

CAN-COM

Can Converter

CAN(FD)转换器使用手册

版本：V2.7

适用型号：

TTL/RS232/RS485/USB-CAN(FD)

更新日期：2025/12/19

版本	修订日期	修订说明	维护人
2.0	2019-04-10	V2 系列转换器发布	JZJ
2.1	2019-10-14	增加包模式下，帧尾可自定义功能	JZJ
2.2	2020-01-08	增加 Modbus 从机模式，支持 10H 和 03H 指令	JZJ
2.3	2020-06-12	升级 V3 版本，更新硬件参数	JZJ
2.4	2021-11-30	重新整理	JZJ
2.5	2023-09-18	串口波特率支持 921600，增加 AT 指令模式 更新 CAN 配置软件	JZJ
2.6	2024-02-20	更新 CAN “配置软件” 为 “调试工具”，支持自动连接设备、CAN 数据解析和筛选、CAN 软件滤波功能	JZJ
2.7	2025-12-19	增加 CAN 自定义波特率设置 支持 CANFD 转换器参数设置和数据收发解析	JZJ

本手册可能会随着产品的更新而进行修改，请以最新版本的手册为准，手册修改不另行通知

珠海市泥人电子科技有限公司保留对本手册所有内容的最终解释权及修改权

目录

1. 概述	1
2. 简介	1
2.1. CAN 接口	1
2.2. CANFD 接口	1
2.3. COM 接口	1
2.4. 功能特性	2
3. 快速调试	3
3.1. 驱动安装	3
3.2. 连接调试工具	4
3.3. 通信测试-CAN	6
3.4. 通信测试-CANFD	8
3.5. 故障和处理办法	10
4. CAN 应用	11
4.1. CAN 速率	11
4.1.1. 常规波特率	11
4.1.2. 自定义波特率	11
4.2. CAN id	14
4.3. CAN 滤波器	14
4.4. CAN 阻抗	16
5. 调试工具	17
5.1. 配置模式	17
5.2. 收发模式	18
5.2.1. 进入收发模式	19
5.2.2. 发送 CAN 数据	20
5.2.3. 接收 CAN 数据	22
5.2.4. 智能过滤器	22
5.2.5. 搜索数据	23
5.2.6. 记录数据	23
6. 数据模式	25
6.1. 透传模式-MODE0	25
6.1.1. 串行帧转 CAN/FD	26
6.1.2. CAN/FD 转串行帧	30
6.2. 带标识符透传模式-MODE1	31
6.2.1. 串行帧转 CAN/FD	32
6.2.2. CAN/FD 转串行帧	35

6.3. 包模式-MODE2	36
6.3.1. 串行帧转 CAN/FD	37
6.3.2. CAN/FD 转串行帧	38
6.4. Modbus 模式-MODE3	39
6.4.1. 串行帧转 CAN/FD	40
6.4.2. CAN/FD 转串行帧	42
6.5. 格式转发模式- MODE4	43
6.5.1. 串行帧转 CAN/FD	43
6.5.2. CAN/FD 转串行帧	45
6.6. Modbus 从机模式- MODE5	46
6.7. AT 指令模式- MODE6	49
7. 固件升级	53

1. 概述

本手册适用于泥人科技 CAN/FD 转换器系列，全系列包含 TTL、RS232、RS485 和 USB 串口转 CAN/FD。串口和 CAN/FD 之间数据支持双向转换，CAN/FD 接口支持硬件过滤器设置，可设置接收需要 ID，串口波特率和 CAN/FD 波特率均可以自定义设置。

转换器参数均通过串口进行配置，参数支持掉电保存。用户可通过本手册快速了解设备各个功能的作用及使用方法。手册详细介绍了参数配置、各种数据转换模式的使用，具体操作请看下文介绍。

2. 简介

2.1. CAN 接口

项目	说明
链路层协议	CAN 2.0A 和 CAN2.0B 协议
CAN-ID 类型	11bit-CAN2.0A(标准帧)，29bit-CAN2.0B(扩展帧)
波特率	3K~1Mbit/s
CAN 帧类型	标准数据帧、标准扩展帧、标准远程帧和扩展远程帧
支持节点数	大于 64 个
终端匹配电阻	120Ω

2.2. CANFD 接口

项目	说明
链路层协议	CAN 2.0A/B 协议、ISO-CANFD、BOSCH-CANFD
CAN-ID 类型	11bit-CAN2.0A(标准帧)，29bit-CAN2.0B(扩展帧)
波特率	仲裁域：3K~1Mbit/s 数据域：100K~5Mbit/s 支持 CANFD 加速(可变速率)
CAN 帧类型	标准数据帧、标准扩展帧、标准远程帧和扩展远程帧
支持节点数	大于 64 个
终端匹配电阻	120Ω

2.3. COM 接口

项目	说明
波特率	2400~921600bps,可配置奇、偶校验和停止位
TTL 串口	支持 3.3V 和 5V 电平
RS232 接口	支持 232 串口,最高波特率仅支持到 512000bps
RS485 接口	内置 120Ω终端电阻，支持大于 128 个节点
USB 接口	USB 转串口，支持 Win7/Win10/Win11/Linux

2.4. 功能特性

项目	说明
CAN	● 过滤器：屏蔽位模式下，支持 14 组过滤器，标识符列表模式下，支持 28 个标识符 ID； ● 离线自恢复； ● 支持正常模式、回环模式、静默模式、回环静默模式； ● 支持 200 帧 CAN 数据缓存
COM 串口	● 支持串口数据包发送时间间隔设置 ● 支持 4K 串口数据缓存
数据模式	● 透传模式 ● 带标识符透传模式 ● 包模式 ● Modbus 模式 ● 格式转发模式 ● Modbus 从机模式 ● AT 指令模式
默认参数-CAN 转换器	● 串口波特率：9600，无校验，8 位数据位，1 位停止位 ● CAN 波特率：500K ● CAN 过滤器：屏蔽位模式，允许所有帧接收 ● 数据模式：透传模式
默认参数-CANFD 转换器	● 串口波特率：9600，无校验，8 位数据位，1 位停止位 ● CAN 波特率：仲裁域 500K，数据域 2M，ISOFD 协议，使能 FD 加速 ● CAN 过滤器：屏蔽位模式，允许所有帧接收 ● 数据模式：透传模式，标准帧，ID=0

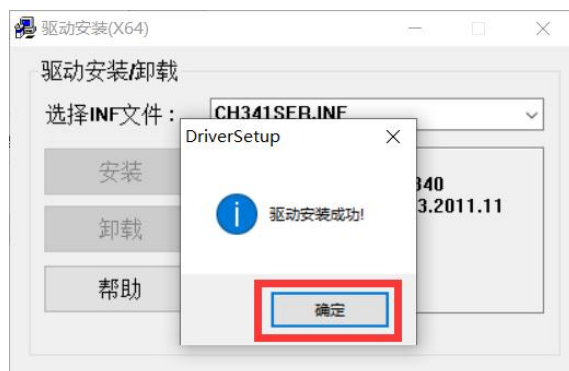
3. 快速调试

3.1. 驱动安装

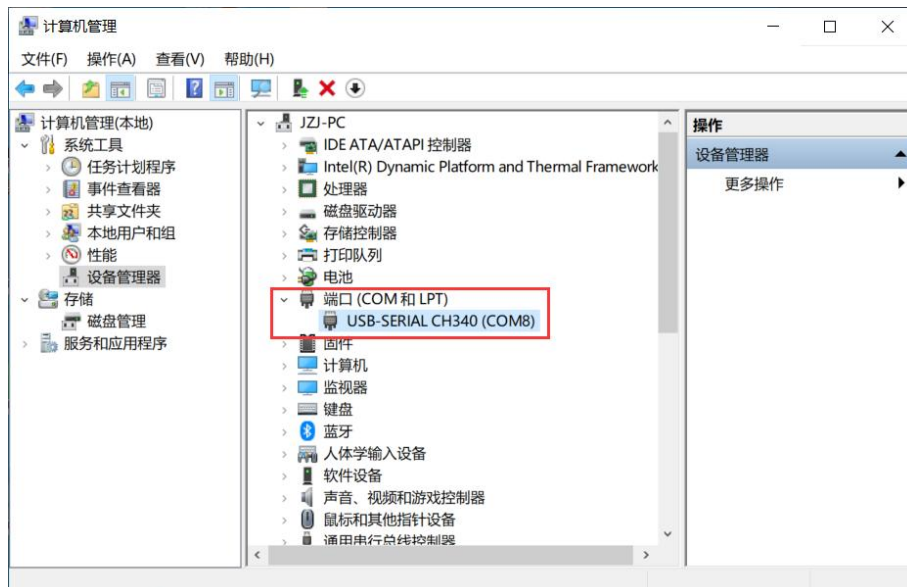
- USB 转 CAN/FD 设备需要安装驱动，其他转换器不需要安装(可跳过该章节)，但需自备串口线连接电脑；
- 下载并解压资料包，打开以下路径：**配套资料\USB 驱动\ ch341ser**；
- 把 USB 转 CAN 设备插入电脑 USB 口；
- 双击“**ch341ser**”启动安装，请使用 Win7 或更高版本的 Windows 操作系统。
- 点击“安装”。



- 等待安装完成。



- 点击“确定”完成安装。如果安装失败，可以先点击“卸载”再安装。
- 打开电脑设备管理器，可以查看到已经安装完成的 COM 口。



3.2. 连接调试工具

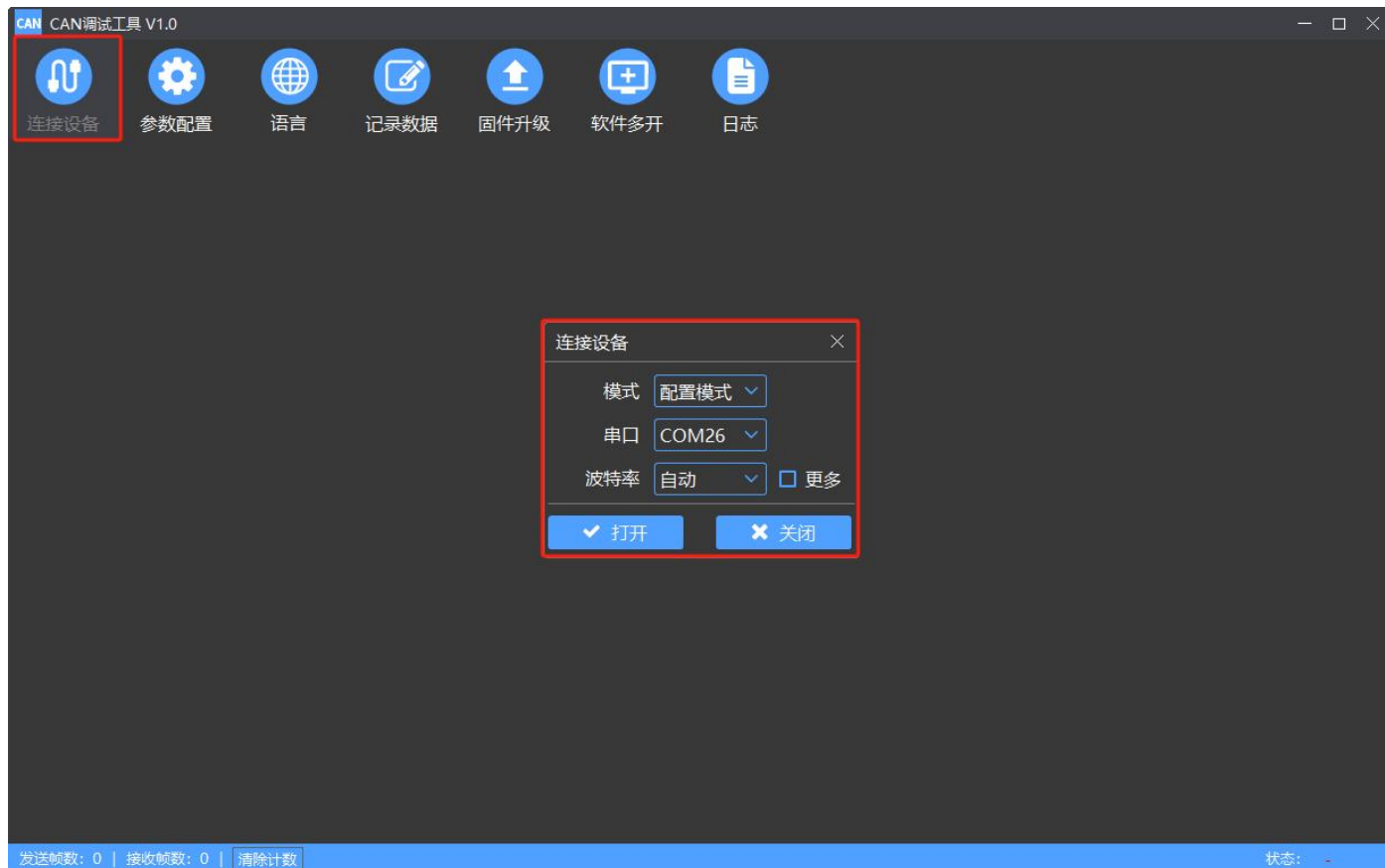
USB-CAN/FD 设备，直接插到电脑 USB 口即可；

TTL-CAN、RS232-CAN、RS485-CAN 需要经过相应的转接线才可以连接到电脑，同时接通电源。具体接口说明请查阅对应型号的产品“规格书”；



➤ 双击 **CAN调试** “调试工具.exe”，打开软件。

➤ 单击“连接设备”，弹出连接设备选项框。



➤ 模式选择“配置模式”，选择连接的串口号，串口波特率选“自动”，点击“打开”。



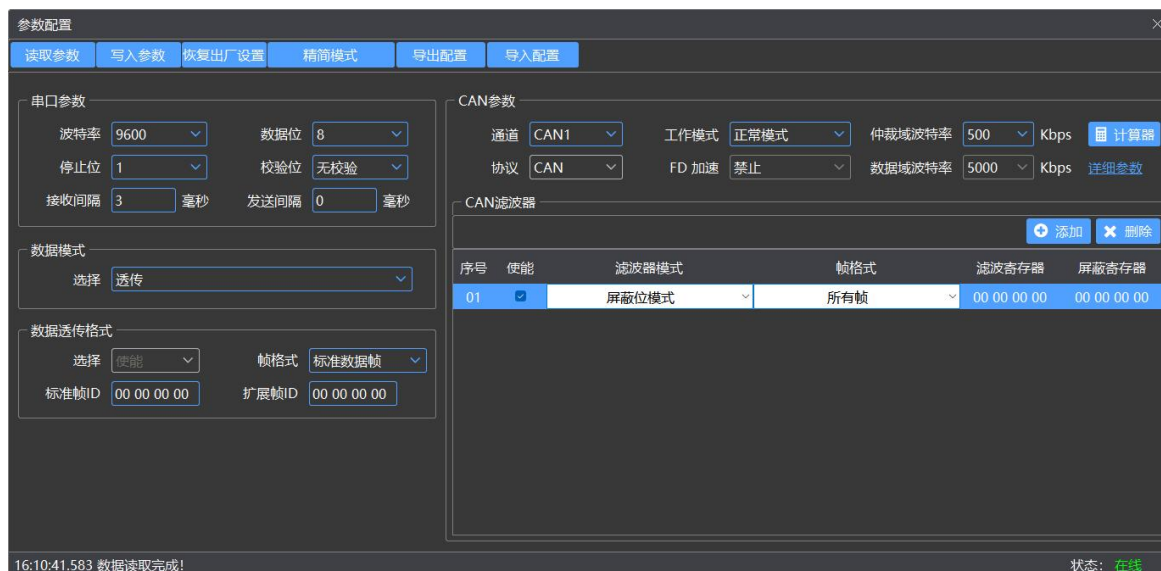
- 进入参数配置页面，右下角连接状态：在线，即调试工具和设备已经链接上，并读取了设备参数。



- “精简模式”的参数配置如图所示，可以点击“专业模式”切换到“专业页面”。



- “专业模式”可以配置转换器的所有参数。

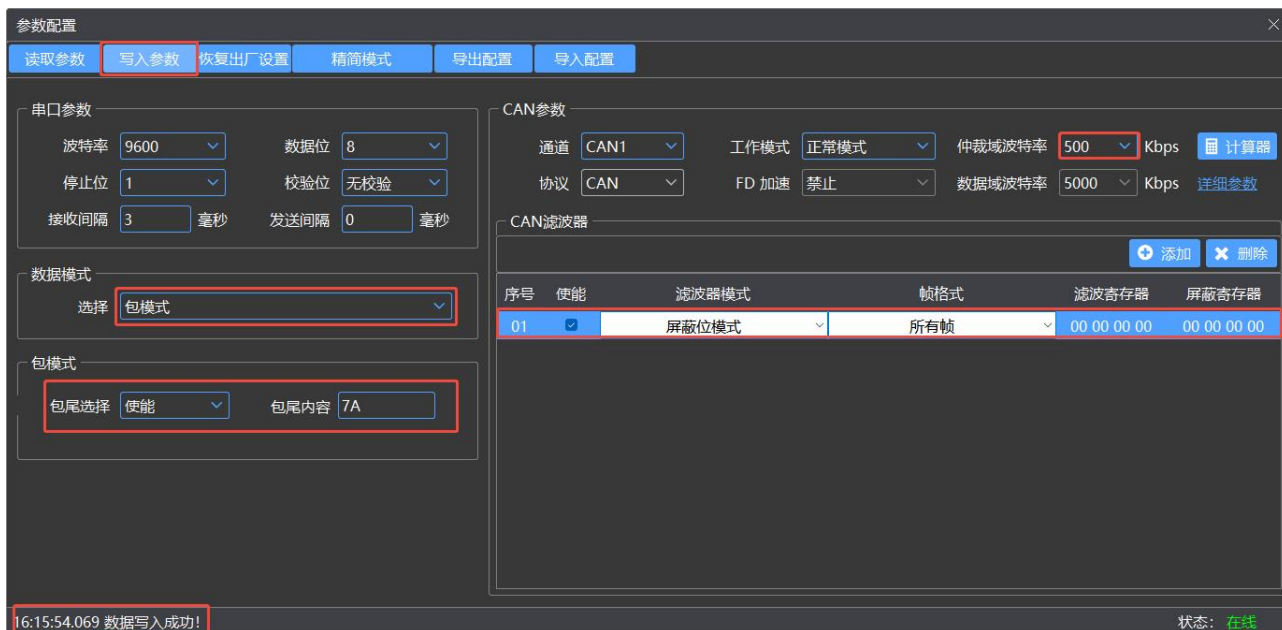


3.3. 通信测试-CAN

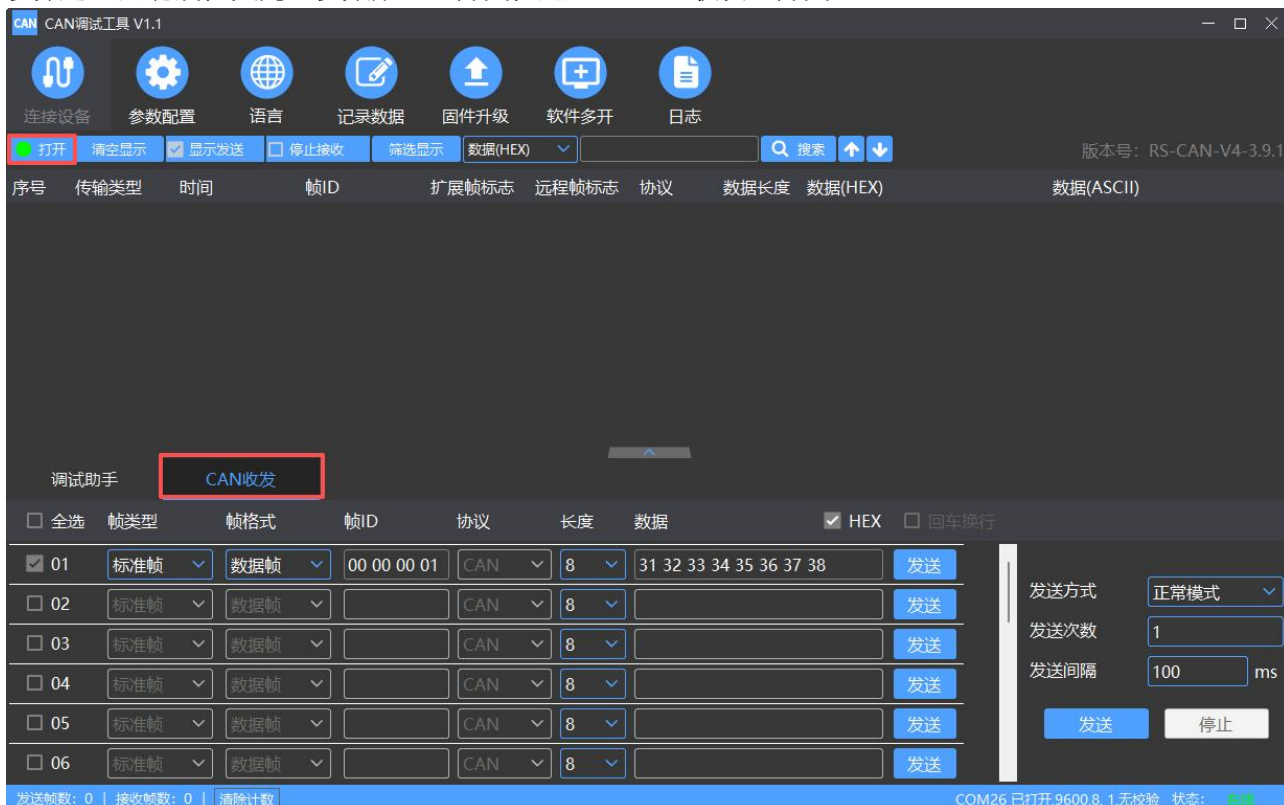
- 了解清楚需要和转换器通信的 CAN 设备的协议, 包含 CAN 速率、CAN-id、帧类型。需要设置相同的速率 CAN 才能互相通信。

例如: CAN 总线的波特率是 “500K”, 转换器接收 CAN 总线的所有数据, 数据模式设置 “包模式”。

- 接通 CAN 总线, 将 CAN 总线的 H 信号接到转换器的 H, 总线的 L 信号接到转换器的 L, 根据需要接入终端电阻。
- 设置 CAN 波特率为 500K, 滤波器使能 “屏蔽位, 所有帧”, 数据模式选择 “包模式”。点击 “写入参数” 保存到设备。



- 参数写入成功后, 关闭 “参数配置” 界面, 进入 “CAN 收发” 界面。



- 接收 CAN 数据，打开设备后，接收到 CAN 总线的数据会在窗口内显示。

The screenshot shows the 'CAN调试工具 V1.1' (CAN Debug Tool V1.1) interface. The top bar includes icons for '连接设备' (Connect Device), '参数配置' (Parameter Configuration), '语言' (Language), '记录数据' (Record Data), '固件升级' (Firmware Upgrade), '软件多开' (Software Multi-instance), and '日志' (Log). Below this is a status bar with buttons for '打开' (Open), '清空显示' (Clear Display), '显示发送' (Show Send), '停止接收' (Stop Receive), '筛选显示' (Filter Display), and a search bar. The version number '版本号: RS-CAN-V4-3.9.1' is displayed on the right.

