



# 硬件使用说明书—TC1001

版本：V1.10 | 中文

## 文档修订历史:

文件版本	日期	更新内容	备注
<b>V1.00</b>	2023.5.5	创建文档	
V1.10	2023.7.5	修改文档	

## 版权信息

上海同星智能科技有限公司

上海市嘉定区曹安公路 4801 号 6/8 层

本着为用户提供更好服务的原则，上海同星智能科技有限公司（下称“同星智能”）在本手册中将尽可能地为用户呈现详实、准确的产品信息。但鉴于本手册的内容具有一定的时效性，同星智能不能完全保证该文档在任何时段的时效性与适用性。

本用户手册中的信息和数据如有更改，恕不另行通知。为了得到最新版本的信息，请您访问[同星智能官方网站](#)或者与同星智能工作人员联系。感谢您的包容与支持！

未经同星智能书面许可，不得以任何形式或任何方式复制本手册的任何部分。

@版权所有 2023，上海同星智能科技有限公司。保留所有权利。

## 目录

1. 产品简介 .....	1
1.1 产品概述 .....	1
1.2 典型应用 .....	1
1.3 功能和参数 .....	1
1.3.1 功能特征 .....	1
1.3.2 技术参数 .....	2
1.3.3 电气参数 .....	2
1.4 发货清单 .....	2
2. 硬件接口说明 .....	3
2.1 接口说明 .....	3
2.2 LED 指示灯说明 .....	3
3. 快速使用 .....	4
3.1 下载安装 TSMaster 上位机 .....	4
3.2 连接设备，配置通道 .....	7
3.3 报文发送 .....	8
3.4 帮助文档及视频教学 .....	9
3.5 TSMaster API 二次开发 .....	9
3.5.1 Python 调用动态库 .....	10
3.5.2 C 调用动态库 .....	10
3.5.3 调用接口示例 .....	10
3.6 示例工程 .....	12
4. 检查和维护 .....	13
5. 常见问题及解答 .....	14
5.1 线路连接正确但无法正常通信： .....	14
5.2 报文观察不便以及信号过滤： .....	15
5.3 如何加载数据库： .....	16
5.4 如何自动记录报文： .....	16
5.5 如何回放报文（离线回放和在线回放）： .....	17
6. 附录 CAN2.0B 协议帧格式 .....	18
6.1 CAN2.0 标准帧： .....	18
6.2 CAN2.0 扩展帧： .....	19
7. 免责声明 .....	19
关于同星 .....	

## 1. 产品简介

### 1.1 产品概述

TC1001 是同星智能推出的便携式、易安装的 1 路 CAN 总线转 USB 接口的设备，最高速率 1Mbps，产品采用高速 USB2.0 接口与 PC 连接，Windows 系统免驱设计使得设备具备极佳的系统兼容性。

配合功能强大的 TSMaster 软件，支持加载 DBC 和 ARXML 数据库文件，可以很方便地监控、分析、仿真 CAN 总线数据，也可以支持 UDS 诊断、ECU 刷写、CCP/XCP 标定等功能。

可用于 Windows 和 Linux 的二次开发 API，可支持各类开发环境，如 C++、C#、LabView、Python 等，方便集成到各种测试系统中，高效易用。

### 1.2 典型应用

- ✓ 整车 CAN 总线数据采集
- ✓ 域控制器测试
- ✓ 各种自动化测试系统

### 1.3 功能和参数

#### 1.3.1 功能特征

- ✓ us（微秒）级硬件报文时间戳，满足高阶需求。
- ✓ 便携式设计，独特设计的安装孔，便于集成到各种设备中或仪表面板上。
- ✓ 高速 USB2.0 接口，Windows、Linux 系统系统免驱设计，具极佳的系统兼容性。
- ✓ CAN 通道 DC2500V 隔离。
- ✓ 汽车级设计，支持 dbc 文件、a2l 文件、blf 文件、asc 文件。
- ✓ CAN 通道波特率 125Kbps—8Mbps 可调。
- ✓ 支持 blf，asc 格式格式数据记录和离线/在线回放。
- ✓ 可支持 UDS 诊断及 CCP 标定。
- ✓ 支持基于 UDS 的 Flash Bootloader。
- ✓ 支持 Windows、Linux 系统二次开发接口。
- ✓ 内置 120 欧终端电阻可软件配置。
- ✓ 可加载 TSMaster 所有收费 license。

### 1.3.2 技术参数

通道	1 *CAN
PC 端接口	高速 USB2.0
CAN 端接口	DB9
驱动	Windows, Linux 系统免驱设计, 具备极佳的系统兼容性
缓存	硬件缓存, 每个通道发送缓冲支持到 1000 帧 CAN
CAN	支持 CAN2.0A、B 协议, 符合 ISO11898-1 规范, 波特率 5Kbps—1Mbps
时间戳精度	1us, 硬件报文时间戳, 满足高阶需求
终端电阻	内置 120 欧终端电阻可软件配置
每秒发送报文*	最大 20000 帧/秒
每秒接收报文*	最大 20000 帧/秒
隔离	CAN 通道 DC2500V 隔离, 静电等级接触放电±8KV
供电	USB 供电
外壳材质	塑料
工作温度	-40℃~80℃
工作湿度	10% ~ 90% (无凝露)
工作环境	远离腐蚀性气体

