)
1	0x01	—	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	—	250k
11	0x0B	—	400k
12	0x0C	128000	500k
13	0x0D	—	666k
14	0x0E	—	800k
15	0x0F	—	1M
16	0x10	—	—

UART 和 CAN 波特率代码表

写配置命令后回应帧格式

模块接收到写配置参数命令帧后，会更新当前配置，无论是否更新成功，都会发送回应帧应答。写配置回应帧的格式如下表所示。

帧起始	命令码	状态码	校验码
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

帧起始：2 字节，依次为 0xF7、0xF8。

命令码：1 字节，固定为 0x01。

状态码：0x13 表示命令配置成功，0x07 表示命令执行失败。

校验码：1 字节，为前面所有字节的异或。

获取硬件信息帧格式

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下表所示。

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

帧起始：2 字节，依次为 0xF7、0xF8。

命令码：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。

校验码：1 字节，为前面所有字节的异或，即 0x07。

硬件信息回应帧格式

帧起始	命令码	状态码	校验码
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

帧起始：2 字节，依次为 0xF7、0xF8。

命令码：1 字节，固定为 0x02。

状态代码：0x13 表示硬件匹配成功，0x07 表示硬件匹配失败。

校验码：1 字节，为前面所有字节的异或。

读配置命令格式

读配置参数命令帧，可获取模块当前配置参数。其命令帧的格式如下表所示。

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

帧起始：2 字节，依次为 0xF7、0xF8。

命令码：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。

校验码：1 字节，为前面所有字节的异或，即 0x0C。

读配置命令后回应帧格式

模块在收到读配置参数命令帧后，会通过回应帧将配置参数发送到主控制端。其回应帧的格式如下表所示。

帧起始	命令码	数据长度	数据域	校验码
2 字节	1 字节	1 字节	60 字节	1 字节
0xF7, 0xF8	0x03	0x3C	参考写配置命令帧定义	前面所有字节的异或

帧起始：2 字节，依次为 0xF7、0xF8。

命令码：1 字节，固定为 0x03。

数据长度：1 字节，固定为 0x3C。

数据域：60 字节，参考写配置命令帧定义。

校验码：1 字节，为前面所有字节的异或。

配置步骤

通过 SPI 配置

1. MODE 置高，选择 SPI 模式，CFG 置低，进入配置状态，之后 RST 拉低至少 100 μ s 复位模块，拉高 RST，再等待至少 55ms 模块进入配置模式。
2. CTL0、CTL1 置高，模块进入写命令状态，等待 50 μ s。
3. SSEL 置低，SPI 主机发送配置命令帧，发送完成后拉高 SSEL，检测 INT 引脚，如果为低，将 CTL0 拉高、CTL1 拉低，模块进入读状态，至少等待 50 μ s，再将 SSEL 置低主机开始发送 5 字节无效数据开始读取回应帧，读取完拉高 SSEL。
4. 主机接收回应帧判断配置是否成功。
5. 成功发送配置命令后，CFG 拉高，使模块进入 SPI 转 CAN 数据传输模式，RST 拉低至少等待 100 μ s，再拉高 RST，复位模块，至少等待 55ms，此时发送的配置命令才真正写入模块中。此时可以开始数据的转换传输，CTL0 置高、CTL1 置高，主机通过 SPI 写数据给模块，CTL0 置高、CTL1 置低，主机通过 SPI 读取模块数据。
6. 1、2、3、4 步骤同样使用于读配置命令、获取硬件信息命令。