The main data table shows received CAN data:

序号	传输类型	时间	帧ID	扩展帧标志	远程帧标志	协议	数据长度	数据(HEX)	数据(ASCII)
1	接收	18:37:57.963	00 00 00 01	标准帧	数据帧	CAN	8	31 32 33 34 35 36 37 38	12345678
2	接收	18:38:03.302	00 00 00 02	标准帧	数据帧	CAN	8	31 32 33 34 35 36 37 38	12345678
3	接收	18:38:06.949	00 00 00 03	标准帧	数据帧	CAN	8	31 32 33 34 35 36 37 38	12345678
4	接收	18:38:18.363	00 00 00 03	标准帧	数据帧	CAN	4	31 32 33 34	1234
5	接收	18:38:31.392	00 00 00 05	标准帧	数据帧	CAN	2	31 32	12

Below the table is the '调试助手' (Debug Assistant) section, which includes a 'CAN收发' (CAN Send/Receive) tab. It features a table for configuring CAN frames to be sent, with columns for '帧ID', '协议', '长度', and '数据'. The '发送' (Send) button is highlighted. To the right, there are settings for '发送方式' (Send Method), '发送次数' (Send Count), and '发送间隔' (Send Interval).

- 发送数据到 CAN 总线，选择需要的帧类型和帧格式，填入 ID 和数据，点击“发送”。

The screenshot shows the 'CAN调试工具 V1.1' (CAN Debug Tool V1.1) interface, similar to the previous one, but with the '发送' (Send) button highlighted in the 'CAN收发' (CAN Send/Receive) section. The main data table now includes sent data:

序号	传输类型	时间	帧ID	扩展帧标志	远程帧标志	协议	数据长度	数据(HEX)	数据(ASCII)
1	接收	18:37:57.963	00 00 00 01	标准帧	数据帧	CAN	8	31 32 33 34 35 36 37 38	12345678
2	接收	18:38:03.302	00 00 00 02	标准帧	数据帧	CAN	8	31 32 33 34 35 36 37 38	12345678
3	接收	18:38:06.949	00 00 00 03	标准帧	数据帧	CAN	8	31 32 33 34 35 36 37 38	12345678
4	接收	18:38:18.363	00 00 00 03	标准帧	数据帧	CAN	4	31 32 33 34	1234
5	接收	18:38:31.392	00 00 00 05	标准帧	数据帧	CAN	2	31 32	12
6	发送	18:39:53.906	00 00 00 01	标准帧	数据帧	CAN	8	31 32 33 34 35 36 37 38	12345678
7	发送	18:40:12.036	00 00 12 34	扩展帧	数据帧	CAN	8	31 32 33 34 35 36 37 38	12345678
8	发送	18:40:22.669	00 00 12 34	扩展帧	数据帧	CAN	4	31 32 33 34	1234

The '调试助手' (Debug Assistant) section shows the configuration for the sent frames. The '帧ID' (Frame ID) is set to '00 00 12 34', and the '数据' (Data) field is filled with '31 32 33 34'. The '发送' (Send) button is highlighted.

- CAN 总线通过连接另一个转换器，可以验证接收到 CAN 总线的数据。

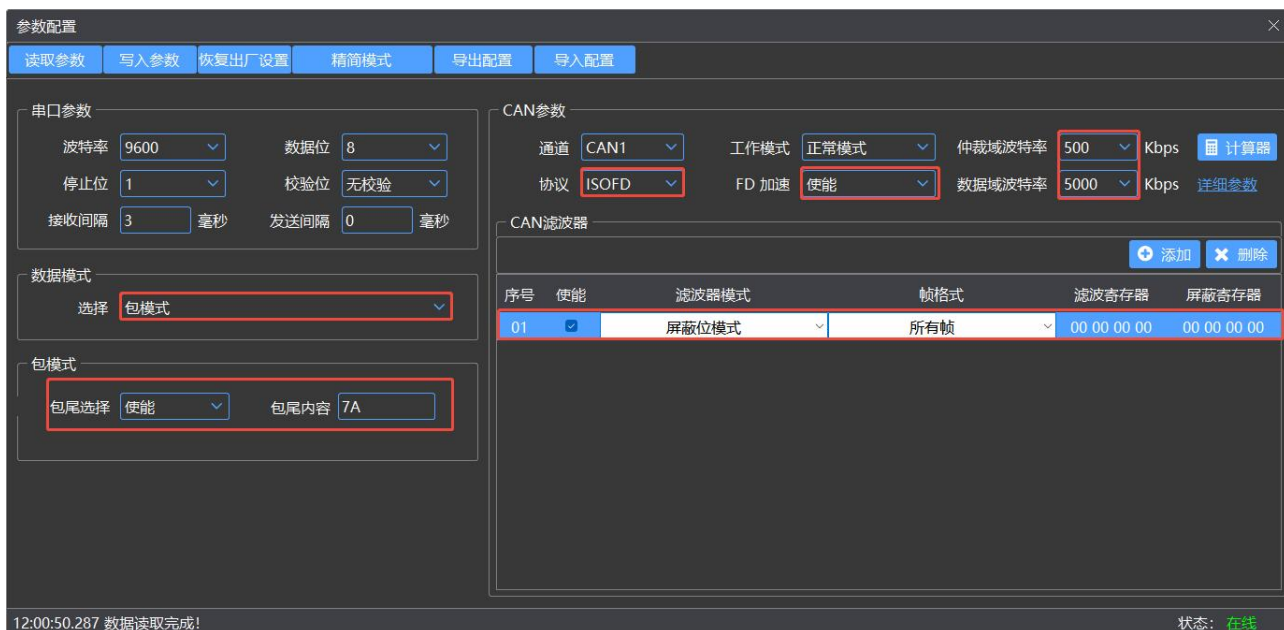
序号	传输类型	时间	帧ID	扩展帧标志	远程帧标志	协议	数据长度	数据(HEX)	数据(ASCII)
1	接收	18:39:54.052	00 00 00 01	标准帧	数据帧	CAN	8	31 32 33 34 35 36 37 38	12345678
2	接收	18:40:12.189	00 00 12 34	扩展帧	数据帧	CAN	8	31 32 33 34 35 36 37 38	12345678
3	接收	18:40:22.815	00 00 12 34	扩展帧	数据帧	CAN	4	31 32 33 34	1234

3.4. 通信测试-CANFD

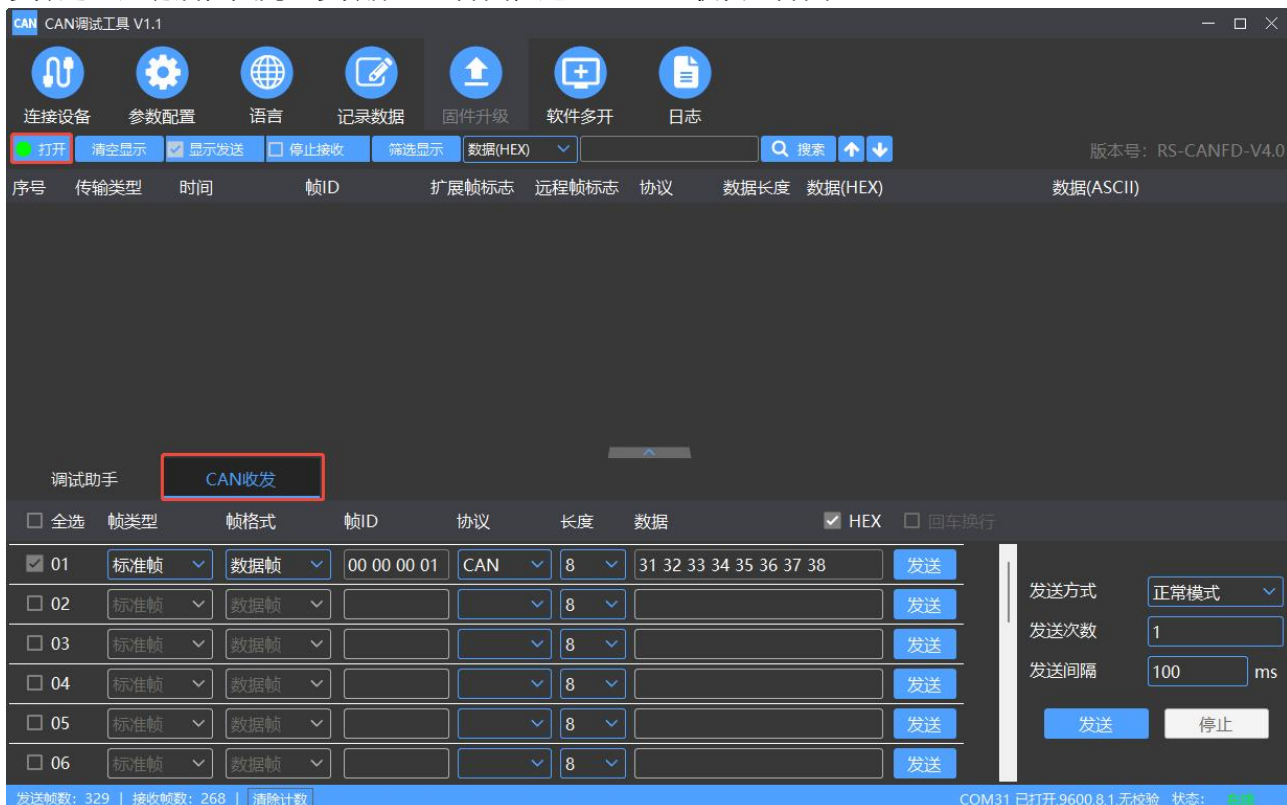
- 了解清楚需要和转换器通信的 CANFD 设备的协议，包含 CAN 仲裁域和数据域的速率、CAN-id、帧类型。需要设置相同的速率 CAN 才能互相通信。

例如：CAN 协议采用 ISO-CANFD，仲裁域速率 500K，数据域速率 2M，转换器接收 CAN 总线的所有数据，数据模式设置“包模式”。

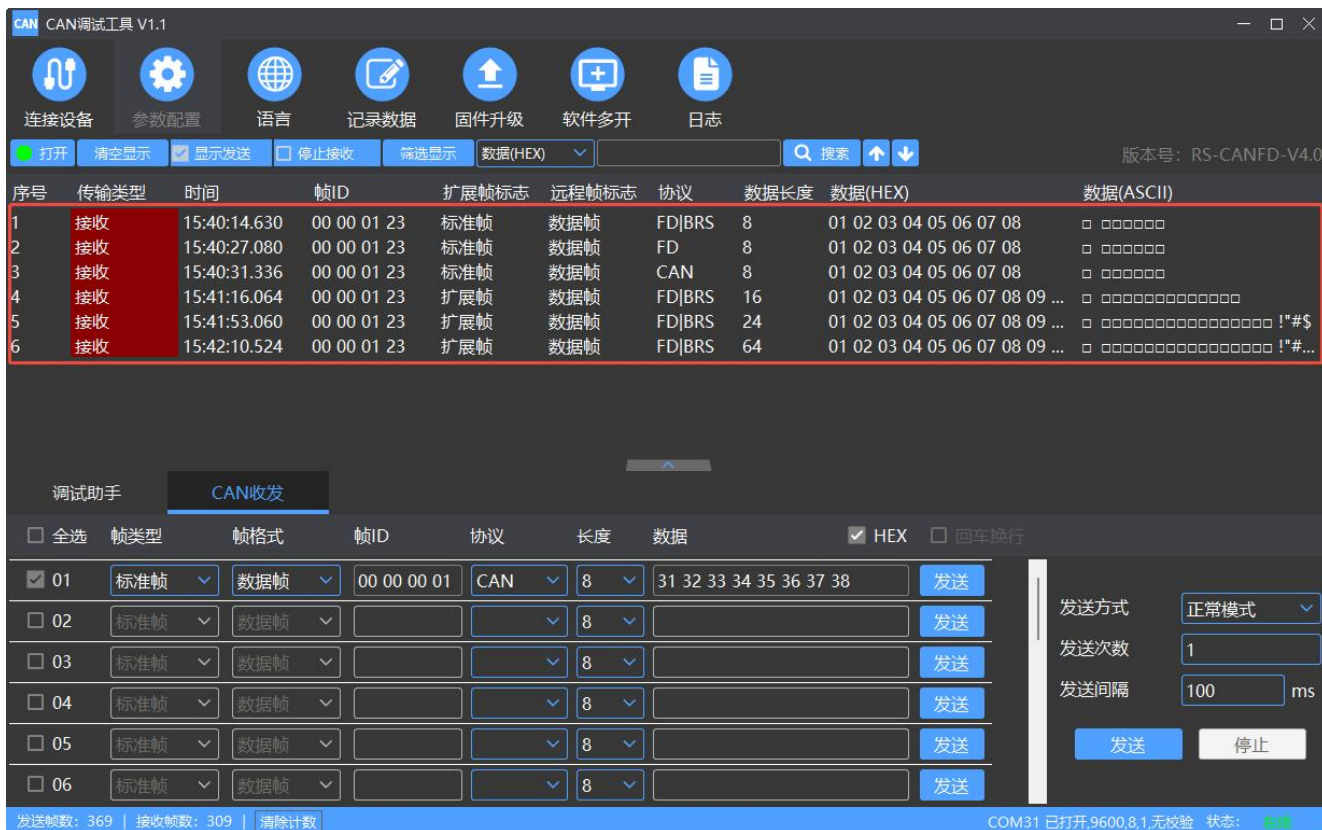
- 接通 CAN 总线，将 CAN 总线的 H 信号接到转换器的 H，总线的 L 信号接到转换器的 L，根据需要接入终端电阻。
- 设置 CAN 协议选择“ISOFD”，仲裁域速率“500K”，使能 FD 加速，数据域速率“2M”，滤波器使能“屏蔽位，所有帧”，数据模式选择“包模式”。点击“写入参数”保存到设备。



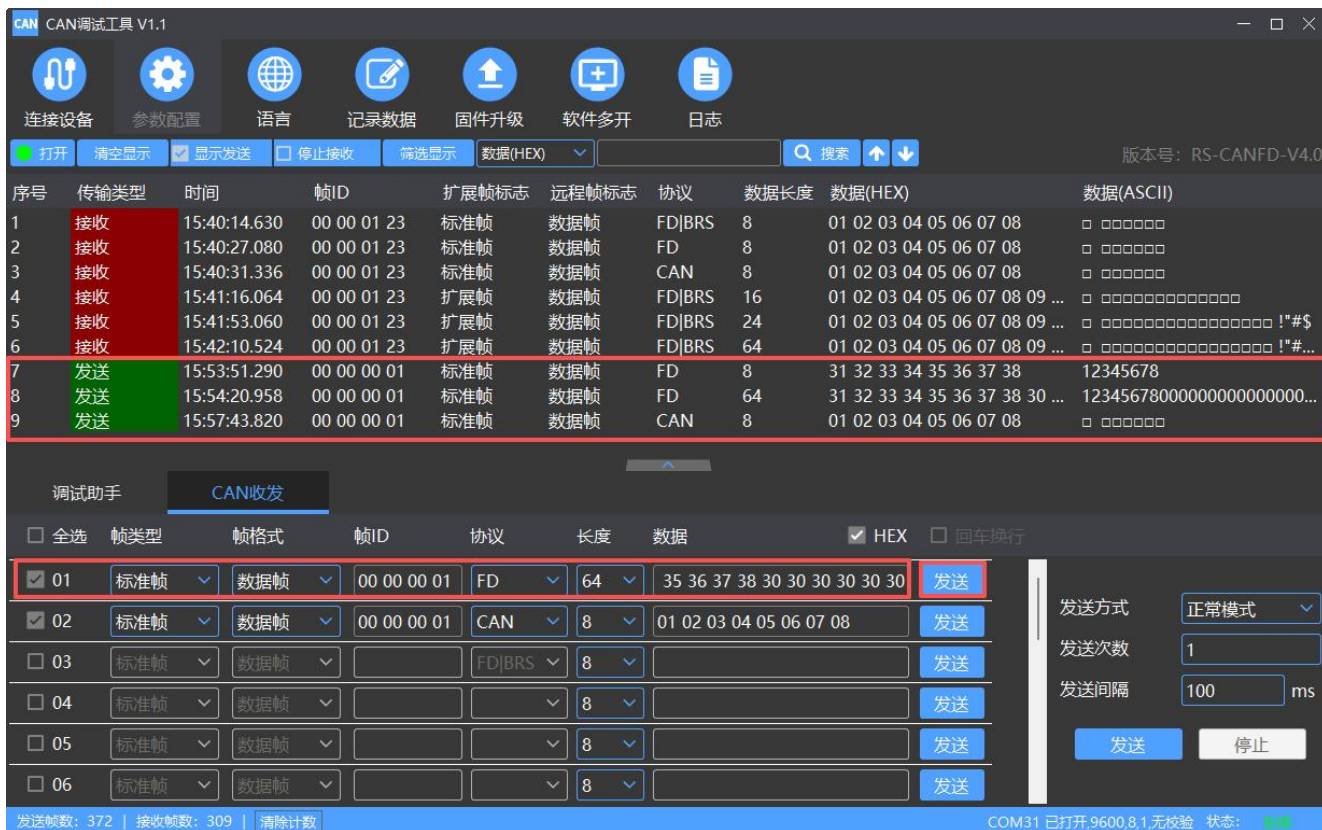
- 参数写入成功后，关闭“参数配置”界面，进入“CAN 收发”界面。



- 接收 CAN 数据，打开设备后，接收到 CAN 总线的数据会在窗口内显示，支持接收 CAN 和 CANFD 的数据。协议中 CAN 代表 CAN2.0 协议，FD 代表 CANFD 不加速(速率不变)，FD|BRS 代表 CANFD 加速(可变速率)。



- 发送数据到 CAN 总线，选择需要的协议、帧类型和帧格式，填入 ID 和数据，点击“发送”。



- CAN 总线通过连接另一个转换器，可以验证接收到 CAN 总线的数据。

序号	传输类型	时间	帧ID	扩展帧标志	远程帧标志	协议	数据长度	数据(HEX)	数据(ASCII)
1	接收	16:01:13.495	00 00 00 01	标准帧	数据帧	FD	8	31 32 33 34 35 36 37 38	12345678
2	接收	16:01:40.111	00 00 00 01	标准帧	数据帧	FD	64	31 32 33 34 35 36 37 38 30 ...	123456780000000000000000...
3	接收	16:02:27.069	00 00 00 01	标准帧	数据帧	CAN	8	01 02 03 04 05 06 07 08	□ □ □ □ □ □ □ □