\*单通道 1Mbps, 0 字节数据域情况

### 1.3.3 电气参数

参数		测试条件	最小值	典型值	最大值	单位
工作电压	USB 供电	四两 CAN 收发	5.10	5.12	5.14	V
工作电流	USB 供电	两路 CAN 收发	0.15	0.17	0.19	A
功率	USB 供电	两路 CAN 收发	0.77	0.87	0.98	W
CAN 接口	总线引脚耐压	CANH、CAHL	-42	--	42	V
	终端电阻	使能终端电阻	--	120	--	Ω
	隔离耐压	漏电流小于 1mA	2500	--	--	VDC

### 1.4 发货清单

✓ TC1001 主机



## 2. 硬件接口说明

### 2.1 接口说明

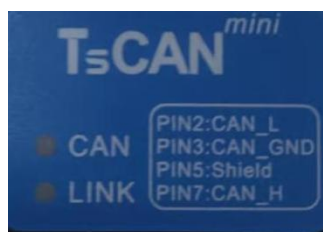


- USB 高速 2.0 接口;
- DB9 Male:

DB9 针脚	引脚	定义
	PIN2	CAN_Low
	PIN3	CAN_GND
	PIN5	CAN_Shield
	PIN7	CAN_High

### 2.2 LED 指示灯说明

指示灯实物图:



指示灯说明:

指示灯	定义
CAN1	CAN 通道指示灯
LINK	硬件连接指示灯

指示灯颜色说明：

颜色	描述
LINK 绿灯	设备硬件已连接
CAN 绿灯	CAN 通道数据帧发送或者接收正确
CAN 红灯	CAN 通道发送或接收错误帧，配置、协议或者接线错误

注：闪烁频率取决于总线负载。

## 3. 快速使用

### 3.1 下载安装 TSMaster 上位机

TSMaster 软件下载链接：[http://www.tosun.tech/TOSUNSoftware/TSMaster\\_Setup\\_beta.exe](http://www.tosun.tech/TOSUNSoftware/TSMaster_Setup_beta.exe)