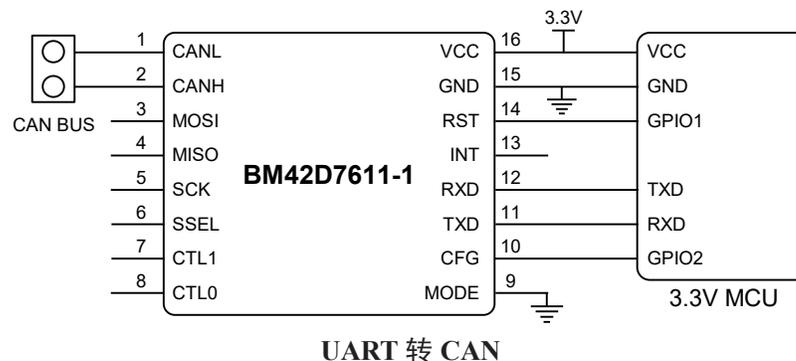
通过 UART 配置

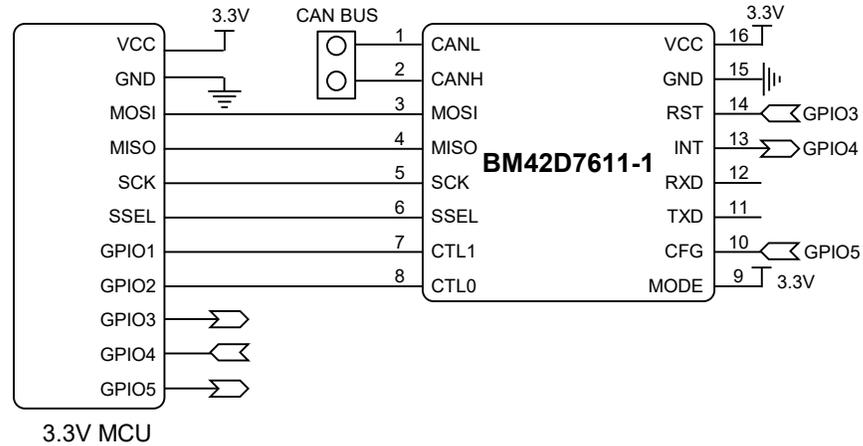
配置时，也就是当 CFG 为低时，串口波特率固定为 9600bps，这是用户需要注意的地方。

1. MODE 置低进入 UART 模式，CFG 置低，进入配置状态，RST 置低，至少等待 100 μ s 后，拉高 RST，再等待 55ms 等待模块复位完成。
2. 此时，主机可以发送配置命令帧，模块在收到配置命令帧后主动发送回应帧给主机。
3. 主机接收回应帧判断配置是否成功。
4. 配置成功后，CFG 置高，RST 置低，至少等待 100 μ s 后，拉高 RST，再等待 55ms 等待模块复位完成，将配置信息写进相应配置端口。此时可以开始数据的转换传输。

应用电路

硬件电路

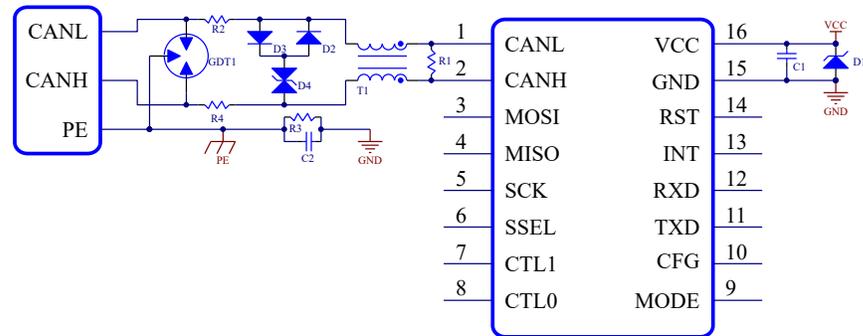




SPI 转 CAN

端口保护电路

当模块用在比较恶劣的环境下 (高压电力、雷击环境)，那么建议用户在模块的 CANH/CANL 端增加保护电路，防止模块工作在恶劣环境下损坏。下图提供一个针对大量雷击浪涌的土建防护电路设计方案，并在参数说明中列出一组推荐电路参数，用户可根据实际情况进行调整。