3.5. 故障和解决办法

◆ 配置模式下，打开串口，无法读取参数或写入参数。

原因 1：串口号选错。

解决办法：确认打开的串口号是连接转换器的，或刷新串口。

原因 2：串口波特率选错。

解决办法：软件会自动判断转换器波特率，正常情况下会连接成功，如果自动连接不上，可以手动选择。默认出厂是 9600，如果修改为其他的波特率，需要选择正确的波特率，再打开串口。如果忘记之前改过的波特率，可以硬件恢复出厂设置，将模式参数恢复成出厂状态，具体办法查阅“硬件参考手册”。

原因 3：硬件接线错误。

解决办法：根据转换器型号，查阅相应的“硬件参考手册”，确认接线。

◆ 写入参数失败

原因 1：存在非法参数。

解决办法：先点“读取参数”将模块所有参数读取出来后，更改成需要的参数，再点“写入参数”。

◆ 收发模式下，打开串口后无数据

原因 1：设备没有进入收发模式。

解决办法：关闭设备后，重新打开设备。

原因 2：CAN 滤波器配置错误。

解决办法：可以先使能滤波器，接收所有帧，先确认是否能收到 CAN 数据。

原因 3：CAN 波特率不对。

解决办法：CAN 总线波特率要一致，才能互相通信，确认正确的波特率。

原因 4：CAN 总线无数据。

解决办法：让其他的 CAN 设备主动发送数据到 CAN 总线上。

原因 5：CAN 总线异常。

解决办法：检查总线是否 CANH 接 CANH，CANL 接 CANL，缩短 CAN 总线长度，检查 CAN 总线阻抗，我们设备出厂默认带了 120Ω 的终端电阻，在一个总线上，只需要首尾两端的设备带终端电阻。如果多个终端电阻，需要去掉，具体办法查阅“硬件参考手册”。

4. CAN 应用

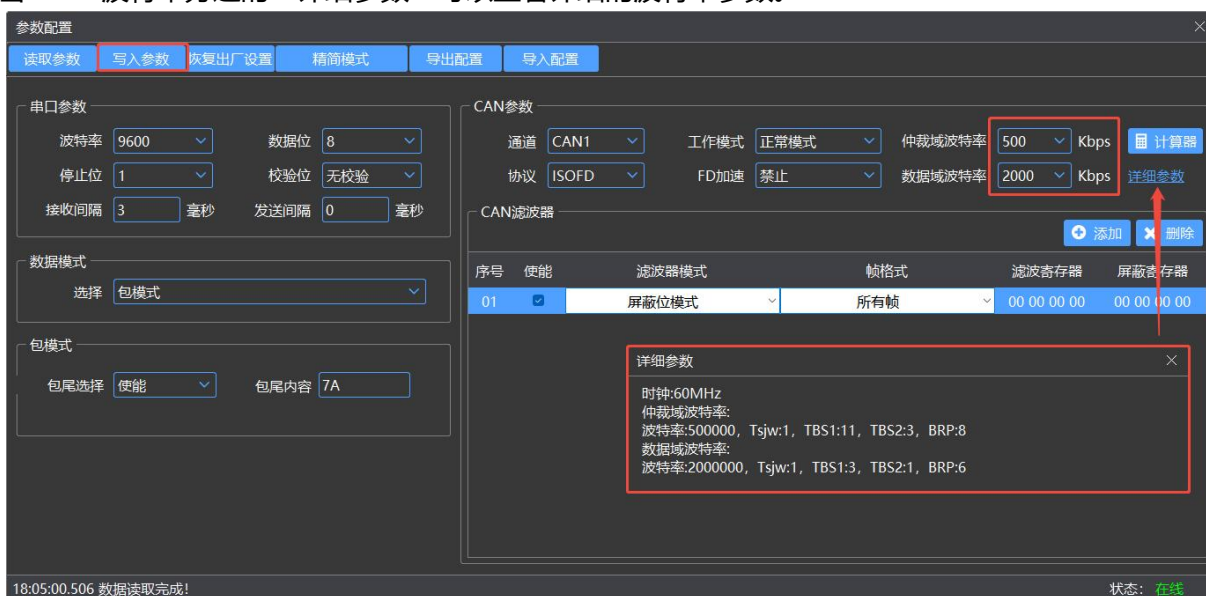
4.1. CAN 速率

CAN 总线上速率一致，设备才可以互相收发数据。当 CANFD 启用加速时，仲裁域和数据域都必须和总线一致。波特率除了设置“常用波特率”外，还可以设定“自定义波特率”。

4.1.1. 常规波特率

- 进入参数配置，设置仲裁域和数据域波特率，CAN 只有仲裁域波特率，CANFD 才需要设置两个速率。根据需要修改后，点击“写入参数”保存设置。

点击 CAN 波特率旁边的“详细参数”可以查看详细的波特率参数。



4.1.2. 自定义波特率

- 自定义波特率，点击波特率右边的“计算器”按钮，进入自定义波特率计算页面。



➤ 输入需要波特率，选择合适的同步宽度，点击“计算”。

计算器

计算

确定

CANFD仲裁域

时钟

60

MHz

总线波特率

500

Kbps

允许误差

0.05

%

同步跳转宽度(sjw)

1

TSEG1	TSEG2	BRP	采样点	波特率	误差
98	21	1	82.50%	500000 bps	0%
97	22	1	81.60%	500000 bps	0%
96	23	1	80.80%	500000 bps	0%
95	24	1	80.00%	500000 bps	0%
94	25	1	79.10%	500000 bps	0%
93	26	1	78.30%	500000 bps	0%
92	27	1	77.50%	500000 bps	0%

CANFD数据域

时钟

60

MHz

总线波特率

2000

Kbps

允许误差

0.05

%

同步跳转宽度(sjw)

1

TSEG1	TSEG2	BRP	采样点	波特率	误差
12	2	2	86.60%	2000000 bps	0%
11	3	2	80.00%	2000000 bps	0%
10	4	2	73.30%	2000000 bps	0%
8	1	3	90.00%	2000000 bps	0%
7	2	3	80.00%	2000000 bps	0%
6	3	3	70.00%	2000000 bps	0%
4	1	5	83.30%	2000000 bps	0%
3	1	6	80.00%	2000000 bps	0%

- 根据需要选中，需要的自定义参数，点击“确定”。

计算器

计算

确定

CANFD仲裁域

时钟

60

MHz

总线波特率

500

Kbps

允许误差

0.05

%

同步跳转宽度(sjw)

1

TSEG1	TSEG2	BRP	采样点	波特率	误差
98	21	1	82.50%	500000 bps	0%
97	22	1	81.60%	500000 bps	0%
96	23	1	80.80%	500000 bps	0%
95	24	1	80.00%	500000 bps	0%
94	25	1	79.10%	500000 bps	0%
93	26	1	78.30%	500000 bps	0%
92	27	1	77.50%	500000 bps	0%

CANFD数据域

时钟

60

MHz

总线波特率

2000

Kbps

允许误差

0.05

%

同步跳转宽度(sjw)

1

TSEG1	TSEG2	BRP	采样点	波特率	误差
12	2	2	86.60%	2000000 bps	0%
11	3	2	80.00%	2000000 bps	0%
10	4	2	73.30%	2000000 bps	0%
8	1	3	90.00%	2000000 bps	0%
7	2	3	80.00%	2000000 bps	0%
6	3	3	70.00%	2000000 bps	0%
4	1	5	83.30%	2000000 bps	0%
3	1	6	80.00%	2000000 bps	0%

- CAN 的波特率参数即为自定义参数，其它参数确认好，点击“写入参数”完成设置。

CAN参数

通道

CAN1

工作模式

正常模式

仲裁域波特率

500

Kbps

数据域波特率

2000

Kbps

计算

详细参数

协议

ISOFD

FD 加速

禁止

4.2. CAN id

➤ Id 分为标准帧 id 和扩展帧 id

项目	标准帧 id	扩展帧 id
链路层协议	CAN 2.0A	CAN2.0B
CAN-ID 类型	11bit	29bit-
取值范围	0~0x7FF	0~0x1FFFFFFF
CAN 帧类型	标准数据帧、标准远程帧	标准扩展帧、扩展远程帧

➤ 配置软件 ID 设置方式，高位补 0

标准帧 id=3, 输入框填入 00 00 00 03

标准帧 id=0x123,输入框填入 00 00 01 23

扩展帧 id=1, 输入框填入 00 00 00 01

扩展帧 id=0x1234567,输入框填入 01 23 45 67

☐ 全选	扩展帧	远程帧	帧ID	数据	☐ HEX	☐ 回车换行
☒ 01	标准帧	数据帧	00 00 00 03	12345678		发送
☒ 02	标准帧	数据帧	00 00 01 23	12345678		发送
☒ 03	扩展帧	数据帧	00 00 00 01	12345678		发送
☒ 04	扩展帧	数据帧	01 23 45 67	12345678		发送

4.3. CAN 滤波器

滤波器是用来设置 CAN 接收总线的哪些 id 的数据，通过配置滤波器，可以选择只接收自己需要的数据。

➤ 接收所有 CAN 数据

使能任意一组滤波器，选择“屏蔽位模式”、“所有帧”，“滤波寄存器”和“屏蔽寄存器”都填 00。

序号	使能	滤波器模式	帧格式	滤波寄存器	屏蔽寄存器
01	☒	屏蔽位模式	所有帧	00 00 00 00	00 00 00 00

➤ 接收指定一个 id 的标准数据帧，id=1。

使能其中一组滤波器，选择“屏蔽位模式”、“标准数据帧”、“滤波寄存器”填入需要的 id，“屏蔽寄存器”填入“7FF”。

序号	使能	滤波器模式	帧格式	滤波寄存器	屏蔽寄存器
01	☒	屏蔽位模式	标准数据帧	00 00 00 01	00 00 07 FF

➤ 接收指定一个 id 的扩展数据帧，id=0x12345。

使能其中一组滤波器，选择“屏蔽位模式”、“扩展数据帧”、“滤波寄存器”填入需要的 id，“屏蔽寄存器”填入“1FFFFFFF”。

序号	使能	滤波器模式	帧格式	滤波寄存器	屏蔽寄存器
01	☒	屏蔽位模式	扩展数据帧	00 01 23 45	1F FF FF FF

➤ 接收多个不同类型的 id,列举了 3 种方式实现。

A. 可以使用上面的办法使能多个滤波器,这样就可以使能最多 14 个不同的 id。

序号	使能	滤波器模式	帧格式	滤波寄存器	屏蔽寄存器
01	☒	屏蔽位模式	扩展数据帧	00 01 23 45	1F FF FF FF
02	☒	屏蔽位模式	标准数据帧	00 00 00 01	00 00 07 FF
03	☒	屏蔽位模式	标准数据帧	00 00 00 02	00 00 07 FF
04	☒	屏蔽位模式	标准数据帧	00 00 00 03	00 00 07 FF
05	☒	屏蔽位模式	扩展数据帧	00 11 22 33	1F FF FF FF
06	☒	屏蔽位模式	扩展数据帧	00 11 22 44	1F FF FF FF

B. 可以使用列表模式，最多可以接收 28 个 id，列表模式下，滤波寄存器和屏蔽寄存器都代表需要接收的 id 值。

序号	使能	滤波器模式	帧格式	滤波寄存器	屏蔽寄存器
01	☒	列表模式	标准数据帧	00 00 00 01	00 00 00 02
02	☒	列表模式	标准数据帧	00 00 00 03	00 00 00 04
03	☒	列表模式	标准数据帧	00 00 00 05	00 00 00 06
04	☒	列表模式	扩展数据帧	00 11 22 33	00 44 55 66
05	☒	列表模式	扩展数据帧	00 11 11 11	00 22 22 22

C. 依然采用屏蔽位模式，根据需要计算屏蔽寄存器的值，实现对个别位忽略判断。

比如标准帧，滤波寄存器：00 00 01 00，屏蔽寄存器：00 00 07 F0，那么就可以接收标准帧 id 0x100~0x10F 的数据。

序号	使能	滤波器模式	帧格式	滤波寄存器	屏蔽寄存器
01	☒	屏蔽位模式	标准数据帧	00 00 01 00	00 00 07 F0

屏蔽位计算方式：屏蔽寄存器 0 的位，就是忽略对滤波寄存器判断，1 的位，就必须和滤波寄存器一致。

项目	十六进制	二进制											
滤波寄存器	0x100	0	0	1	0	0	0	0	0	0	0	0	0
屏蔽寄存器	0x7F0	1	1	1	1	1	1	1	0	0	0	0	0
允许接收的 id	0x100~0x10F	0	0	1	0	0	0	0	X	X	X	X	X

4.4. CAN 阻抗

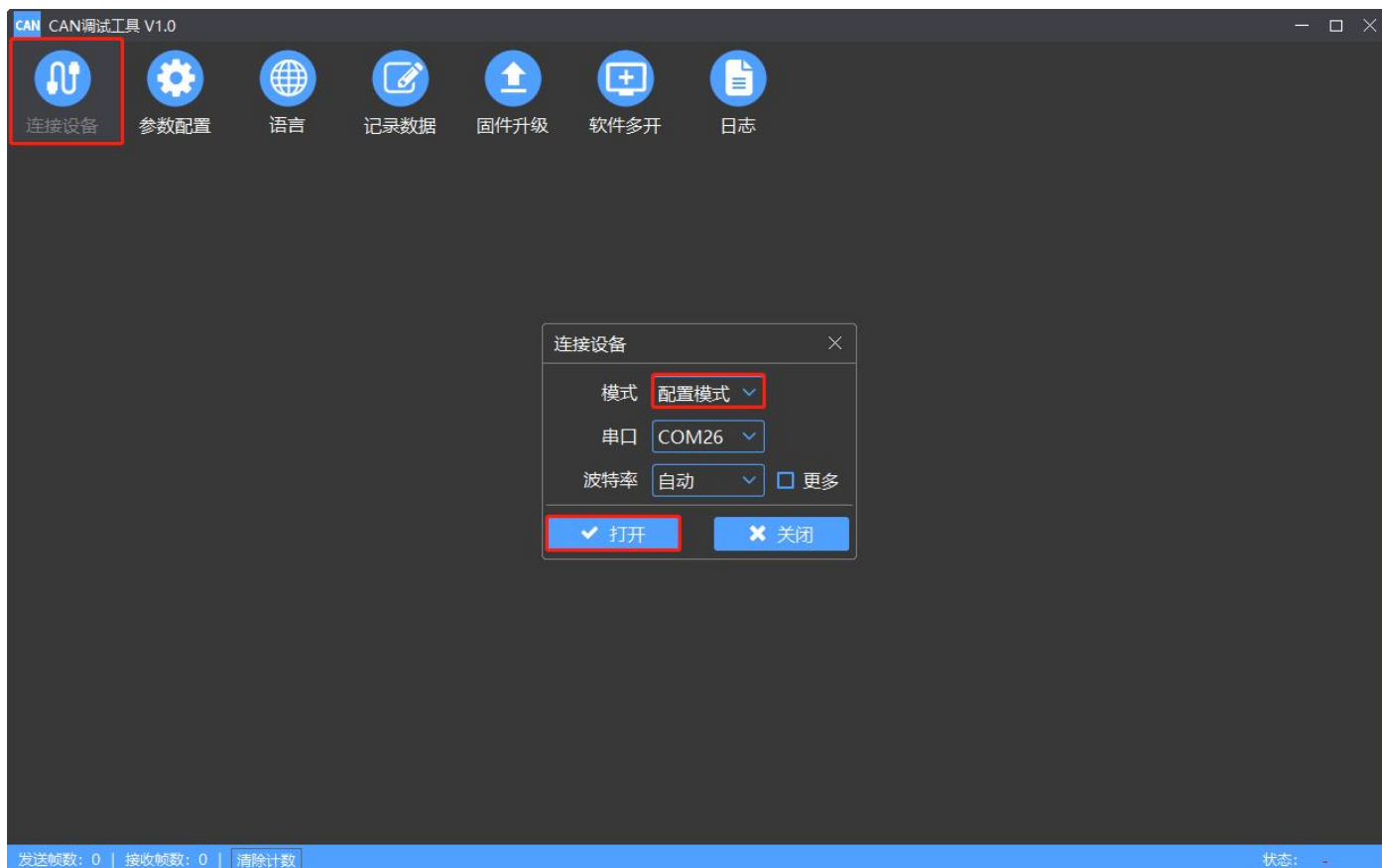
我们的转换器出厂默认带了 120Ω 的终端电阻，多节点通讯的时候，需要去除转换器多余的终端电阻，只保留首尾两端的电阻。如果 CAN 总线上其他设备也有终端电阻，也需要考虑。

5. 调试工具

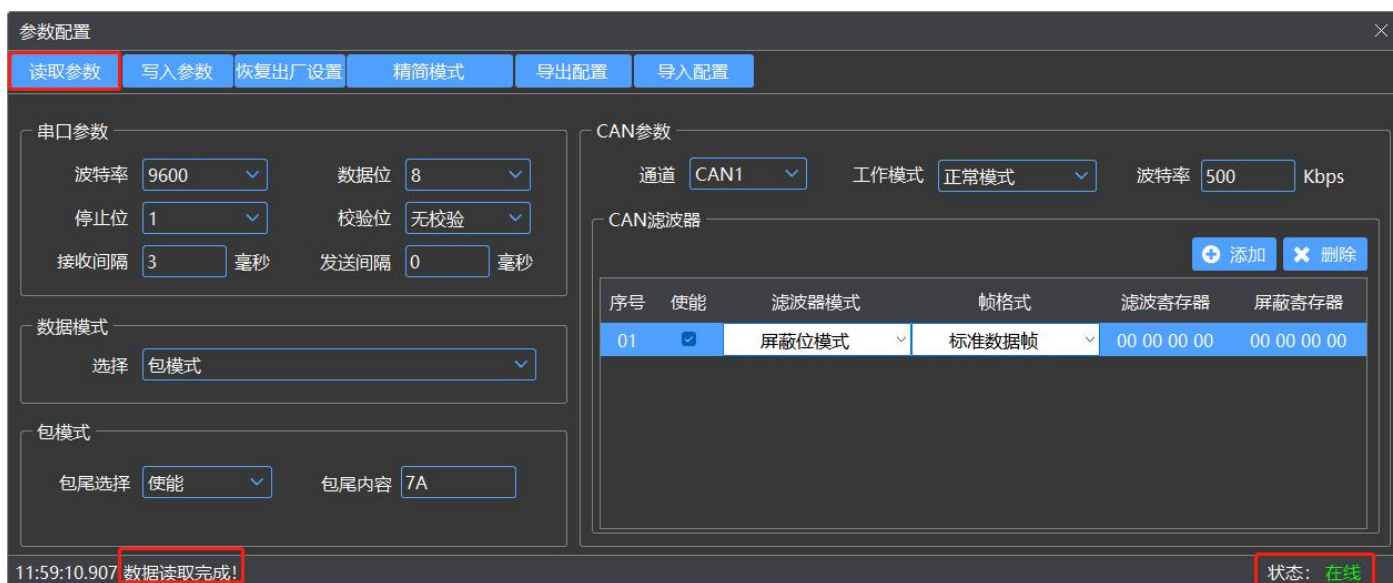
5.1. 配置模式

配置模式可以配置转换器的参数，支持一键读取、一键写入、支持参数导入和导出。

➤ 点击“连接设备”，选择“配置模式”，点击“打开”。



➤ 进入“参数配置”页面，软件会自动读取设备参数，点击“读取参数”也可读取设备参数。



➤ 根据实际需要，修改参数后，点击“写入参数”保存到设备。

参数配置

读取参数 写入参数 恢复出厂设置 精简模式 导出配置 导入配置

串口参数

波特率 9600 数据位 8

停止位 1 校验位 无校验

接收间隔 3 毫秒 发送间隔 0 毫秒

数据模式

选择 包模式

包模式

包尾选择 使能 包尾内容 7A

CAN参数

通道 CAN1 工作模式 正常模式 波特率 500 Kbps

CAN滤波器

+ 添加 - 删除

序号	使能	滤波器模式	帧格式	滤波寄存器	屏蔽寄存器
01	☒	屏蔽位模式	标准数据帧	00 00 00 00	00 00 00 00

12:05:38.407 数据写入成功!

状态: 在线

➤ 支持导入和导出配置参数，导入参数后，需要点“写入参数”才会保存到设备。

参数配置

读取参数 写入参数 恢复出厂设置 精简模式 导出配置 导入配置

串口参数

波特率 9600 数据位 8

停止位 1 校验位 无校验

接收间隔 3 毫秒 发送间隔 0 毫秒

数据模式

选择 包模式

包模式

包尾选择 使能 包尾内容 7A

CAN参数

通道 CAN1 工作模式 正常模式 波特率 500 Kbps

CAN滤波器

+ 添加 - 删除

序号	使能	滤波器模式	帧格式	滤波寄存器	屏蔽寄存器
01	☒	屏蔽位模式	标准数据帧	00 00 00 00	00 00 00 00
02	☒	屏蔽位模式	标准数据帧	00 00 00 01	00 00 00 00
03	☒	屏蔽位模式	标准数据帧	00 00 00 02	00 00 00 00

12:05:38.407 数据写入成功!