若无法访问，可联系对应销售人员或登录同星官网获取上位机，亦可扫码关注公众号获取下载链接。



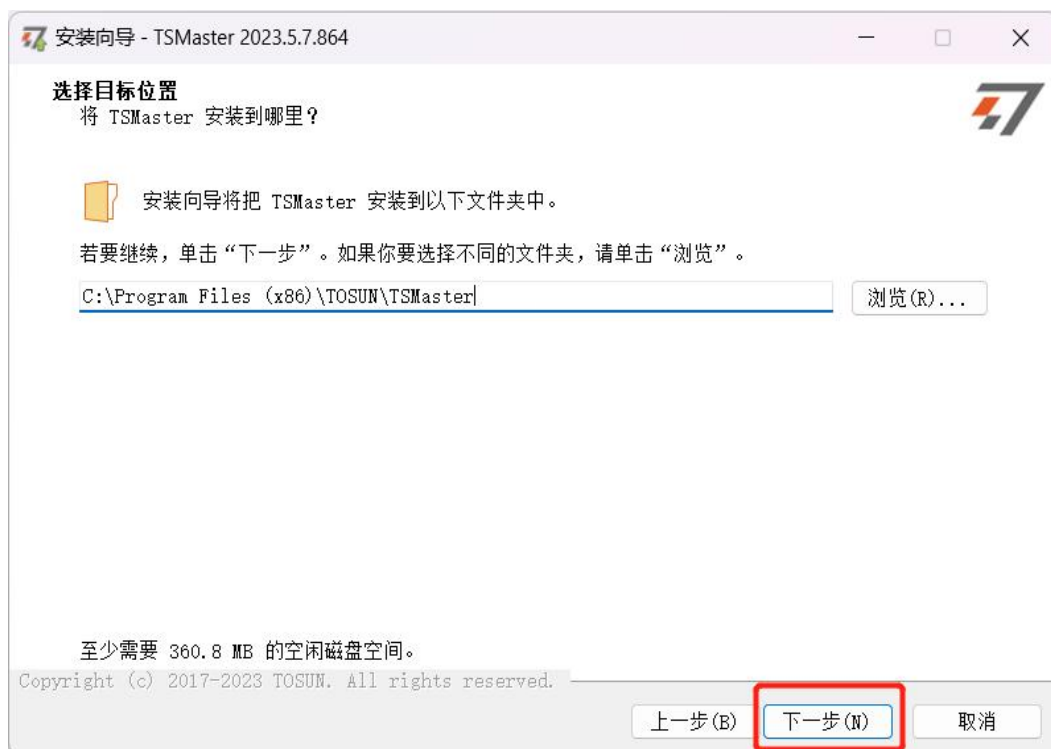
步骤一：



## 步骤二：

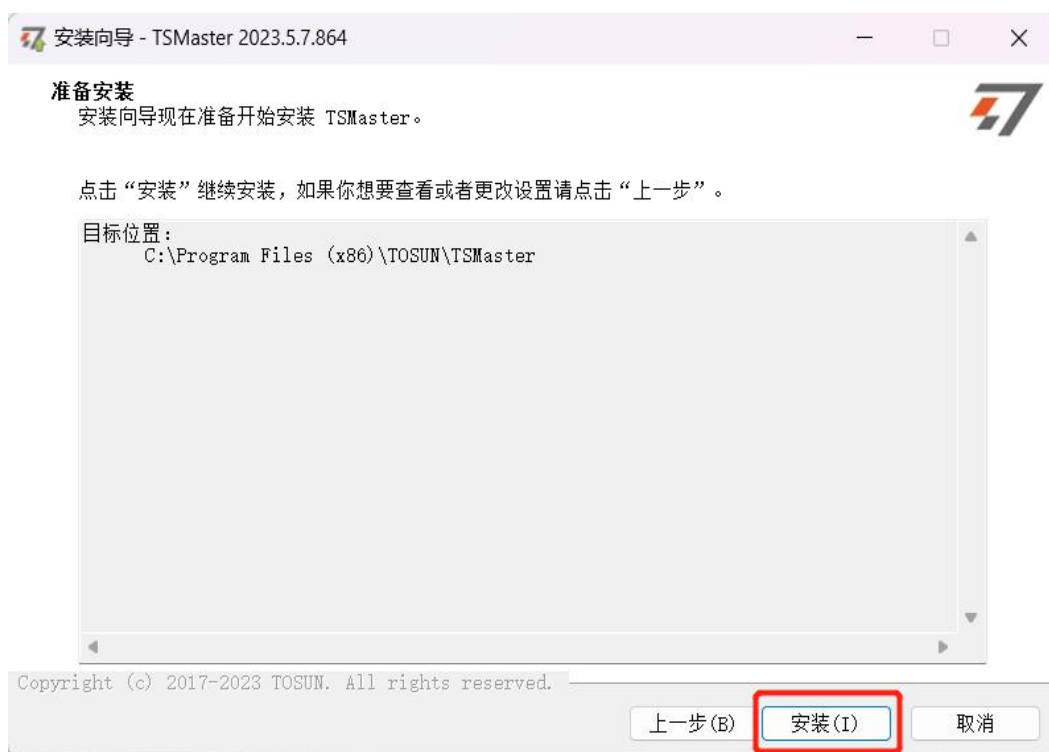


## 步骤三：





步骤四:



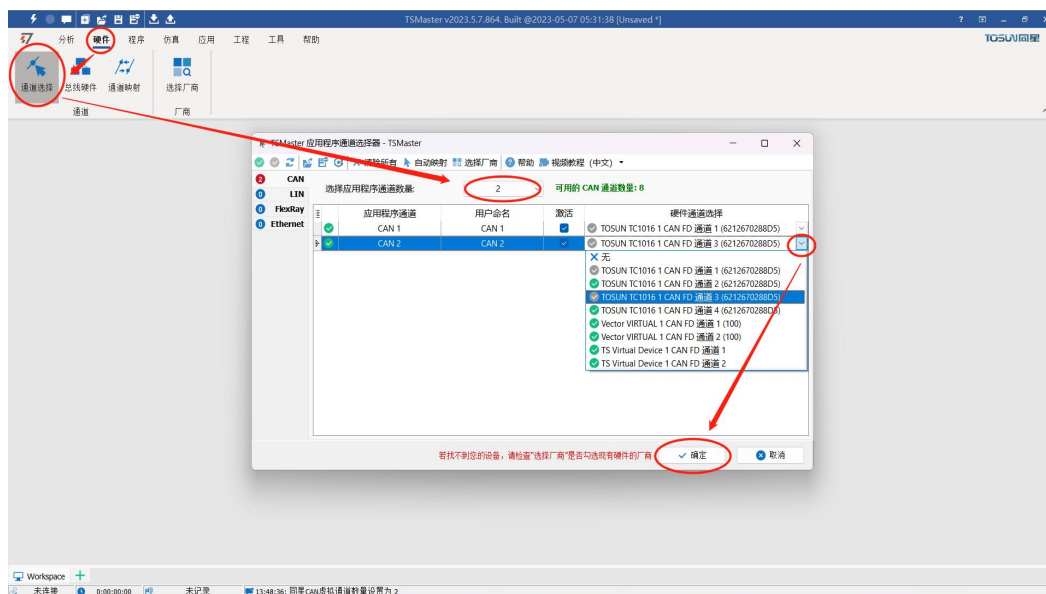
完成安装:



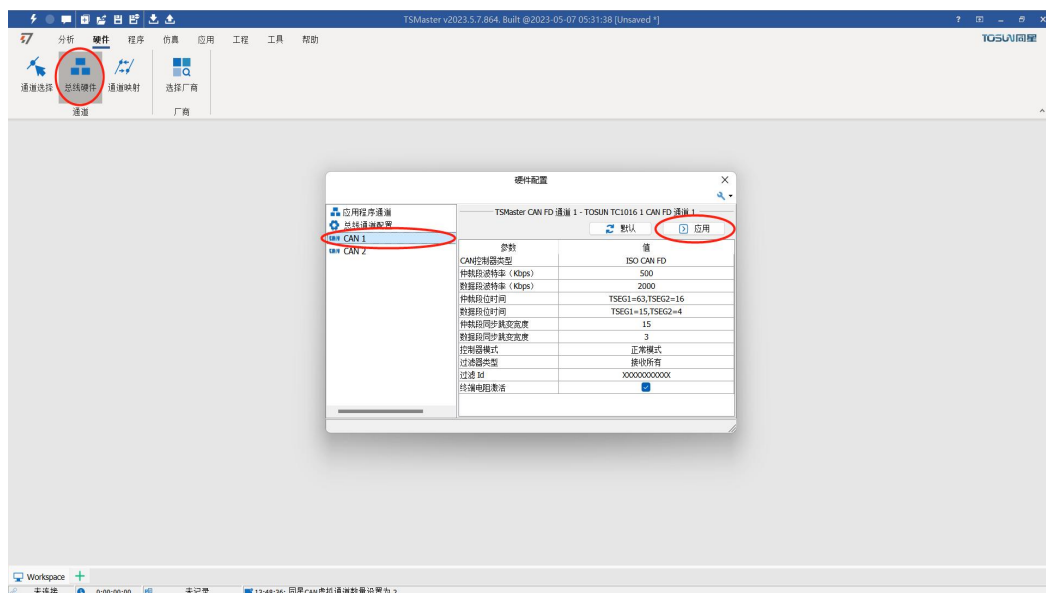
### 3.2 连接设备，配置通道

TOSUN 全系设备为免驱设计，无需下载驱动，直接连接即可。

在 TSMaster 软件界面：点击硬件-点击通道选择-下拉框选择通道数量-选择硬件通道-点击确定

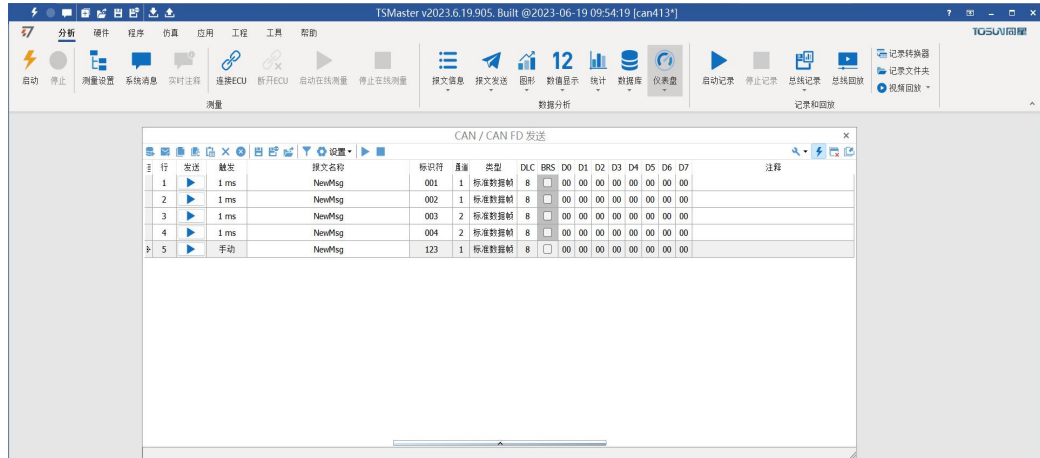


硬件配置中可切换 CAN/CANFD 协议，调整波特率以及开关终端电阻，配置完成后点击应用即可生效。



### 3.3 报文发送

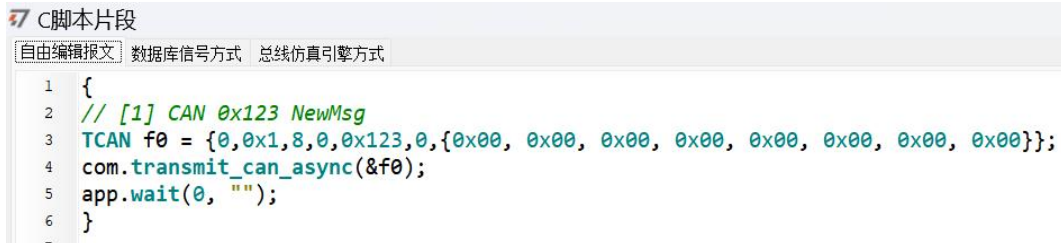
硬件连接完成，且软件进行配置之后，就可以实现报文发送的功能：



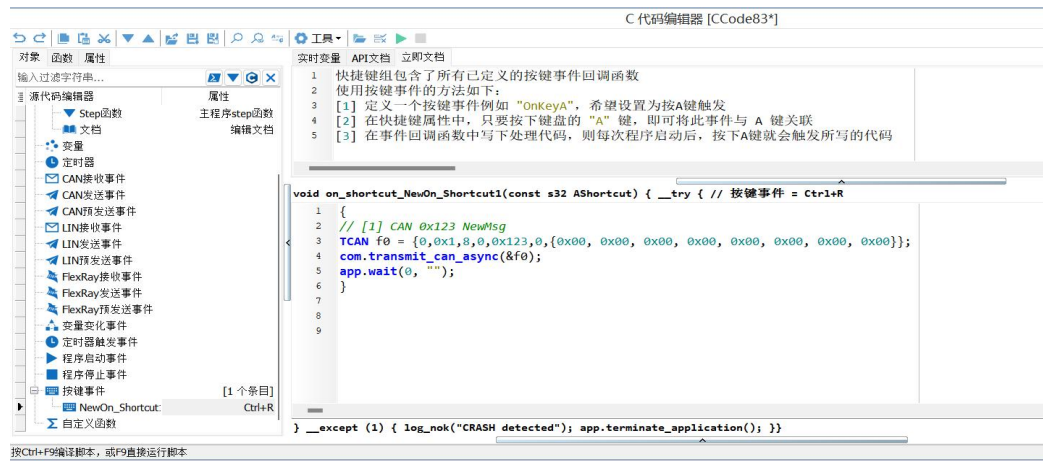
操作步骤：