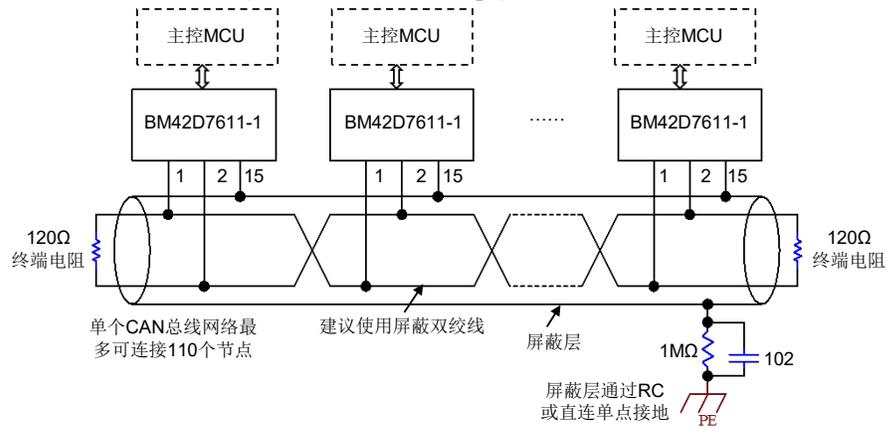


参数说明:

元件	推荐参数
C1	10 μ F, 35V
D1	SMBJ5.0A
R1	120 Ω
R3	1M Ω
C2	1nF, 2kV
R2、R4	2.7 Ω /2W
D2、D3	1N4007
D4	SMBJ30CA
GDT1	B3D090L
T1	ACM2520-301-2P

推荐组网方式

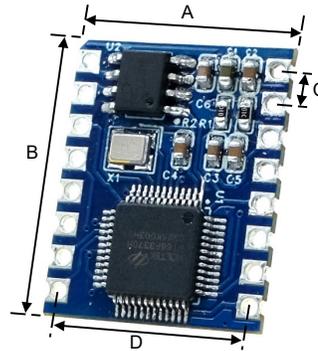
CAN 总线通常采用直线型布线方式，每个网络可接入节点数为 110，通用模块最长通信距离为 10km (CAN 波特率为 5kbps)。



Layout 说明

CANL、CANH 应采用差分信号走线，提高抗干扰性能。

尺寸图



编号	单位	mm	inch
A (产品长度)		17.5	0.7
B (产品宽度)		18.9	0.74
C (引脚间距)		2.54	0.1
D (引脚间距)		15.24	0.6

注：尺寸单位：mm(inch)，公差：±0.25(±0.010)。

参考信息

修订历史

日期	作者	发行	修订说明
20241127	王靖钧	V1.0	第一版

在线购买

[倍创科技](#)

Copyright® 2024 by BEST MODULES CORP. All Rights Reserved.

本文件出版时倍创已针对所载信息为合理注意，但不保证信息准确无误。文中提到的信息仅是提供作为参考，且可能被更新取代。倍创不承担任何明示、默示或法定的，包括但不限于适合商品化、令人满意的质量、规格、特性、功能与特定用途、不侵害第三方权利等保证责任。倍创就文中提到的信息及该信息之应用，不承担任何法律责任。此外，倍创并不推荐将倍创的产品使用在会由于故障或其他原因而可能会对人身安全造成危害的地方。倍创特此声明，不授权将产品使用于救生、维生或安全关键零部件。在救生 / 维生或安全应用中使用倍创产品的风险完全由买方承担，如因该等使用导致倍创遭受损害、索赔、诉讼或产生费用，买方同意出面进行辩护、赔偿并使倍创免受损害。倍创 (及其授权方，如适用) 拥有本文件所提供信息 (包括但不限于内容、数据、示例、材料、图形、商标) 的知识产权，且该信息受著作权法和其他知识产权法的保护。倍创在此并未明示或暗示授予任何知识产权。倍创拥有不事先通知而修改本文件所载信息的权利。如欲取得最新的信息，请与我们联系。