状态: 在线

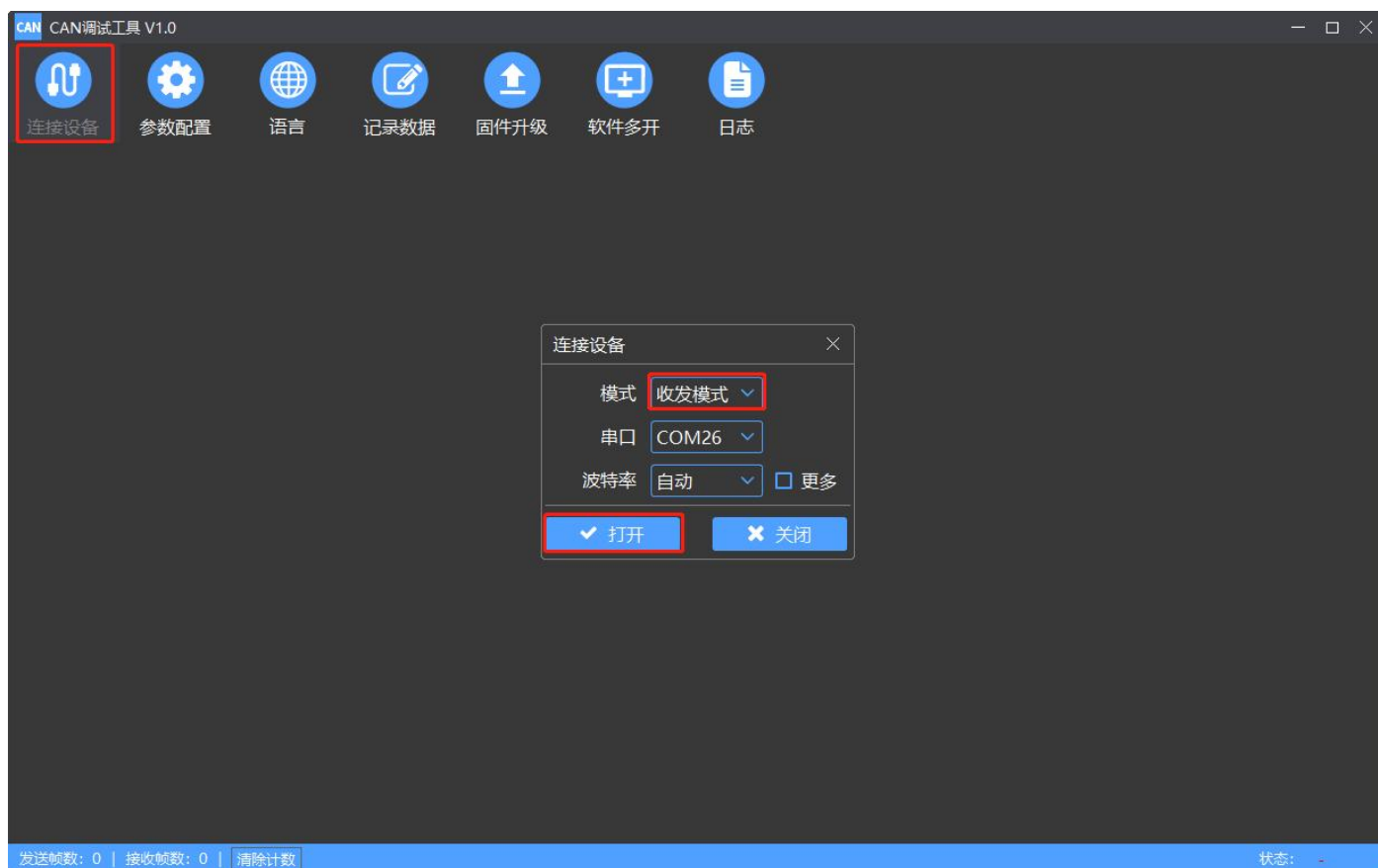
5.2. 收发模式

收发模式下，可以实现 CAN 数据的发送和接收，支持标准帧、扩展帧、数据帧和远程帧的收发。可以对接收的数据进行筛选和过滤，支持软件智能滤波实现对特定 ID 的显示或屏蔽。发送数据支持多帧循环间隔发送。具备完善的数据分析和调试功能。

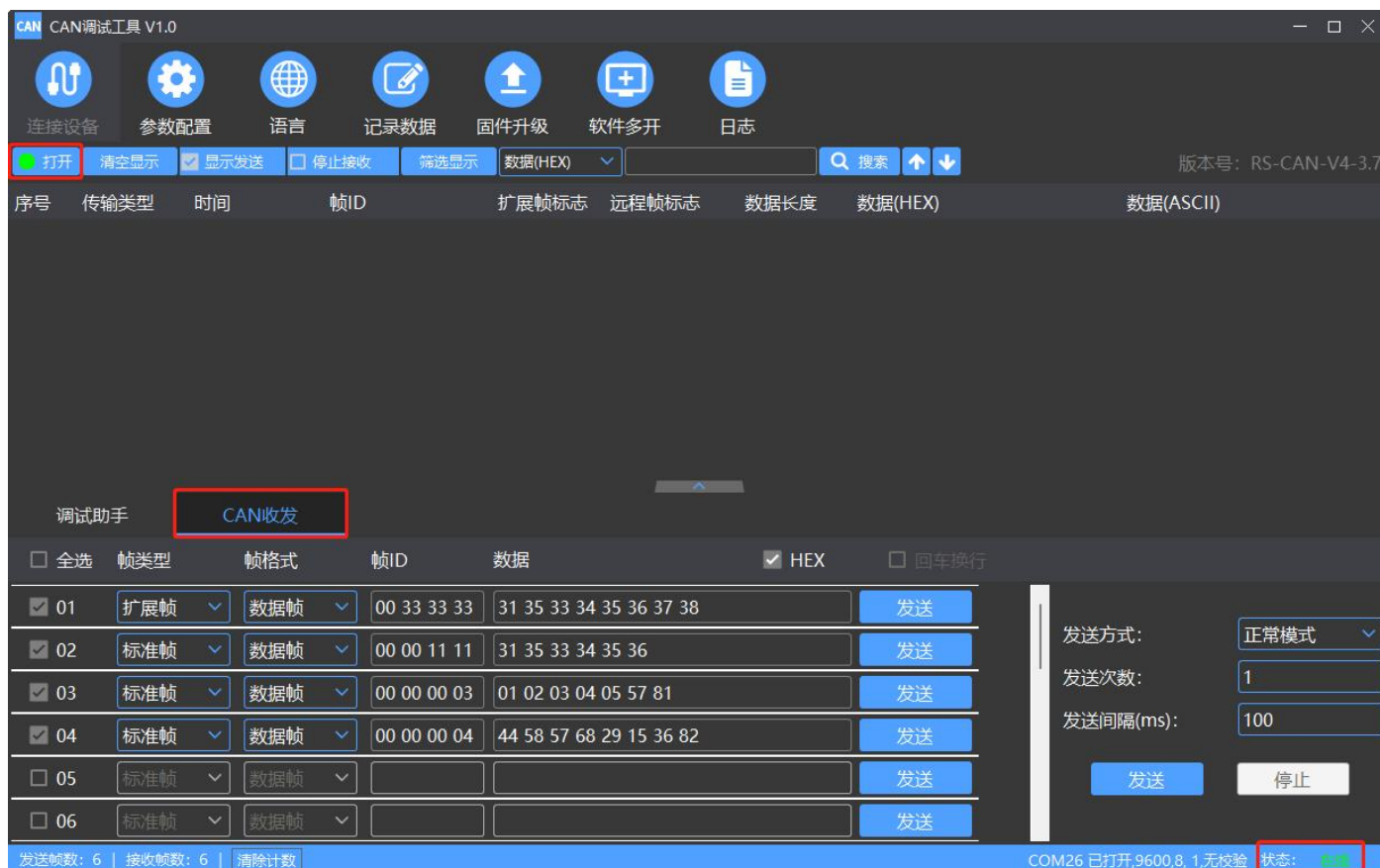
注：收发模式会强制设置转换器进入包模式、并使能包尾。

5.2.1. 进入收发模式

➤ 点击“连接设备”，模式选择“收发模式”，点击“打开”。

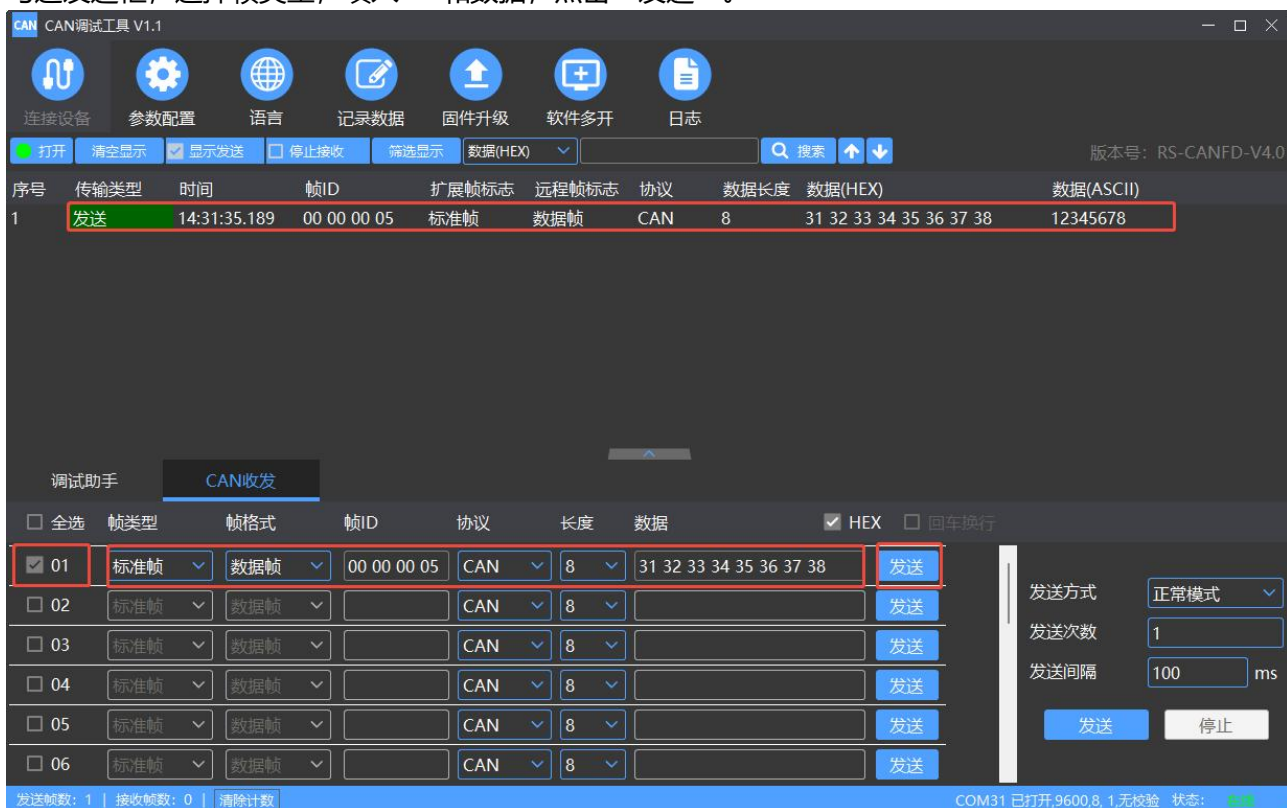


➤ 进入收发模式界面。如果在配置界面进入收发界面，可以点击左上角的“开关”，关闭后再重新打开会强制进入收发模式。



5.2.2. 发送 CAN 数据

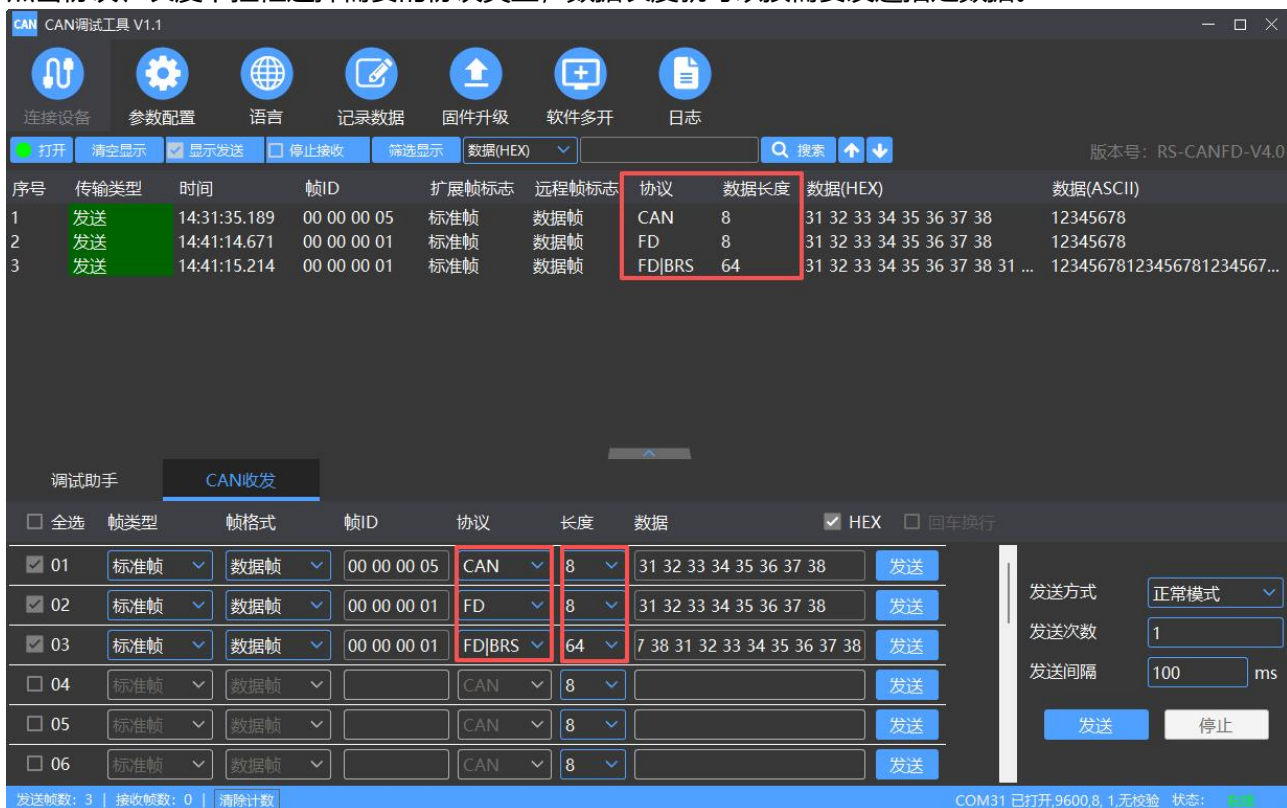
- 勾选发送框，选择帧类型，填入 ID 和数据，点击“发送”。



- CAN 转换器协议仅支持 CAN，数据长度支持 0~8。

CANFD 转换器，协议支持 CAN、CANFD(FD)和 CANFD 加速(FD|BRS)，数据长度支持 0~64。

点击协议、长度下拉框选择需要的协议类型，数据长度就可以按需要发送指定数据。



- 支持多帧数据顺序发送和循环发送，勾选需要发送的发送框，点击最右边“发送”。

例如：顺序发送数据帧 1、2、3，每帧间隔 100ms，连续发 2 次。

The screenshot shows the CAN调试工具 V1.1 interface. The top toolbar includes icons for connecting devices, parameter configuration, language, data recording, firmware upgrade, software multi-open, and logs. Below the toolbar, there are buttons for '打开' (Open), '清空显示' (Clear Display), '显示发送' (Show Send), '停止接收' (Stop Receive), '筛选显示' (Filter Display), and a dropdown for '数据(HEX)'. A search bar and version number '版本号: RS-CANFD-V4.0' are also present.

序号	传输类型	时间	帧ID	扩展帧标志	远程帧标志	协议	数据长度	数据(HEX)	数据(ASCII)
1	发送	14:48:39.589	00 00 00 05	标准帧	数据帧	CAN	8	31 32 33 34 35 36 37 38	12345678
2	发送	14:48:39.730	00 00 00 01	标准帧	数据帧	FD	8	31 32 33 34 35 36 37 38	12345678
3	发送	14:48:39.871	00 00 00 01	标准帧	数据帧	FD BRS	64	31 32 33 34 35 36 37 38 31 ...	12345678123456781234567...
4	发送	14:48:40.026	00 00 00 05	标准帧	数据帧	CAN	8	31 32 33 34 35 36 37 38	12345678
5	发送	14:48:40.180	00 00 00 01	标准帧	数据帧	FD	8	31 32 33 34 35 36 37 38	12345678
6	发送	14:48:40.334	00 00 00 01	标准帧	数据帧	FD BRS	64	31 32 33 34 35 36 37 38 31 ...	12345678123456781234567...

The '调试助手' (Debug Assistant) section shows the 'CAN收发' (CAN Send/Receive) tab. It includes a table for configuring frames to be sent:

帧ID	协议	长度	数据	发送
01	CAN	8	31 32 33 34 35 36 37 38	发送
02	FD	8	31 32 33 34 35 36 37 38	发送
03	FD BRS	64	31 32 33 34 35 36 37 38 31 ...	发送
04	CAN	8		发送
05	CAN	8		发送
06	CAN	8		发送

On the right, the '发送方式' (Send Mode) is set to '正常模式' (Normal Mode), '发送次数' (Send Count) is 2, and '发送间隔' (Send Interval) is 100 ms. There are '发送' (Send) and '停止' (Stop) buttons.

At the bottom, the status bar shows '发送帧数: 82 | 接收帧数: 0 | 清除计数' (Send Frame Count: 82 | Receive Frame Count: 0 | Clear Count) and 'COM31 已打开, 9600, 8, 1, 无校验 状态: 就绪' (COM31 is open, 9600, 8, 1, no parity, status: ready).

例如：一直间隔 100ms，发送数据帧 1。

The screenshot shows the CAN调试工具 V1.1 interface with the 'CAN收发' (CAN Send/Receive) tab selected. The '发送方式' (Send Mode) is set to '循环模式' (Loop Mode), '发送次数' (Send Count) is 1, and '发送间隔' (Send Interval) is 100 ms. The '发送' (Send) button is highlighted.

The '数据(HEX)' dropdown is set to '数据(HEX)'. The status bar shows '发送帧数: 117 | 接收帧数: 0 | 清除计数' (Send Frame Count: 117 | Receive Frame Count: 0 | Clear Count) and 'COM31 已打开, 9600, 8, 1, 无校验 状态: 就绪' (COM31 is open, 9600, 8, 1, no parity, status: ready).