- 报文发送-添加 CAN/CAN FD 报文发送
- 鼠标右键新建原始报文/从数据库添加报文，并对报文名称/标识符/通道等进行设置
- 报文触发设置，可手动触发/周期触发，周期触发可设置发送周期
- 报文信息右键可生成 C 脚本快捷添加到 C 小程序进行编程

以下为生成 C 脚本示例：

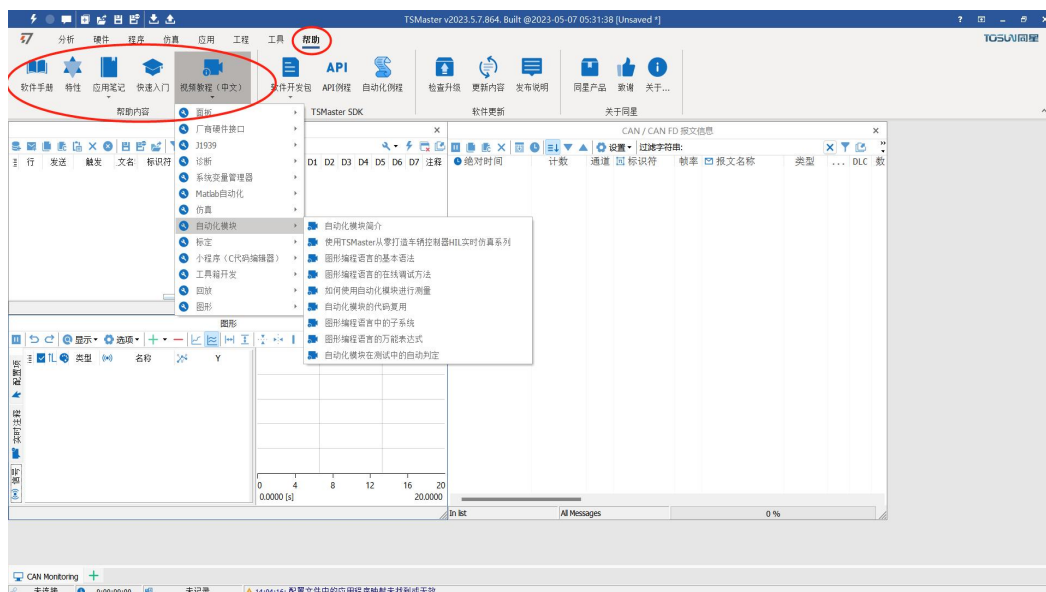


支持快捷复制粘贴到 C 脚本添加发送事件：



### 3.4 帮助文档及视频教学

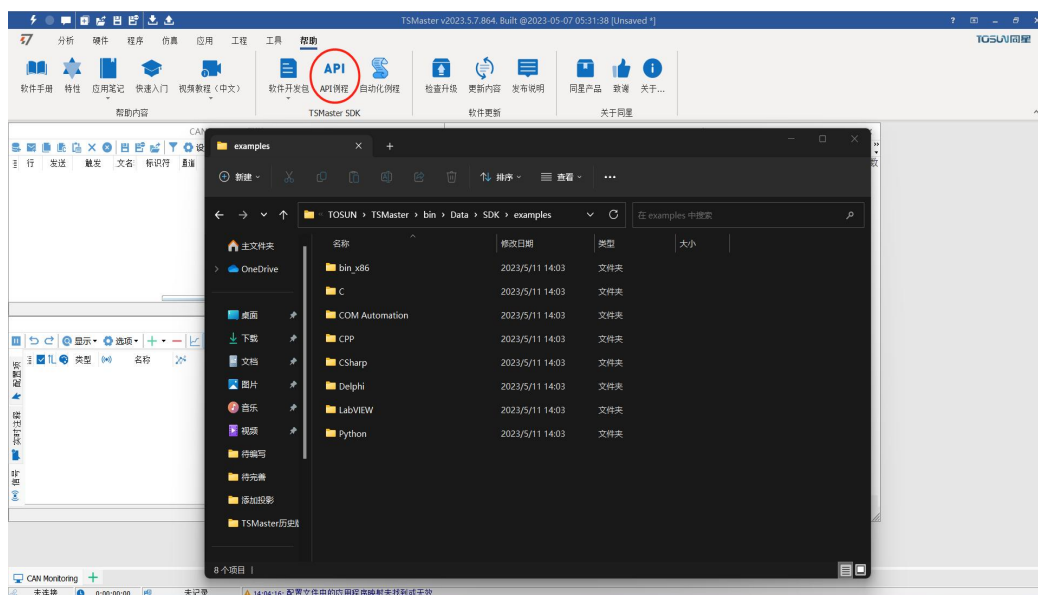
在 TSMaster 帮助栏中提供了多种的说明文档及帮助手册。



同时也提供了大量的教学视频，可进入 B 站 <https://space.bilibili.com/2042371333>，关注同星智能官方号，观看所有的教学视频。