5.2.3. 接收 CAN 数据

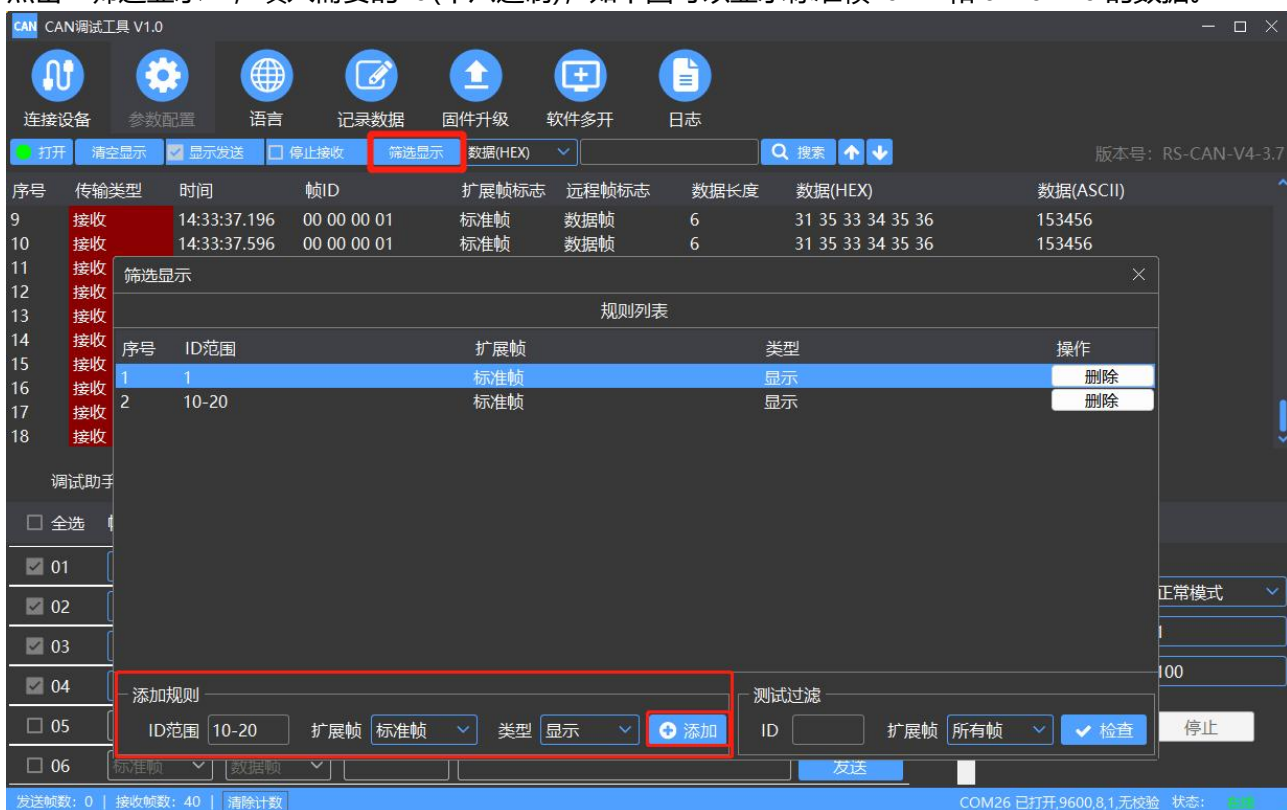
接收到的数据会展示到数据窗口，可以通过另一个转换器发送数据进行测试。



5.2.4. 智能过滤器

通过 4.3 章节的 CAN 过滤器，进行设置，可以实现转换器只接收指定的 CANid。这种方式是通过设置转换器参数实现的，这种方式设置起来比较繁琐。软件提供了另一种方式实现，进行筛选显示。

点击“筛选显示”，填入需要的 id(十六进制)，如下图可以显示标准帧 id=1 和 0x10~20 的数据。



5.2.5. 搜索数据

可以对数据窗口的内容进行搜索，支持对 id 和数据进行搜索。

选择“帧 ID”还是“数据”，填入搜索的内容，点击“搜索”就可以跳转到对应的数据，点击“搜索”旁边的上下箭头，可以跳转到上一条和下一条数据。



5.2.6. 记录数据

➤ 点击“记录数据”开始记录数据窗口的内容。



➤ 点击“停止记录”结束记录，并“确定”保存，将数据保存为.xls 格式文件。

CAN调试工具 V1.0

连接设备 参数配置 语言 停止记录 固件升级 软件多开 日志

打开 清空显示 显示发送 停止接收 筛选显示 数据(HEX) 搜索 版本: RS-CAN-V4-3.7

序号	传输类型	时间	帧ID	扩展帧标志	远程帧标志	数据长度	数据(HEX)	数据(ASCII)
5	接收	15:00:46.597	00 00 01 11	标准帧	数据帧	6	31 35 33 34 35 36	153456
6	接收	15:00:47.259					31 35 33 34 35 36 37 38	15345678
7	发送	15:04:08.273					31 35 33 34 35 36 37 38	15345678
8	发送	15:04:08.476					31 35 33 34 35 36 37 38	15345678
9	发送	15:04:08.681					31 35 33 34 35 36 37 38	15345678
10	发送	15:04:08.904					31 35 33 34 35 36 37 38	15345678
11	发送	15:04:09.437					31 35 33 34 35 36	153456
12	发送	15:04:09.664					31 35 33 34 35 36	153456
13	接收	15:04:13.526					31 35 33 34 35 36 37 38	15345678
14	接收	15:04:13.739					31 35 33 34 35 36 37 38	15345678

调试助手 CAN收发

确定 取消

发送帧数: 14 | 接收帧数: 12 | 清除计数

COM26 已打开,9600,8,1,无校验 状态: 在线

➤ 打开记录的.xls 文件,查看数据。

CAN收发记录									
序号	传输类型	时间	帧ID	扩展帧标志	远程帧标志	数据长度	数据(HEX)	数据(ASCII)	
7	发送	15:04:08.273	00 33 33 33	扩展帧	数据帧	8	31 35 33 34 35 36 37 38	15345678	
8	发送	15:04:08.476	00 33 33 33	扩展帧	数据帧	8	31 35 33 34 35 36 37 38	15345678	
9	发送	15:04:08.681	00 33 33 33	扩展帧	数据帧	8	31 35 33 34 35 36 37 38	15345678	
10	发送	15:04:08.904	00 33 33 33	扩展帧	数据帧	8	31 35 33 34 35 36 37 38	15345678	
11	发送	15:04:09.437	00 00 11 11	标准帧	数据帧	6	31 35 33 34 35 36	153456	
12	发送	15:04:09.664	00 00 11 11	标准帧	数据帧	6	31 35 33 34 35 36	153456	
13	接收	15:04:13.526	00 33 33 33	扩展帧	数据帧	8	31 35 33 34 35 36 37 38	15345678	
14	接收	15:04:13.739	00 33 33 33	扩展帧	数据帧	8	31 35 33 34 35 36 37 38	15345678	

6. 数据模式

转换器支持多种数据转发模式，通过调试工具，可以选择采用哪种数据转发模式。设置好后可以脱离我们的调试工具。自行通过串口按数据格式收发 CAN 总线数据。

模式支持透传模式、带标识透传模式、包模式、Modbus 模式、格式转发模式、Modbus 从机模式和 AT 指令模式 7 种数据转换模式。如果这些模式还不能满足您的需求，请您联系我们远程升级设备。

基本参数介绍：



串口参数配置界面，包含以下参数：

波特率	9600	数据位	8
停止位	1	校验位	无校验
接收间隔	3 毫秒	发送间隔	0 毫秒

接收间隔 (RPACK_TIME)：设置串口接收包间隔时间：串口接收数据帧结束时间间隔 0~255ms。RPACK_TIME 最小值为 1，设置值不能小于对应波特率 3 个字节时间，比如 9600 波特率最小值为 3。

发送间隔 (SPACK_TIME)：设置串口发送包间隔时间：串口发送数据帧时间间隔 0~255ms。

6.1. 透传模式-MODE0

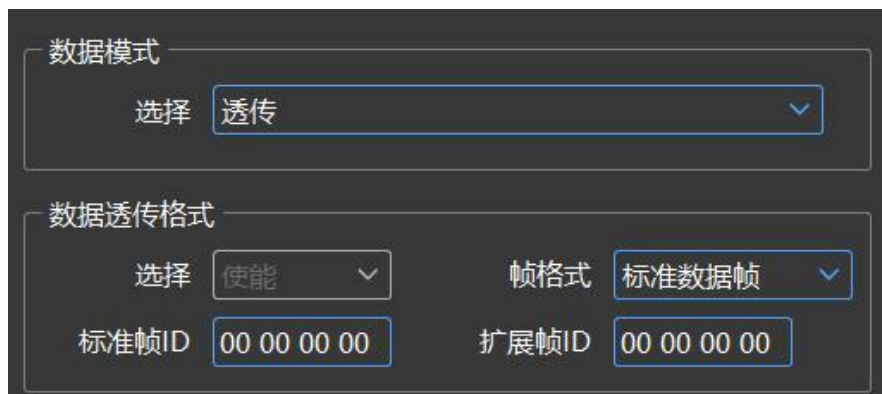
串口发送和接收到的数据为 CAN(FD)总线上报文数据，CAN(FD)总线数据格式由模块自行封装，实现串口“实时传输，所发即所收”，即实现了串口和 CAN 物理接口的交换，又没有改变数据内容，两端总线数据如同透明。

CAN 总线报文封装方式，即报文中的 ID、帧类型等需要通过配置软件提前配置好。

CAN 协议：串口接收机制，串口收到大于 8 个字节就会自动发到 CAN 总线，最后数据不足 8 个字节则等到 RPACK_TIME 时间到就打包发送。

CANFD 协议：串口接收机制，串口收到大于 64 个字节就会自动发到 CAN 总线，最后数据不足 64 个字节则等到 RPACK_TIME 时间到就打包发送。

串口发送机制，收到 CAN 数据就会往串口发送，当设置了发送间隔时间，则数据包之间会间隔发送，间隔时间最小值为 0。



数据模式配置界面，包含以下参数：

数据模式	选择 透传
数据透传格式	选择 使能
帧格式	标准数据帧
标准帧ID	00 00 00 00
扩展帧ID	00 00 00 00

6.1.1. 串行帧转 CAN/FD

串行帧的全部数据依序填充到 CAN (FD) 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 (FD) 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 and 帧 ID 一直保持不变。

CAN 模式: 数据转换对应格式如下表所示。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 (如下表中 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。

串行帧				CAN 报文
			帧信息	用户配置
			帧 ID1	用户配置
			帧 ID2	用户配置
地址	数据		数据	数据 1
0	数据 1			数据 2
1	数据 2			数据 3
2	数据 3			数据 4
3	数据 4			数据 5
4	数据 5			数据 6
5	数据 6			数据 7
6	数据 7			数据 8
7	数据 8			

转换示例: 转换模式: CAN 协议、透传、标准数据帧、标准帧 ID=123 (十六进制)

参数配置

读取参数 写入参数 恢复出厂设置 精简模式 导出配置 导入配置

串口参数

波特率: 9600 数据位: 8 停止位: 1 校验位: 无校验 接收间隔: 3 毫秒 发送间隔: 0 毫秒

数据模式

选择: 透传

数据透传格式

选择: 透传 帧格式: 标准数据帧 标准帧ID: 00 00 01 23 扩展帧ID: 00 00 00 00

CAN 参数

通道: CAN1 工作模式: 正常模式 仲裁域波特率: 500 Kbps 协议: CAN FD加速: 禁止 数据域波特率: 2000 Kbps

CAN 滤波器

序号	使能	滤波器模式	帧格式	滤波寄存器	屏蔽寄存器
01	☒	屏蔽位模式	所有帧	00 00 00 00	00 00 00 00

17:38:19.254 数据读取完成! 状态: 在线

- (左边) 用串口助手给设备发送 2 帧数据 (长度不超过 8 个字节), (右边) 接收端收到 2 帧 CAN 报文

The screenshot displays the CAN debugging tool interface with two panels. The left panel shows the '发送' (Send) tab with two frames of data being sent. The right panel shows the '接收' (Receive) tab with two frames of data being received. A red arrow points from the left panel to the right panel.

序号	传输类型	时间	数据(HEX)
1	发送	17:51:47.715	01 02 03 04 05 06
2	发送	17:51:48.235	01 02 03 04 05 06 07 08

序号	传输类型	时间	帧ID	扩展帧标志	远程帧标志	协议	数据长度	数据(HEX)
1	接收	17:51:47.861	00 00 01 23	标准帧	数据帧	CAN	6	01 02 03 04 05 06
2	接收	17:51:48.379	00 00 01 23	标准帧	数据帧	CAN	8	01 02 03 04 05 06 07 08

- (左边) 用串口助手给设备发送 1 帧数据 10 个字节 (长度超过 8 个字节), (右边) 接收端收到 2 帧 CAN 报文, 1 帧 8 个字节, 1 帧 2 个字节。

The screenshot displays the CAN debugging tool interface with two panels. The left panel shows one frame of data being sent (01 02 03 04 05 06 07 08 09 10). The right panel shows two frames of data being received (01 02 03 04 05 06 07 08 and 09 10). A red arrow points from the left panel to the right panel.

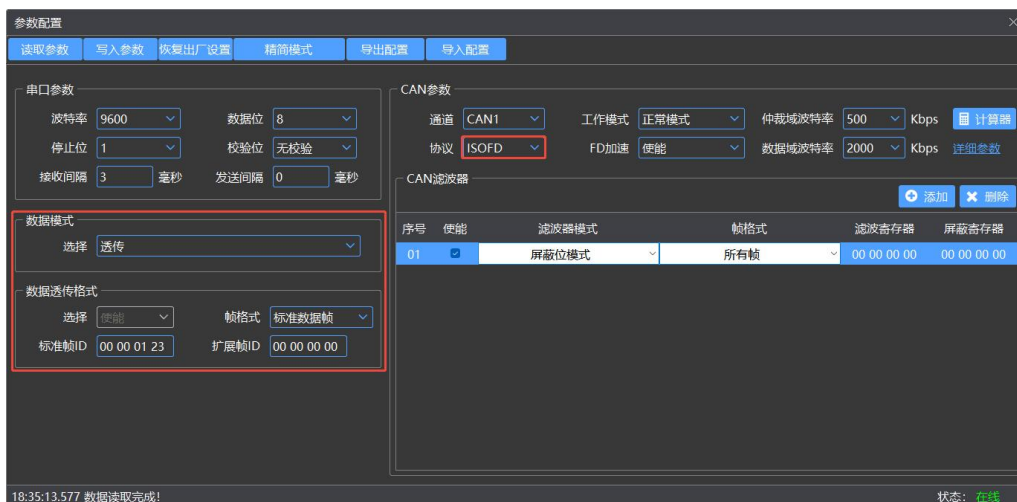
序号	传输类型	时间	数据(HEX)
1	发送	17:58:14.328	01 02 03 04 05 06 07 08 09 10

序号	传输类型	时间	帧ID	扩展帧标志	远程帧标志	协议	数据长度	数据(HEX)
1	接收	17:58:14.545	00 00 01 23	标准帧	数据帧	CAN	8	01 02 03 04 05 06 07 08
2	接收	17:58:14.545	00 00 01 23	标准帧	数据帧	CAN	2	09 10

CANFD 模式:数据转换对应格式下表所示。如果收到串行帧长度小于等于 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 n 为 12)。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为 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。

串行帧				CANFD 报文
			帧信息	用户配置
			帧 ID1	用户配置
			帧 ID2	用户配置
			数据	数据 1
地址	数据		数据 2	
0	数据 1		数据 3	
1	数据 2		数据 4	
2	数据 3		数据 5	
3	数据 4		数据 6	
4	数据 5		数据 7	
5	数据 6		数据 8	
6	数据 7		...	
7	数据 8			
...	...	...		
63	数据 64		数据 64	

转换示例：转换模式：CANFD 协议、透传、标准数据帧、标准帧 ID=123 (十六进制)



- (左边) 串口助手给设备发送 12 个数据 (刚好 1 帧 FD 的长度), (右边) 接收端收到 1 帧 CANFD 报文。

The screenshot shows the CAN debugging tool V1.1 interface. On the left, the 'Serial Assistant' (串口助手) tab is active, showing a list of data to be sent. The first entry is selected, with data '01 02 03 04 05 06 07 08 09 10 01 02'. On the right, the 'CAN收发' (CAN TX/RX) tab is active, showing a list of received frames. The first entry is selected, showing a received frame with ID '00 00 01 23', protocol 'FD|BRS', and length '12'. A red arrow points from the sent data to the received frame.

序号	传输类型	时间	数据(HEX)
1	发送	18:39:01.521	01 02 03 04 05 06 07 08 09 10 01 02

序号	传输类型	时间	帧ID	扩展帧标志	远程帧标志	协议	数据长度	数据(HEX)
1	接收	18:39:01.675	00 00 01 23	标准帧	数据帧	FD BRS	12	01 02 03 04 05 06 07 08 09 10 01 02

- (左边) 串口助手给设备发送 62 个数据, 由于 CANFD 数据长度没有 62, 而最接近 62 的是 48, 故先转成 1 帧 48 的报文。剩余 14 个数据, CANFD 数据长度没有 14, 而最接近 14 的是 12, 故又转成 12 的报文。最后剩余 2 个数据, 直接转成 2 个长度的报文。综上, 一共转成 3 个 CANFD 报文(62=48+12+2)。(右边) 接收端收到 3 帧 CANFD 报文, 长度分别是 48, 12, 2。

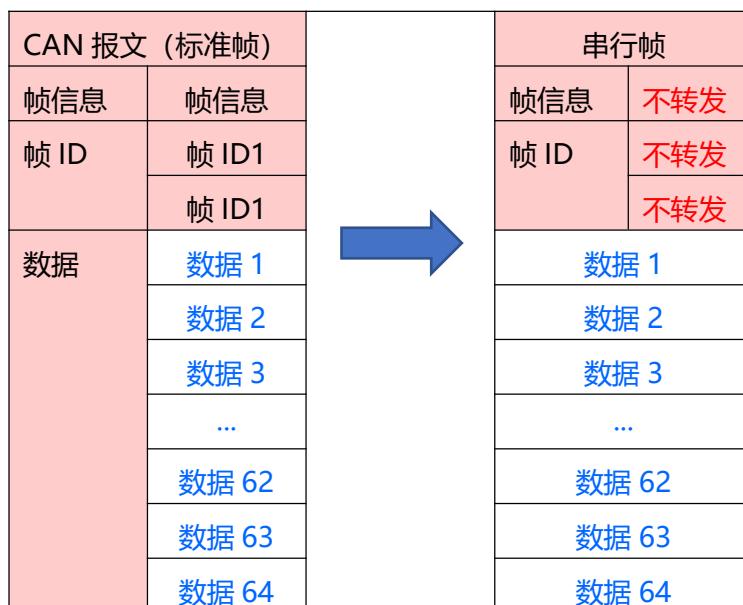
The screenshot shows the CAN debugging tool V1.1 interface. On the left, the 'Serial Assistant' (串口助手) tab is active, showing a list of data to be sent. The first entry is selected, with data '02 03 04 05 06 07 08 09 10 01 02 03 04 05'. On the right, the 'CAN收发' (CAN TX/RX) tab is active, showing a list of received frames. The first three entries are selected, showing received frames with IDs '00 00 01 23', '00 00 01 23', and '00 00 01 23', protocols 'FD|BRS', 'FD|BRS', and 'FD|BRS', and lengths '48', '12', and '2'. A red arrow points from the sent data to the received frames.

序号	传输类型	时间	数据(HEX)
1	发送	18:46:11.320	02 03 04 05 06 07 08 09 10 01 02 03 04 05

序号	传输类型	时间	帧ID	扩展帧标志	远程帧标志	协议	数据长度	数据(HEX)
1	接收	18:46:11.563	00 00 01 23	标准帧	数据帧	FD BRS	48	01 02 03 04 05 06 07 08 09 09 10 01 02 03 04 05 06 07
2	接收	18:46:11.625	00 00 01 23	标准帧	数据帧	FD BRS	12	09 10 01 02 03 04 05 06 07
3	接收	18:46:11.625	00 00 01 23	标准帧	数据帧	FD BRS	2	01 02

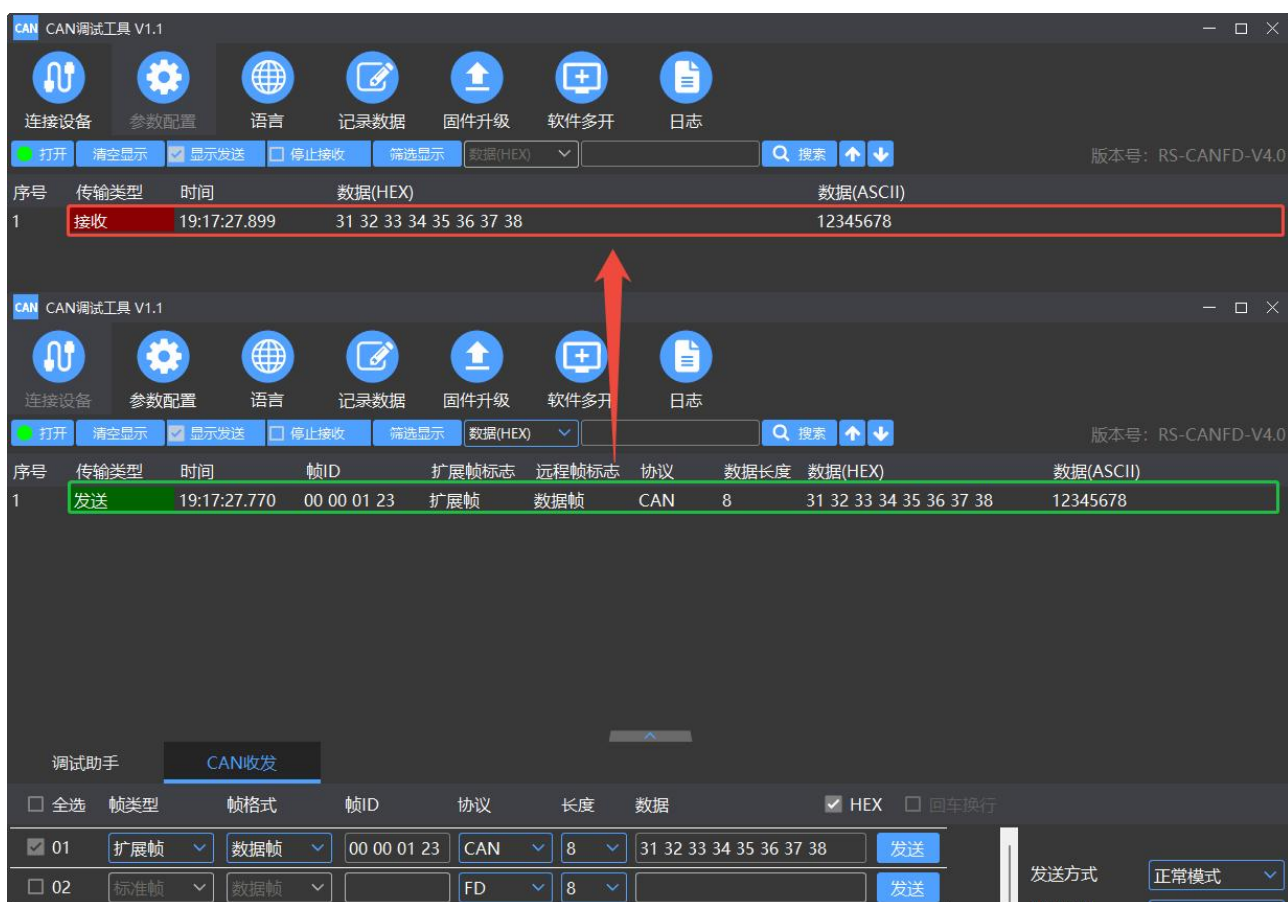
6.1.2. CAN/FD 转串行帧

对于 CAN (CANFD) 总线的报文也是收到一帧就立即转发一帧，注意 CAN 报文数据域长度最大是 8。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。