### 3.5 TSMaster API 二次开发

在 TSMaster 帮助栏 API 例程中提供了多种常用语言的 API，方便用户二次开发。高效易用的二次开发函数，可支持各类开发环境，如 C, Python, C#, Labview 等。



### 3.5.1 Python 调用动态库

#### Windows32 位 Python:

- (1) pip install TSMasterAPI
- (2) 使用 TSMasterAPI form TSMasterAPI import \*
- (3) 示例同步上传 github,地址为:<https://github.com/sy950915/TSMasterAPI>. git

#### Windows64 位 Python / Li nux:

- (1) pip install libTSCANAPI
- (2) 使用 TSMasterAPI form libTSCANAPIimport \*
- (3) 示例同步上传 github,地址为: <https://github.com/sy950915/libTSCANAPI>. git

### 3.5.2 C 调用动态库

- (1) 在路径为 TSMaster\bin\Data\SDK\lib\x86 的文件中包含 TSMaster.h 头文件。

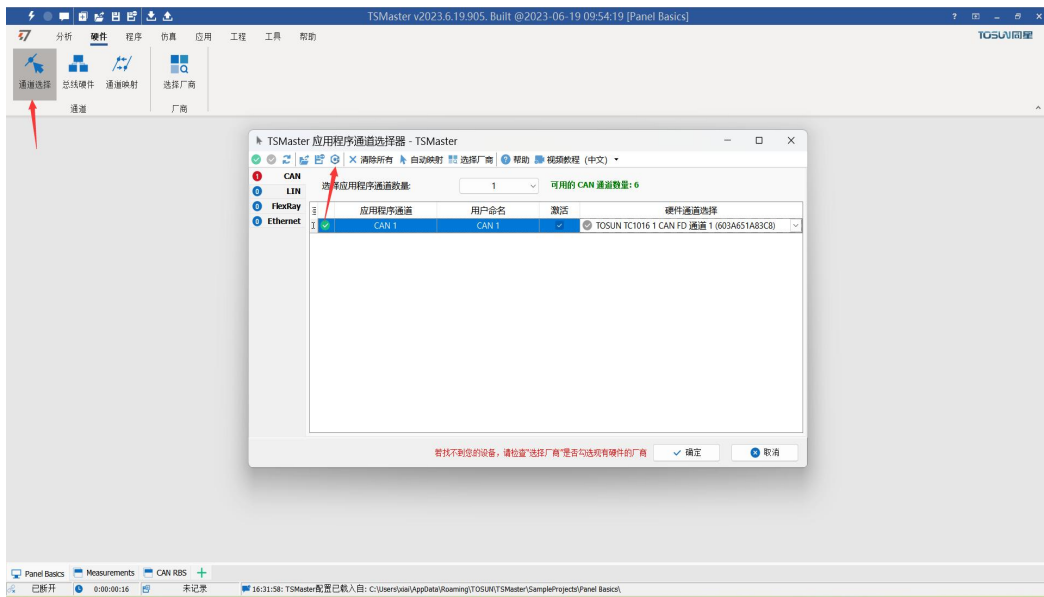
如: #include “TSMaster.h”

- (2) 在路径为 TSMaster\bin\Data\SDK\lib\x86 的文件中包含 TSMaster.lib 文件, 工程的连接器设置中连接到 TSMaster.lib 文件。

如: 在 C 环境下, 在项目属性页里的配置属性→连接器→输入→附加依赖项中添加 TSMaster.lib 文件。

### 3.5.3 调用接口示例

Windows、Linux 系统提供二次开发接口, 便于连接及使用设备。操作步骤为: 选择通道-生成 C 代码-使用 C 代码/python 代码来调用接口。以 C 代码为例:



C 脚本片段:

#### C脚本片段

```

1  initialize_lib_tsmaster("TSMaster");
2  tsapp_set_can_channel_count(1);
3  tsapp_set_lin_channel_count(0);
4  tsapp_set_flexray_channel_count(0);
5  tsapp_set_ethernet_channel_count(0);
6
7  TLIBTSMapping m;
8
9  // TSMaster CAN 通道 1 - TOSUN TC1001 1 CAN 通道 1
10 m.init();
11 sprintf_s(m.FAppName, "%s", "TSMaster");
12 sprintf_s(m.FHWDeviceName, "%s", "TOSUN TC1001");
13 m.FAppChannelIndex = 0;
14 m.FAppChannelType = (TLIBApplicationChannelType)0;
15 m.FHWDeviceType = (TLIBBusToolDeviceType)3;
16 m.FHWDeviceSubType = 3;
17 m.FHWIndex = 0;
18 m.FHWChannelIndex = 0;
19 if (0 != tsapp_set_mapping(&m)) { /* handle error */ };
20
21 if (0 != tsapp_connect()){ /* handle error */ };
22
23 /* do your work here */
24
25 tsapp_disconnect();
26 finalize_lib_tsmaster();
27