转换示例：转换模式：透传，收到 CAN 发送的 1 帧数据

➤ (下面) 另一个设备向 CAN 总线发送一个 CAN 报文，(上面) 接收端收到报文后只显示数据部分，其它 CAN 信息都不转发。



6.2. 带标识符透传模式-MODE1

CAN 的帧类型和 ID 由串口数据特定字段决定。数据转发不改变数据内容，实现所发即所收，两端总线数据如同透明。

CAN 总线报文封装方式，即报文中的 ID、帧类型由每一帧串口数据的特定数据决定。当 ID 大于 0x7FF 时，CAN 以扩展数据帧发送，不然以标准数据帧发送。

串口接收机制，当接收数据时间间隔大于接收间隔时间认为收到一包完整数据。

串口发送机制，收到 CAN 完整一包数据才会往串口发送，当设置了发送间隔时间，则数据包之间会间隔发送。

串口一包数据最大长度：256 个字节。

CAN 分包组包机制，采用 CAN 数据的**第一个字节**存储分段信息，在使用上，用户可不关心该机制。串口数据依然是透明传输。

帧信息	由 ID 起始位置和长度决定							
数据位	7	6	5	4	3	2	1	0
数据 1	分段标记	分段类型		分段计数器				
数据 2	串口数据 1							
数据 3	串口数据 2							
数据 4	串口数据 3							
数据 5	串口数据 4							
数据 6	串口数据 5							
数据 7	串口数据 6							
数据 8	串口数据 7							
...								
数据 n	串口数据 n-1							

分段标记：0 表示该报文为单帧报文，1 表示该报文属于分段报文中的一帧；

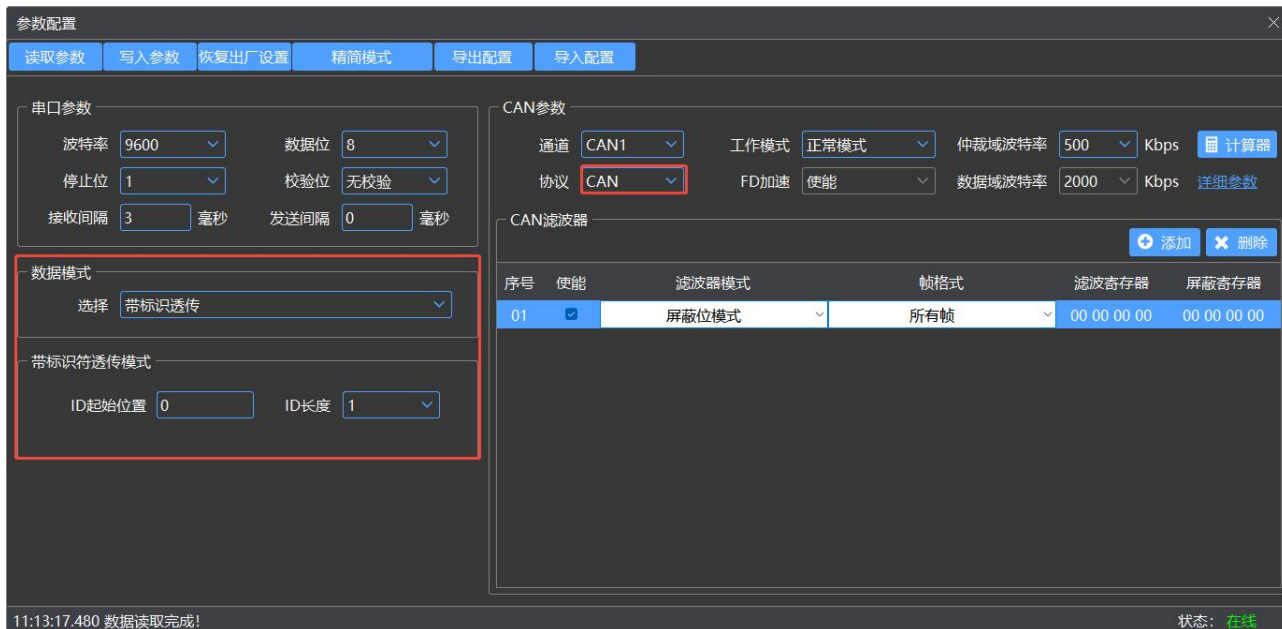
分段类型：0 表示为该报文的第一个分段，1 表示为该报文的中间分段，2 表示为该报文的最后分段，单帧报文该位为 0；

分段计数器：表示该分段在整个报文中的序号，从 1 开始到 31 循环，单帧报文该数值为 0；

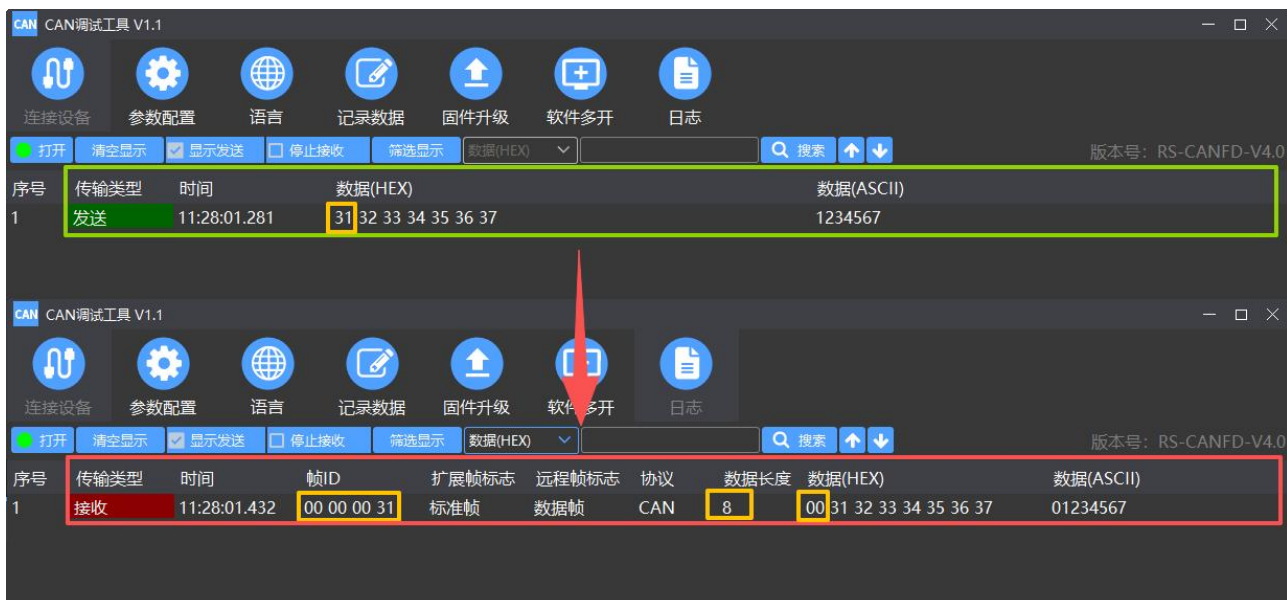
6.2.1. 串行帧转 CAN/FD

CAN 模式: 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，转换器就将收到的数据按分包机制转发到 CAN 总线。数据超过 8 个字节的，会自动拆成多包发送。

转换示例: 转换模式：CAN 协议、带标识符透传，ID 起始位置 0，ID 长度 1



- (上面) 用串口助手给设备发送 7 个数据 (长度不超过 7 个字节)，(下面) 接收端收到 1 帧 CAN 报文。设备收到串口 7 个数据后，将串口的第一个字节 0x31, 做为 CAN 的 ID，在 CAN 数据第一个字节做为分段信息 0x00, 最后把串口的 7 个数据填充到 CAN 数据域的后 7 个字节中，转发到 CAN 总线。



➤ (上面) 用串口助手给设备发送 10 个数据 (长度超过 7 个字节), (下面) 接收端收到 2 帧 CAN 报文。

设备收到串口 10 个数据后, 将串口的第一个字节 0x31, 做为 CAN 的 ID, 在每帧 CAN 数据第一个字节做为分段信息, 最后把串口的 10 个数据分别拆成 2 帧, 第 1 帧 8 个数据 (1 个分段信息 0x81+7 个串口数据), 第 2 帧 4 个数据 (1 个分段信息 0xC2+3 个串口数据)。一共 10 个串口数据转发到 CAN 总线。

Top Screenshot: Transmission Data

序号	传输类型	时间	数据(HEX)	数据(ASCII)
1	发送	11:53:42.576	31 32 33 34 35 36 37 38 39 30	1234567890

Bottom Screenshot: Received CAN Frames

序号	传输类型	时间	帧ID	扩展帧标志	远程帧标志	协议	数据长度	数据(HEX)	数据(ASCII)
1	接收	11:53:42.794	00 00 00 31	标准帧	数据帧	CAN	8	81 31 32 33 34 35 36 37	?1234567
2	接收	11:53:42.794	00 00 00 31	标准帧	数据帧	CAN	4	C2 38 39 30	?890

CANFD 模式: 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, 转换器就将收到的数据按分包机制转发到 CAN 总线。CAN 数据长度按 1~8、12、16、20、24、32、48、64 自动拆包发送。

转换示例: 转换模式: CANFD 协议、带标识符透传, ID 起始位置 0, ID 长度 1

参数配置 (Parameter Configuration)

串口参数 (Serial Port Parameters):

- 波特率: 9600
- 数据位: 8
- 停止位: 1
- 校验位: 无校验
- 接收间隔: 3 毫秒
- 发送间隔: 0 毫秒

数据模式 (Data Mode):

- 选择: 带标识符透传

带标识符透传模式 (With Identifier Transparent Transmission Mode):

- ID起始位置: 0
- ID长度: 1

CAN参数 (CAN Parameters):

- 通道: CAN1
- 协议: ISOFD
- 工作模式: 正常模式
- 仲裁域波特率: 500 Kbps
- 数据域波特率: 2000 Kbps
- FD加速: 使能

CAN滤波器 (CAN Filter):

序号	使能	滤波器模式	帧格式	滤波寄存器	屏蔽寄存器
01	☒	屏蔽位模式	所有帧	00 00 00 00	00 00 00 00

13:54:15.301 数据读取完成! 状态: 在线

- (上面) 用串口助手给设备发送 7 个数据 (长度不超过 7 个字节), (下面) 接收端收到 1 帧 CAN 报文。
- 设备收到串口 7 个数据后, 将串口的第一个字节 0x31, 做为 CAN 的 ID, 在 CAN 数据第一个字节做为分段信息 0x00, 最后把串口的 7 个数据填充到 CAN 数据域的后 7 个字节中, 转发到 CAN 总线。



- (上面) 用串口助手给设备发送 14 个数据 (长度超过 7 个字节), (下面) 接收端收到 2 帧 CAN 报文。
- 设备收到串口 14 个数据后, 将串口的第一个字节 0x31, 做为 CAN 的 ID, 在每帧 CAN 数据第一个字节做为分段信息, 最后把串口的 14 个数据分别拆成 2 帧, 第 1 帧 12 个数据(1 个分段信息 0x81+11 个串口数据), 第 2 帧 4 个数据(1 个分段信息 0xC2+3 个串口数据)。一个 14 个串口数据转发到 CAN 总线。



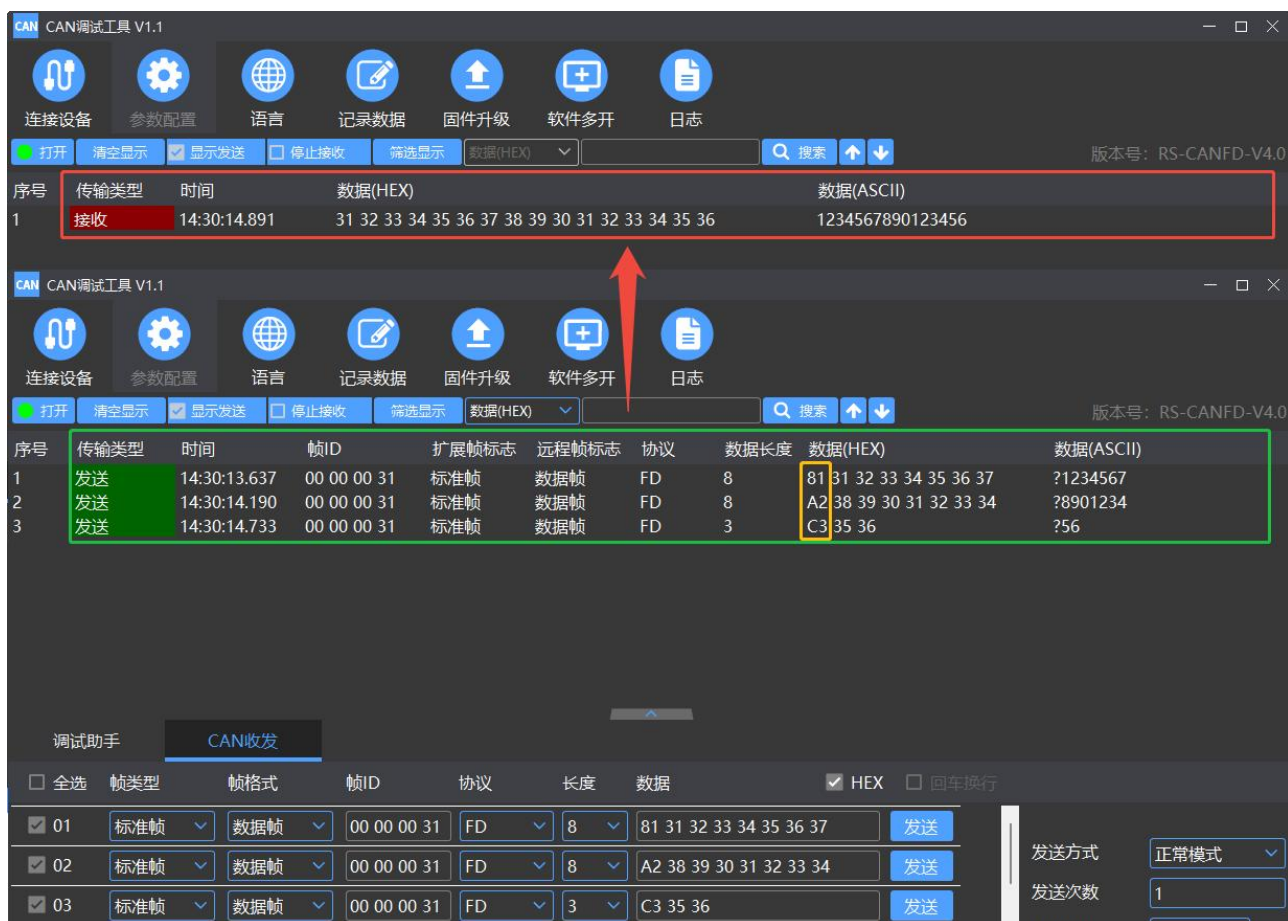
6.2.2. CAN/FD 转串行帧

对于 CAN (CANFD) 总线的报文需要满足分段信息的格式要求，才能组成完整一帧串行数据，转发到串口。

- (下面) 设备往 CAN 总线发送一个单帧报文，CAN 数据第一个字节为 0x00，代表是单帧报文。设备收到后直接转发到串口。



- (下面) 设备往 CAN 总线发送多帧报文，设备收到 CAN 数据后根据分段信息组包转发到串口。
- CAN 报文第 1 帧分段信息=0x81，代表多帧报文的第 1 帧；第 2 帧分段信息=0xA2，代表多帧报文的第 2 帧；第 3 帧分段信息=0xC3，代表多帧报文的第 3 帧也是最后 1 帧；只有收到完整的帧信息，才会转发到串口。



6.3. 包模式-MODE2

包模式根据设置的 CAN 协议类型，分 CAN 协议和 CANFD 协议。CAN 协议只支持 CAN2.0 协议数据收发，CANFD 协议支持 CAN2.0 和 CANFD 协议数据收发。串行帧格式上区别是对应的固定字节数不同，CAN 固定 16 个字节(加包尾 17 个字节)，CANFD 固定 72 个字节(加包尾 73 个字节)。

通过包模式，串口可以收发 CAN 或 CANFD 的所有帧类型，实现串口模拟 CAN 总线通信。

CAN 协议：如下表所示，每包数据包含 16+1 个字节，16 个字节内容（包头+CAN 帧信息+帧 ID+帧数据）+1 个包尾。

CANFD 协议：如下表所示，每包数据包含 72+1 个字节，72 个字节内容（包头+ CANFD 帧信息+帧 ID+帧数据）+1 个包尾。

注意：该转换模式下，设置的 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。

CAN 包模式串行帧(16+1 字节)						
包头	扩展帧标识	远程帧标识	有效数据长度	帧 ID	帧数据	包尾
1Byte	1Byte	1Byte	1Byte	4Byte	8Byte	1Byte
固定值 0xAA	0x00:标准帧 0x01:扩展帧	0x00:数据帧 0x01:远程帧	数据域长度 0~8	标准帧低 11bit 有效 扩展帧低 29bit 有效	有效数据不足 8byte 后面补 0x00	可设置

CANFD 包模式串行帧(72+1 字节)						
包头	扩展帧标识	远程帧标识	有效数据长度	帧 ID	帧数据	包尾
1Byte	1Byte	1Byte	1Byte	4Byte	64Byte	1Byte
固定值 0xAA	00:CAN 标准帧 01:CAN 扩展帧 80:FD 标准帧 81:FD 扩展帧 C0:FD 加速标准帧 C1:FD 加速扩展帧	0x00:数据帧 0x01:远程帧	数据域长度 0~64	标准帧低 11bit 有效 扩展帧低 29bit 有效	有效数据不足 64byte 后面补 0x00	可设置

6.3.1. 串行帧转 CAN/FD

CAN 模式:

转换示例：转换模式：CAN 协议、包模式，使能包尾，包尾=0x7A

参数配置

读取参数 写入参数 恢复出厂设置 精简模式 导出配置 导入配置

串口参数

波特率: 9600 数据位: 8 停止位: 1 校验位: 无校验 接收间隔: 3 毫秒 发送间隔: 0 毫秒

数据模式

选择: 包模式

包模式

包尾选择: 使能 包尾内容: 7A

CAN 参数

通道: CAN1 工作模式: 正常模式 仲裁域波特率: 500 Kbps 协议: CAN FD加速: 禁止 数据域波特率: 2000 Kbps

CAN 滤波器

序号	使能	滤波器模式	帧格式	滤波寄存器	屏蔽寄存器
01	☒	屏蔽位模式	所有帧	00 00 00 00	00 00 00 00

16:01:33.662 数据读取完成!