```

C 脚本调用函数说明:

initialize\_lib\_tsmaster("TSMaster");//TSMaster 初始化函数

tsapp\_set\_can\_channel\_count(1);//设置 can 通道数

tsapp\_set\_lin\_channel\_count(0);//设置 lin 通道数

tsapp\_set\_flexray\_channel\_count(0);//设置 flexray 通道数

tsapp\_set\_ethernet\_channel\_count(0);//设置 ethernet 通道数

TLIBTSMapping m;//初始化结构体

//设置 TSMaster CAN 通道 1 - TOSUN TC1001 CAN 通道 1 通道映射

m.init();//初始化结构体 m

sprintf\_s(m.FAppName, "%s", "TSMaster");//打印应用程序名字"TSMaster"

sprintf\_s(m.FHWDeviceName, "%s", "TOSUN TC1001");//打印硬件设备名

m.FAppChannelIndex = 0;//应用程序通道索引

```

m.FAppChannelType = (TLIBApplicationChannelType)0;//应用程序通道类型
m.FHWDeviceType = (TLIBBusToolDeviceType)3;//硬件设备类型
m.FHWDeviceSubType = 3;//硬件设备对应参数*
m.FHWIndex = 0;//硬件索引
m.FHWChannelIndex = 0;//硬件通道索引
if (0 != tsapp_set_mapping(&m)) { /* handle error */ };//如果返回值不等于0映射失败

tsapp_disconnect();//断开设备连接
finalize_lib_tsmaster();//释放 C 脚本模块

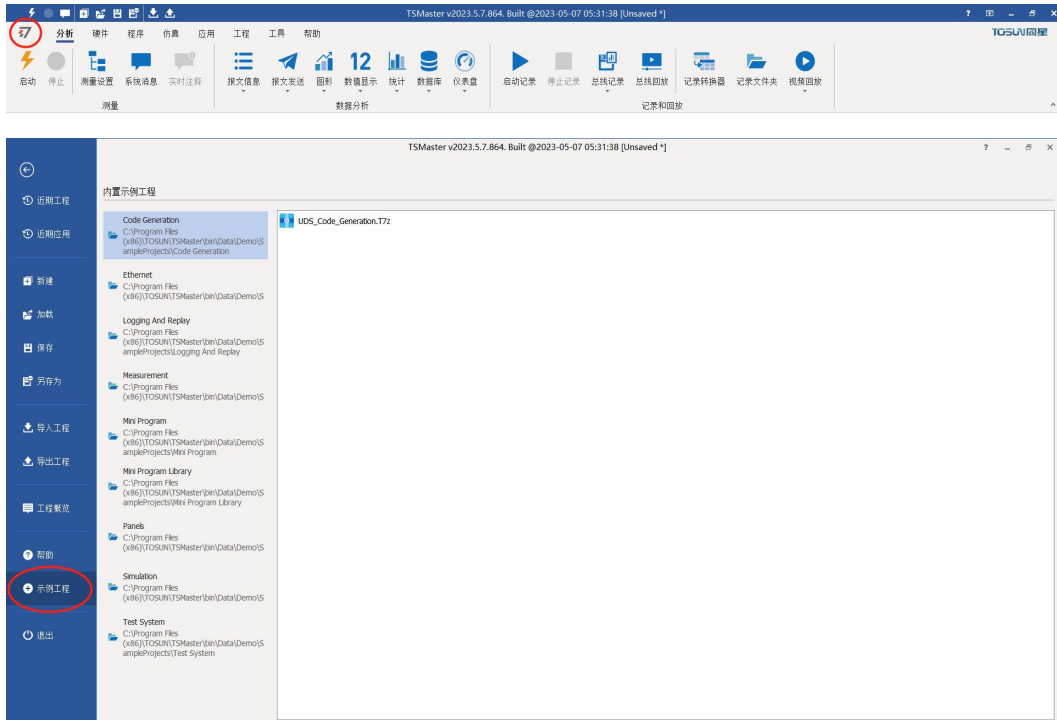
```

\*注：硬件设备对应参数可在 TSMaster-帮助-软件开发包查找对应参数：  
TSMasterAPI\_Hardware\_Map.pdf



### 3.6 示例工程

示例工程中提供了大量的 Demo 供用户参考，大大提高了用户的开发效率。



## 4. 检查和维护

TC1001 的主要电气部件是半导体元件，尽管它有很长的寿命，但在不正确环境下也可能加速老化，使寿命大打折扣。因此，在设备使用过程中应该进行定期检查，以保证使用环境保持所要求的条件。推荐每 6 个月到一年，至少检查一次。在不利的环境条件下，应该进行更频繁的检查。如下表，如果在维护过程中遇到问题，请阅读下面的内容，以便找到问题可能的原因。如果仍无法解决问题，请联系上海同星智能科技有限公司。