状态: 在线

➤ (下面) 用串口助手给设备发送 1 帧串口数据, (上面) 接收端收到 1 帧 CAN 报文。

发送串口数据内容=AA 01 00 08 00 00 12 34 31 32 33 34 35 36 37 38 7A, 根据格式解析为扩展帧(01)+数据帧(00)+有效数据长度(08)+帧 ID(00001234)+数据(31 32 33 34 35 36 37 38)。

CAN 调试工具 V1.1

连接设备 参数配置 语言 记录数据 固件升级 软件多开 日志

打开 清空显示 显示发送 停止接收 筛选显示 数据(HEX) 搜索 版本: RS-CANFD-V4.0

序号	传输类型	时间	帧ID	扩展帧标志	远程帧标志	协议	数据长度	数据(HEX)	数据(ASCII)
1	接收	16:19:48.033	00 00 01 23	扩展帧	数据帧	CAN	8	31 32 33 34 35 36 37 38	12345678

CAN 调试工具 V1.1

连接设备 参数配置 语言 记录数据 固件升级 软件多开 日志

打开 清空显示 显示发送 停止接收 筛选显示 数据(HEX) 搜索 版本: RS-CANFD-V4.0

序号	传输类型	时间	数据(HEX)	数据(ASCII)
1	发送	16:19:47.898	AA 01 00 08 00 00 01 23 31 32 33 34 35 36 37 38 7A	?0000#12345678z

6.4. Modbus 模式-MODE3

Modbus 协议是一种标准的工业协议，广泛应用于各种工控和数据采集的场合。该协议开放，实时性强，通讯验证机制可靠，非常适用于通信可靠性要求高的场合。转换器串口侧使用标准的 Modbus RTU 格式，所以模块可以和任何标准的 Modbus RTU 设备连接。模块在 Modbus 总线上扮演的是协议验证和转发的角色，而不是 Modbus 主机或从机，只要按照 Modbus 协议通信即可。

CAN 总线报文封装方式，报文固定采用**标准数据帧**格式，帧 ID=Modbus 协议地址自动填充。

串口接收机制，当接收数据时间间隔大于接收间隔时间，认为收到一包完整数据，判断为有效数据时就会让 CAN 发送。

串口发送机制，收到 CAN 数据进行组包，当接收到完整包就会往串口发送。

CAN 分包组包机制，采用 CAN 数据的第一个字节存储分段信息，在使用上，用户可不关心该机制。串口数据依然是透明传输。

帧 ID	Modbus RTU 的地址码							
数据位	7	6	5	4	3	2	1	0
数据 1	分段标记	分段类型		分段计数器				
数据 2	串口数据 1							
数据 3	串口数据 2							
数据 4	串口数据 3							
数据 5	串口数据 4							
数据 6	串口数据 5							
数据 7	串口数据 6							
数据 8	串口数据 7							
...								
数据 n	串口数据 n-1							

分段标记：0 表示该报文为单帧报文，1 表示该报文属于分段报文中的一帧；

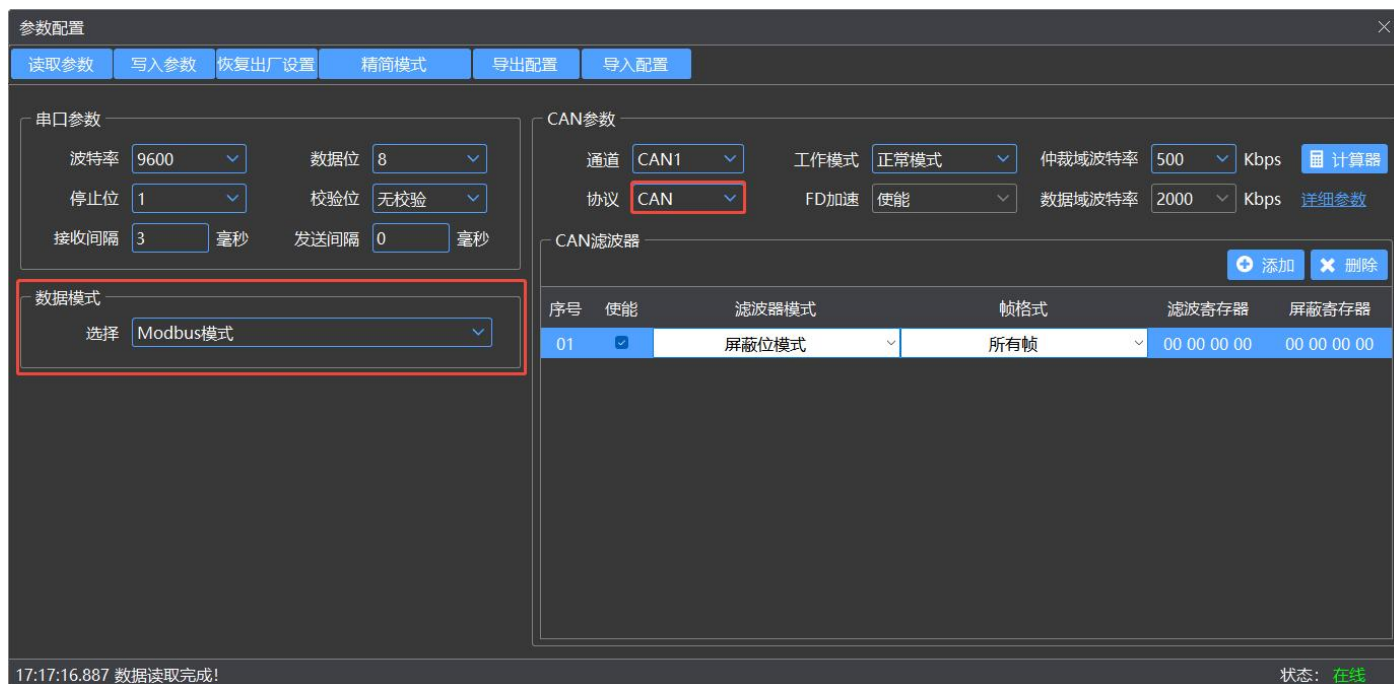
分段类型：0 表示为该报文的第一个分段，1 表示为该报文的中间分段，2 表示为该报文的最后分段，单帧报文该位为 0；

分段计数器：表示该分段在整个报文中的序号，从 1 开始到 31 循环，单帧报文该值为 0；

6.4.1. 串行帧转 CAN/FD

CAN 模式: 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，转换器就将收到的数据判断是否是 Modbus 协议帧。验证是 Modbus 协议帧，就会把 Modbus 的地址设置为 CANid，并以标准数据帧，自动拆成多帧发送。

转换示例： 转换模式：CAN 协议、Modbus 模式



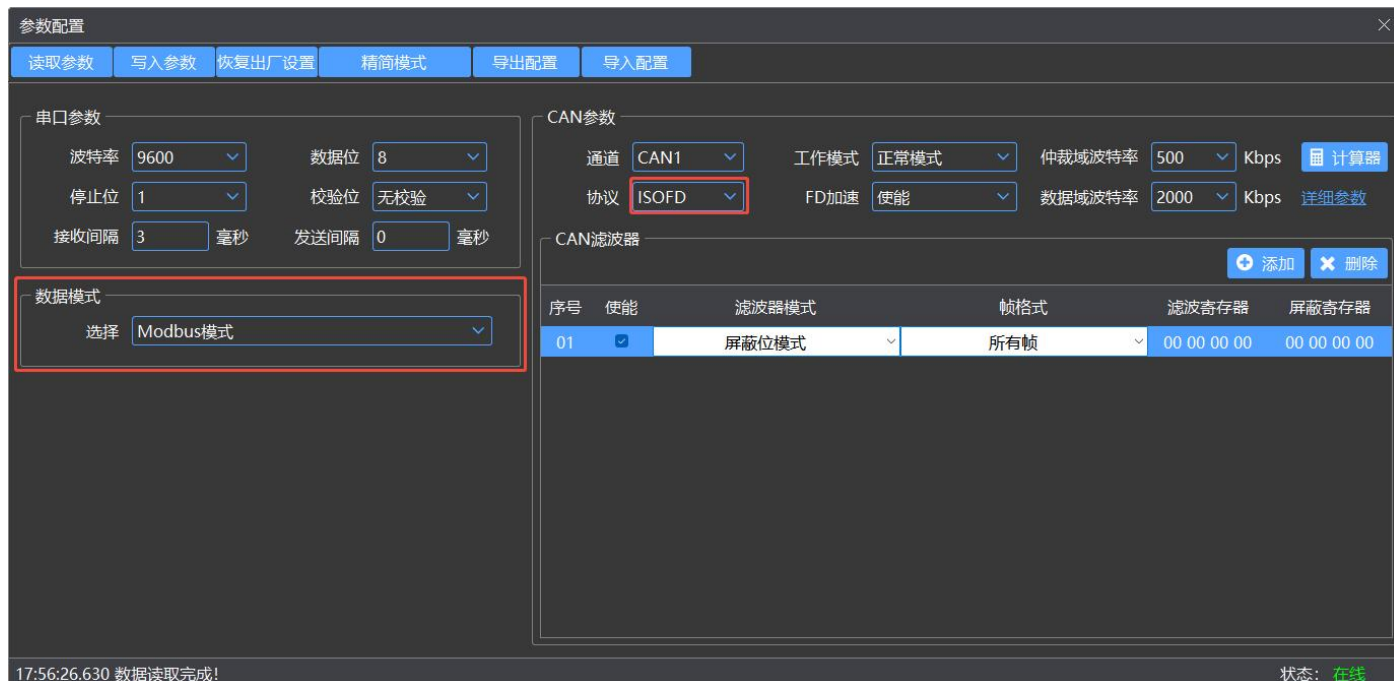
➤ (下面) 用串口助手给设备发送一帧 Modbus 数据，(上面) 接收端收到 2 帧 CAN 报文。

设备收到一帧 Modbus 数据(01 03 00 00 00 01 84 0A)，将 Modbus 地址 0x01, 做为 CAN 的 ID，在每帧 CAN 数据第一个字节做为分段信息, 最后把串口的 8 个数据分别拆成 2 帧, 第 1 帧 8 个数据(1 个分段信息 0x81+7 个串口数据)，第 2 帧 2 个数据(1 个分段信息 0xC2+1 个串口数据)。一共 8 个串口数据转发到 CAN 总线。



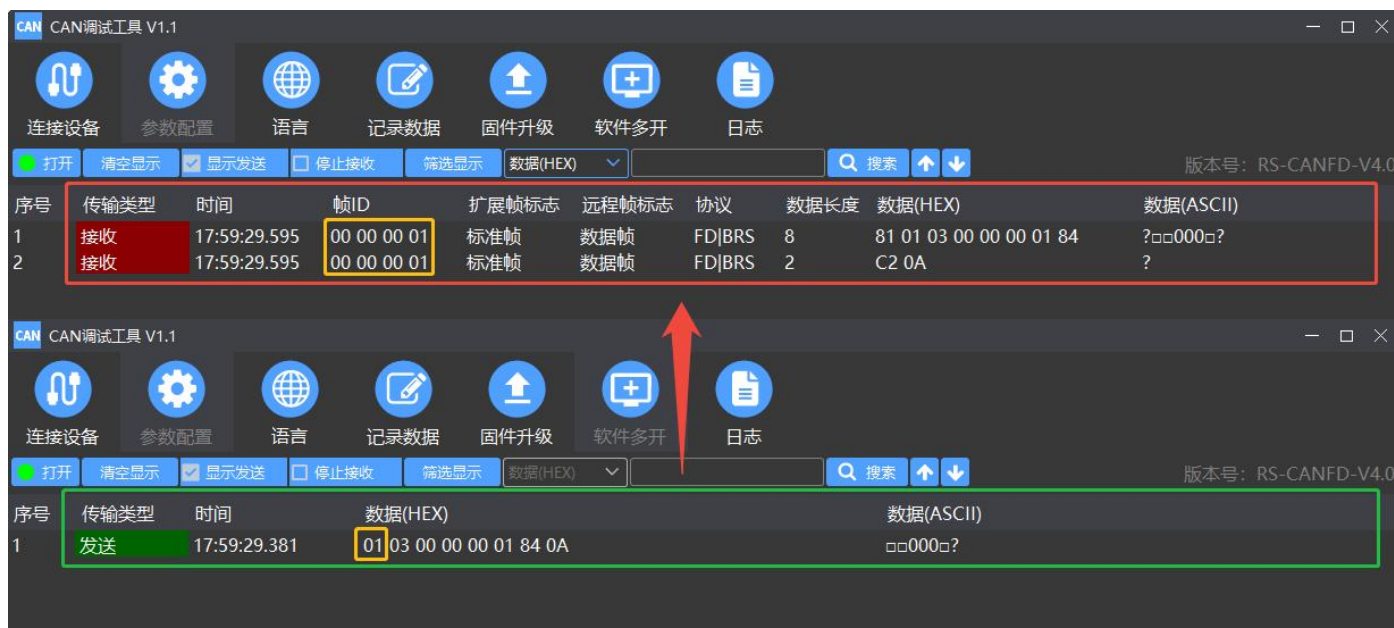
CANFD 模式: 转换器会根据用户配置的串行帧之间的时间间隔进行超时，超时后还没接收到更多数据，转换器就将收到的数据判断是否是 Modbus 协议帧。验证是 Modbus 协议帧，就会把 Modbus 的地址设置为 CANid，并以标准数据帧，数据长度按 1~8、12、16、20、24、32、48、64 拆成多帧发送。

转换示例： 转换模式：CANFD 协议、Modbus 模式



➤ (下面) 用串口助手给设备发送一帧 Modbus 数据，(上面) 接收端收到 2 帧 CANFD 报文。

设备收到一帧 Modbus 数据(01 03 00 00 00 01 84 0A)，将 Modbus 地址 0x01, 做为 CAN 的 ID，在每帧 CAN 数据第一个字节做为分段信息，最后把串口的 8 个数据分别拆成 2 帧，第 1 帧 8 个数据(1 个分段信息 0x81+7 个串口数据)，第 2 帧 2 个数据(1 个分段信息 0xC2+1 个串口数据)。一共 8 个串口数据转发到 CAN 总线。



6.4.2. CAN/FD 转串行帧

对于 CAN (CANFD) 总线的报文需要满足分段信息的格式要求，才能组成完整一帧串行数据，转发到串口。转换器只对分段信息进行判断，不对收到的数据进行 Modbus CRC 判断。

➤ (下面) 设备往 CAN 总线发送多帧报文，转换器收到 CAN 数据后根据分段信息组包转发到串口。

CAN 报文第 1 帧分段信息=0x81，代表多帧报文的第 1 帧；第 2 帧分段信息=0xC2，代表多帧报文的第 2 帧也是最后 1 帧；只有收到完整的帧信息，才会转发到串口。

Top Screenshot: Received Message

序号	传输类型	时间	数据(HEX)	数据(ASCII)
1	接收	18:11:33.598	01 03 00 00 00 01 84 0A	□□ □?

Bottom Screenshot: Transmitted Messages

序号	传输类型	时间	帧ID	扩展帧标志	远程帧标志	协议	数据长度	数据(HEX)	数据(ASCII)
1	发送	18:11:32.930	00 00 00 01	标准帧	数据帧	FD BRS	8	81 01 03 00 00 00 01 84	?□□000□?
2	发送	18:11:33.475	00 00 00 01	标准帧	数据帧	FD BRS	2	C2 0A	?

6.5. 格式转发模式- MODE4

格式转发模式根据设置的 CAN 协议类型，分 CAN 协议和 CANFD 协议。CAN 协议只支持 CAN2.0 协议数据收发，CANFD 协议支持 CAN2.0 和 CANFD 协议数据收发。两种协议的串行帧格式一致。

通过格式转发模式，串口可以收发 CAN 或 CANFD 的所有帧类型，实现串口模拟 CAN 总线通信。

注意：该转换模式下，设置的 CAN ID 无效，因为此时发送的标识符（帧 ID）由串行帧中的帧 ID 数据填充。配置的帧类型也无效，由串行帧中的帧信息来决定帧类型。

格式转发串行帧(6+n 字节)			
帧标识	帧 ID	有效数据长度	帧数据
1Byte	4Byte	1Byte	nByte
0x00:CAN 标准数据帧 0x01:CAN 标准远程帧 0x02:CAN 扩展数据帧 0x03:CAN 扩展远程帧 0x80:CANFD 标准数据帧 0x82:CANFD 扩展数据帧 0xC0:CANFD 加速标准数据帧 0xC2:CANFD 加速扩展数据帧	标准帧低 11bit 有效 扩展帧低 29bit 有效	CAN 数据域长度: 0~8 CANFD 数据域长度: 1~8、12、16、20、24、 32、48、64	数据域内容

6.5.1. 串行帧转 CAN/FD

CAN 模式:

转换示例：转换模式：CAN 协议、格式转发

参数配置

读取参数

写入参数

恢复出厂设置

精简模式

导出配置

导入配置

串口参数

波特率 9600

数据位 8

停止位 1

校验位 无校验

接收间隔 3 毫秒

发送间隔 0 毫秒

数据模式

选择 格式转发

CAN 参数

通道 CAN1

工作模式 正常模式

仲裁域波特率 500 Kbps

协议 CAN

FD加速 使能

数据域波特率 2000 Kbps

计算器

详细参数

CAN 滤波器

添加

删除

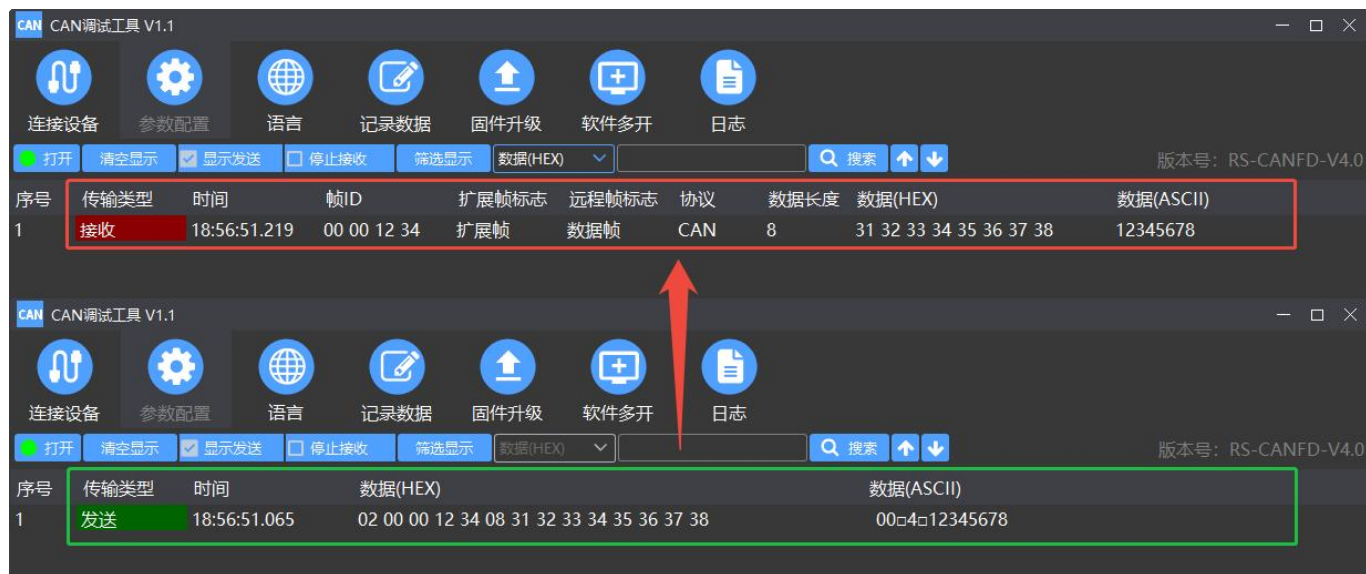
序号	使能	滤波器模式	帧格式	滤波寄存器	屏蔽寄存器
01	☒	屏蔽位模式	所有帧	00 00 00 00	00 00 00 00

18:47:11.935 数据读取完成!

状态: 在线

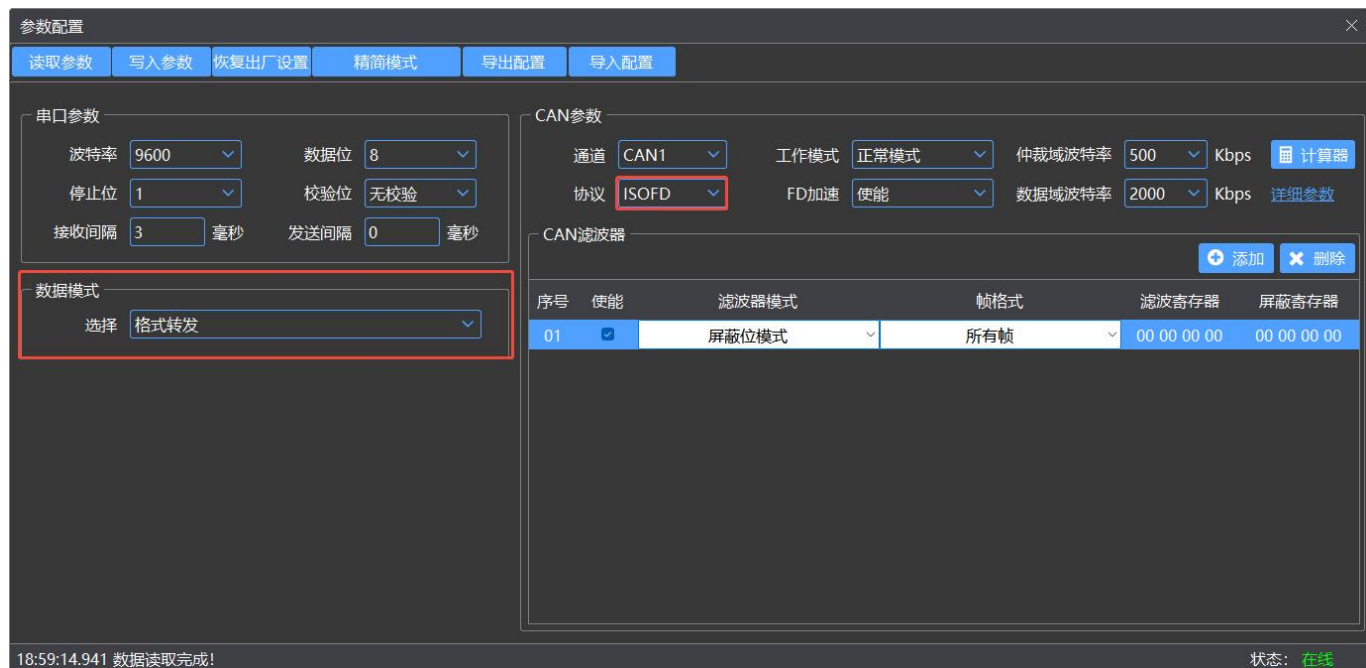
➤ (下面) 用串口助手给设备发送 1 帧串口数据, (上面) 接收端收到 1 帧 CAN 报文。

发送串口数据内容=02 00 00 12 34 08 31 32 33 34 35 36 37 38, 根据格式解析为 CAN 扩展帧数据帧(02)+帧 ID(00001234)+有效数据长度(08)+数据(31 32 33 34 35 36 37 38)。



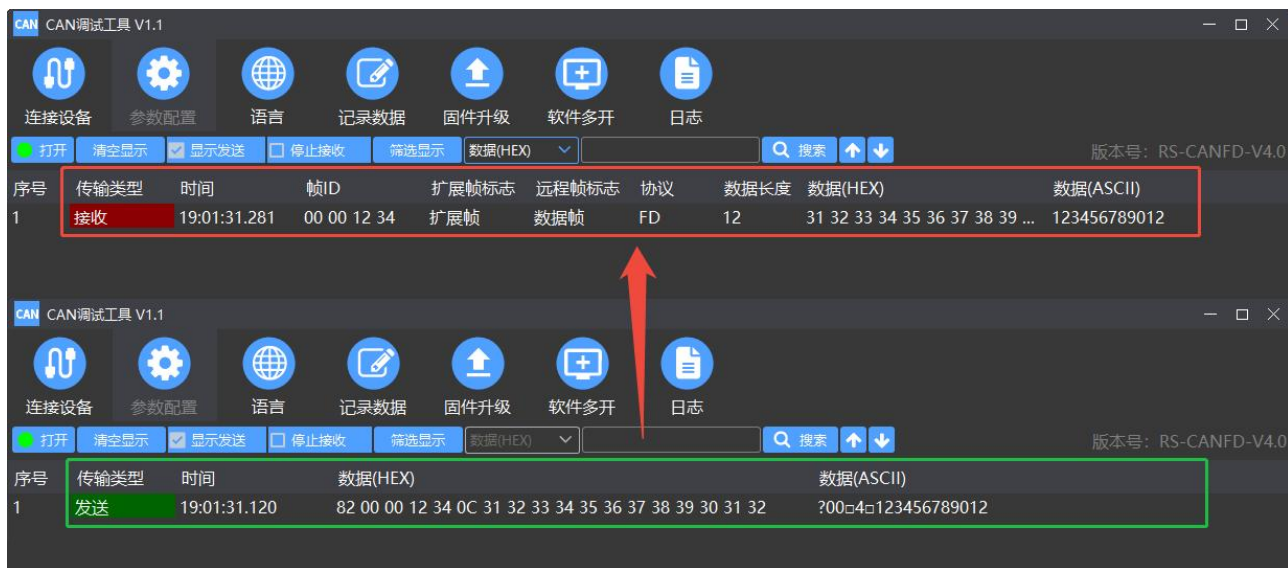
CANFD 模式:

转换示例: 转换模式: CANFD 协议(ISOFD/BOSCHFD)、格式转发



➤ （下面）用串口助手给设备发送 1 帧串口数据，（上面）接收端收到 1 帧 CAN 报文。

发送串口数据内容=82 00 00 12 34 0C 31 32 33 34 35 36 37 38 39 30 31 32，根据格式解析为 CANFD 扩展数据帧(82)+帧 ID(00001234)+有效数据长度(0C=12)+有效数据(31 32 33 34 35 36 37 38 39 30 31 32)。



6.5.2. CAN/FD 转串行帧

包模式下，串行帧和 CAN/FD 帧是双向转换，转换格式是一致的，都是一帧串行帧对应一帧 CAN 帧。CAN/FD 转串行帧的说明和示例可以参考“串行帧转 CAN/FD”。

6.6. Modbus 从机模式- MODE5

Modbus 从机模式目前只支持 CAN 协议，不支持 CANFD 协议。

Mode5 不同于 Mode3 的 Modbus 模式，Mode5 是 Modbus 从机模式，支持标准 Modbus-RTU 协议的 10H 和 03H 功能码。

通过写多个寄存器（10H）指令，可以实现 CAN 数据的发送，读多个寄存器（03H）指令，可以实现 CAN 缓存数据的读取。CAN 收发缓冲器具备 50 帧的 CAN 发送缓存和 128 帧的接收缓存。

通过配置软件可以修改模块地址。

➤ 10H 写多个寄存器指令解析：

请求

功能码	1 个字节	0x10
起始地址	2 个字节	0x0000 至 0xFFFF 不做判断
寄存器数量	2 个字节	0x0001 至 0x0018
字节数	1 个字节	2*N
寄存器值	N*2 个字节	值

N= 寄存器数量

响应正确

功能码	1 个字节	0x10
起始地址	2 个字节	0x0000 至 0xFFFF
寄存器数量	2 个字节	0x0001 至 0x0018

响应错误

差错码	1 个字节	0x90
异常码	1 个字节	0x03

注意：其中起始地址，模块不做任何判断

寄存器值解析

寄存器	功能	
寄存器 0	CAN 帧类型	0 标准数据 1 标准远程 2 扩展数据 3 扩展远程
寄存器 1	CAN 帧 ID	标准帧 ID: 0~0x7FF
寄存器 2		扩展帧 ID: 0~0x 1FFFFFFF
寄存器 3	CAN 帧数据长度	0~8
寄存器 4~n	CAN 数据域	每个寄存器的低字节对应一个 CAN 数据

转换示例:

通过标准帧 ID0x5B,发送 8 个字节的 CAN 数据 0x11~0x88, 模块 Modbus 从机地址为 1

串行帧		说明		标准帧	CAN 报文
从机地址	0x01			帧信息	0x08
功能码	0x10			帧 ID1	0x00
起始地址	0x0000	该值不做判断		帧 ID2	0x5B
寄存器数量	0x000C			数据	0x11
字节数	0x18				0x22
寄存器值	0x0000	CAN 帧类型			0x33
寄存器值	0x0000	ID 高字节			0x44
寄存器值	0x005B	ID 低字节			0x55
寄存器值	0x0008	帧长度			0x66
寄存器值	0x0011	帧数据 低字节有效			0x77
寄存器值	0x0022				0x88
寄存器值	0x0033				
寄存器值	0x0044				
寄存器值	0x0055				
寄存器值	0x0066				
寄存器值	0x0077				
寄存器值	0x0088				
CRC16	0x852C				

➤ 03H 多个寄存器指令解析:

请求

功能码	1 个字节	0x03
起始地址	2 个字节	不做判断
寄存器数量	2 个字节	固定 13

响应

功能码	1 个字节	0x03
字节数	1 个字节	0x0002 至 0x001A
寄存器值	N*2 个字节	N=寄存器的数量

寄存器值解析

寄存器	功能	
寄存器 0	缓存剩余帧数	
寄存器 1	帧类型	0 标准数据 1 标准远程 2 扩展数据 3 扩展远程
寄存器 2	CAN 帧 ID	标准帧 ID: 0~0x7FF
寄存器 3		扩展帧 ID: 0~0x 1FFFFFFF
寄存器 4	CAN 帧长度	0x0000~0x0008
寄存器 5~n	CAN 帧数据	每个寄存器的低字节对应一个 CAN 数据

转换示例:

读 CAN 接收缓存

请求			响应		说明
从机地址	0x01		从机地址	0x01	
功能码	0x03		功能码	0x03	
起始地址	0x0000	该值不做判断	字节数	0x1A	
寄存器数量	0x000D	固定 13	寄存器值	0x0000	剩余帧数量
CRC16	0x840F		寄存器值	0x0002	帧类型 0x02 为扩展数据帧
			寄存器值	0x0001	扩展帧 ID:0x12345
			寄存器值	0x2345	
			寄存器值	0x0008	
			寄存器值	0x0011	帧数据长度
			寄存器值	0x0011	帧数据, 低字节有效 空闲数据, 全部补 0
			寄存器值	0x0022	
			寄存器值	0x0033	
			寄存器值	0x0044	
			寄存器值	0x0055	
			寄存器值	0x0066	
			寄存器值	0x0077	
			寄存器值	0x0088	
			CRC16	0x20E7	

6.7. AT 指令模式- MODE6

AT 指令模式根据设置的 CAN 协议类型, 分 CAN 协议和 CANFD 协议。CAN 协议只支持 CAN2.0 协议数据收发, CANFD 协议支持 CAN2.0 和 CANFD 协议数据收发。

该模式以字符 AT 为帧头, 回车换行符为帧尾, 通过 AT 指令格式实现模拟 CAN 功能, 实现标准帧、扩展帧的收发。

注意: 该转换模式下, 设置的 CAN ID 无效, 因为此时发送的标识符 (帧 ID) 由串行帧中的帧 ID 数据填充。配置的帧类型也无效, 由串行帧中的帧信息来决定帧类型。

AT 指令格式说明

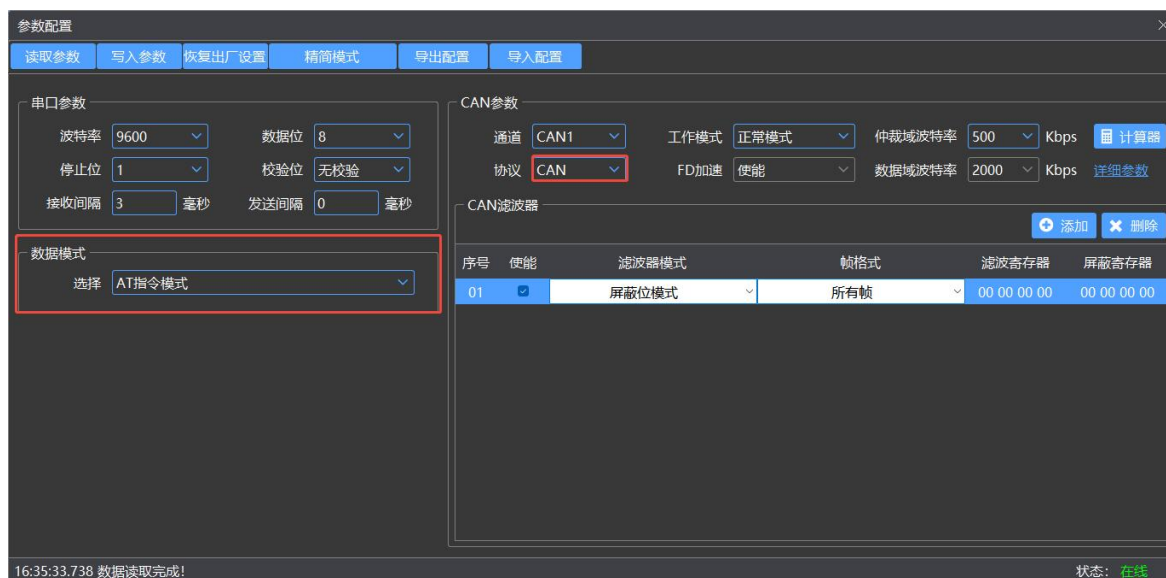
帧格式	内容	长度
帧头	AT	2 个字节
报文标识符	CAN_id	4 个字节
数据长度	Data_len	1 个字节
数据内容	Data	n 个字节
帧尾	<CR> <LF>	2 个字节

CAN(报文标识符)总共 4 个字节, 也就是 32 位, 包含帧 ID 和帧信息。其内容定义如下:

CAN 报文标识符格式				
bit31-bit21	bit20-bit3	bit2	bit1	bit0
标准帧 ID, 扩展帧 ID 前 11 位	扩展帧 ID 后 18 位	1:扩展帧 0:标准帧	1:远程帧 0:数据帧	固定为 0

CANFD 报文标识符格式				
bit31-bit21	bit20-bit3	bit2	bit1	bit0
标准帧 ID, 扩展帧 ID 前 11 位	扩展帧 ID 后 18 位	1:扩展帧 0:标准帧	1:CANFD 加速 0:CANFD 不加速	固定为 1

CAN 模式: 只支持 CAN 协议



转换示例：

1. CAN 标准数据帧, id=0x01, 发送 4 个字节的数据 0x11 22 33 44

报文: 41 54 00 20 00 00 04 11 22 33 44 0D 0A

内容	帧头	报文标识符	数据长度	数据内容	帧尾
16 进制	41 54	00 20 00 00	04	11 22 33 44	0D 0A

2. CAN 扩展数据帧, id=0x1234, 发送 6 个字节数据 0x11 22 33 44 55 66

报文: 41 54 00 00 91 A4 06 11 22 33 44 55 66 0D 0A

内容	帧头	报文标识符	数据长度	数据内容	帧尾
16 进制	41 54	00 00 91 A4	06	11 22 33 44 55 66	0D 0A

3. CAN 标准远程帧, id=0x01, 发送一个远程帧, 请求长度 0

报文: 41 54 00 20 00 02 00 0D 0A

内容	帧头	报文标识符	数据长度	数据内容	帧尾
16 进制	41 54	00 20 00 02	00	无	0D 0A

4. CAN 扩展远程帧, id=0x1234, 发送一个远程帧, 请求长度 0

报文: 41 54 00 00 91 A6 00 0D 0A

内容	帧头	报文标识符	数据长度	数据内容	帧尾
16 进制	41 54	00 00 91 A6	00	无	0D 0A

CANFD 模式：支持 CAN 和 CANFD 协议

参数配置

读取参数 写入参数 恢复出厂设置 精简模式 导出配置 导入配置

串口参数

波特率: 9600 数据位: 8 停止位: 1 校验位: 无校验 接收间隔: 3 毫秒 发送间隔: 0 毫秒

数据模式

选择: AT指令模式

CAN参数

通道: CAN1 工作模式: 正常模式 仲裁域波特率: 500 Kbps 协议: ISOFD FD加速: 使能 数据域波特率: 2000 Kbps

CAN滤波器

序号	使能	滤波器模式	帧格式	滤波寄存器	屏蔽寄存器
01	☒	屏蔽位模式	所有帧	00 00 00 00	00 00 00 00

16:42:02.859 数据读取完成!

状态: 在线

转换示例：

1. CANFD 不加速，标准数据帧，id=0x01，发送 4 个字节的数据 0x11 22 33 44

报文：41 54 00 20 00 01 04 11 22 33 44 0D 0A

内容	帧头	报文标识符	数据长度	数据内容	帧尾
16 进制	41 54	00 20 00 01	04	11 22 33 44	0D 0A

2. CANFD 不加速，扩展数据帧，id=0x1234，发送 6 个字节数据 0x11 22 33 44 55 66

报文：41 54 00 00 91 A5 06 11 22 33 44 55 66 0D 0A

内容	帧头	报文标识符	数据长度	数据内容	帧尾
16 进制	41 54	00 00 91 A5	06	11 22 33 44 55 66	0D 0A

3. CANFD 加速，标准数据帧，id=0x01，发送 4 个字节的数据 0x11 22 33 44

报文：41 54 00 20 00 03 04 11 22 33 44 0D 0A

内容	帧头	报文标识符	数据长度	数据内容	帧尾
16 进制	41 54	00 20 00 03	04	11 22 33 44	0D 0A

4. CANFD 加速，扩展数据帧，id=0x1234，发送 6 个字节数据 0x11 22 33 44 55 66

报文：41 54 00 00 91 A7 06 11 22 33 44 55 66 0D 0A

内容	帧头	报文标识符	数据长度	数据内容	帧尾
16 进制	41 54	00 00 91 A7	06	11 22 33 44 55 66	0D 0A

5. CAN 标准数据帧, id=0x01, 发送 4 个字节的数据 0x11 22 33 44

报文: 41 54 00 20 00 00 04 11 22 33 44 0D 0A

内容	帧头	报文标识符	数据长度	数据内容	帧尾
16 进制	41 54	00 20 00 00	04	11 22 33 44	0D 0A

6. CAN 扩展数据帧, id=0x1234, 发送 6 个字节数据 0x11 22 33 44 55 66

报文: 41 54 00 00 91 A4 06 11 22 33 44 55 66 0D 0A

内容	帧头	报文标识符	数据长度	数据内容	帧尾
16 进制	41 54	00 00 91 A4	06	11 22 33 44 55 66	0D 0A

7. CAN 标准远程帧, id=0x01, 发送一个远程帧, 请求长度 0

报文: 41 54 00 20 00 02 00 0D 0A

内容	帧头	报文标识符	数据长度	数据内容	帧尾
16 进制	41 54	00 20 00 02	00	无	0D 0A

8. CAN 扩展远程帧, id=0x1234, 发送一个远程帧, 请求长度 0

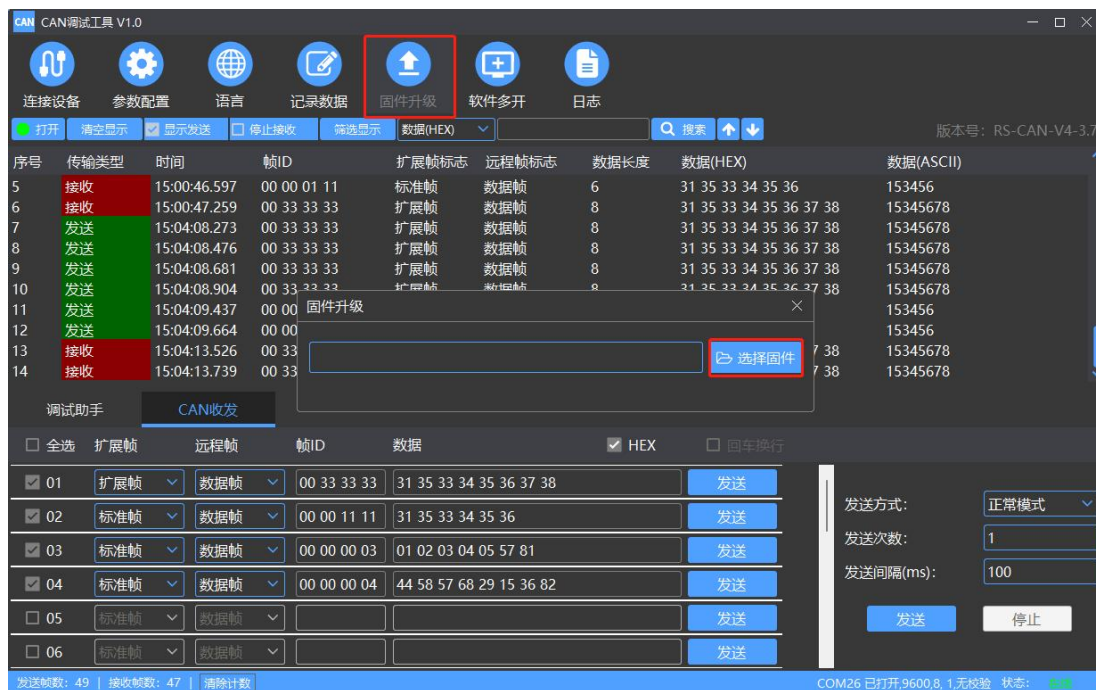
报文: 41 54 00 00 91 A6 00 0D 0A

内容	帧头	报文标识符	数据长度	数据内容	帧尾
16 进制	41 54	00 00 91 A6	00	无	0D 0A

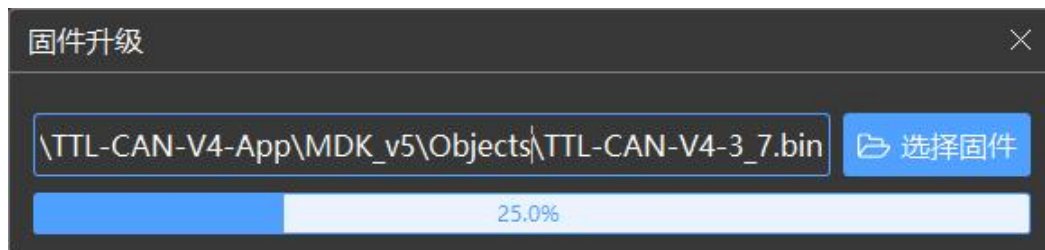
7. 固件升级

泥人科技会根据市场需求不定期都对设备的功能进行升级，最新固件向所有用户公开，用户可选择性进行固件升级。同时用户个性定制软件功能时，不需要将设备返回厂家，而直接升级定制版的固件即可。

- 点击“固件升级”，选择固件路径。



- 选好固件后，会自动启动升级，升级完成会提示“升级成功”或“升级失败”。



- 查看设备版本号，软件右上方，显示当前转换器固件版本。