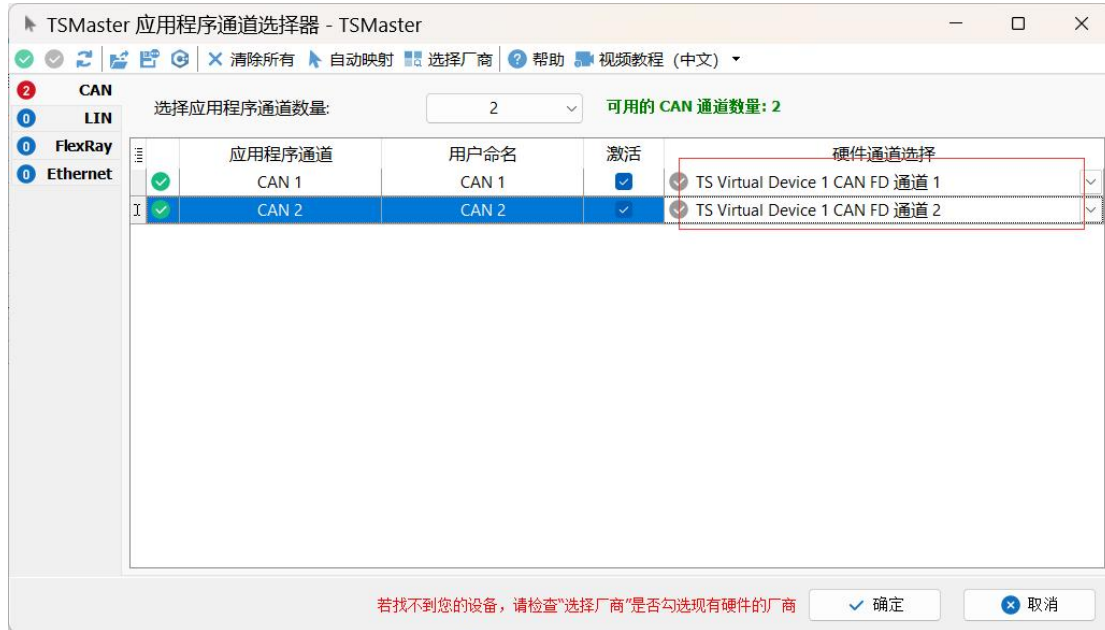
项目	检查	标准	行动
电源供应	在电源供应端检查电压波动	+12V DC	使用电压表在电源输入端检查源。采取必要措施使电压波动在范围之内
周围环境	检查周围环境温度 (包括封闭环境的内部温度)	-40°C~+80°C	使用温度计检查温度并确保环境温度保持在允许范围内
	检查环境湿度 (包括封闭环境的内部湿度)	没有空调时相对湿度必须在 10%~90%	使用湿度计检查湿度并确保环境湿度保持在允许范围内
	检查灰尘、粉末、盐、金属屑的积累	没有积累	清洁并保护设备
	检查水、油或化学喷雾碰撞到设备	没有喷雾碰到设备	如果需要清洁保护设备
	检查在设备区域中易腐蚀或易燃气体	没有易腐蚀或易燃气体	通过闻或使用一个传感器检查
	检查震动和冲击水平	震动和冲击在 规定范围内	如果需要,安装衬垫或其它减震装置
	检查设备附近的噪声源	没有重要噪声信号源	隔离设备和噪声源或保护设备
安装接线	检查外部接线中的压接连接器	在连接器间有足够的空间	肉眼检查如果有必要则调节
	检查外部接线的损坏	没有损坏	肉眼检查和如果有必须则替换接线



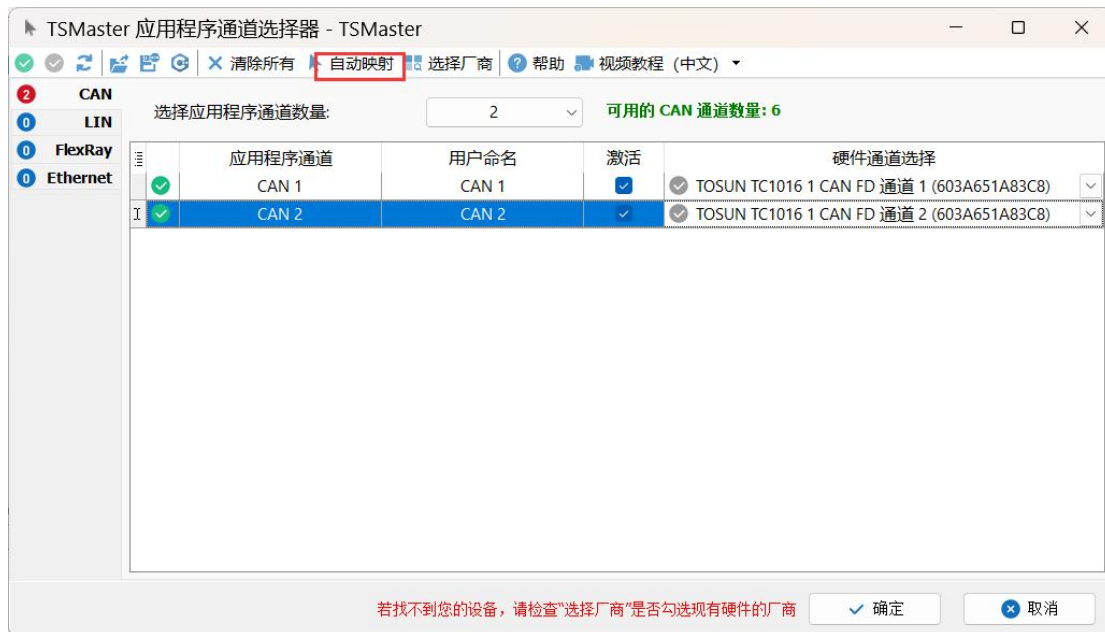
## 5. 常见问题及解答

### 5.1 线路连接正确但无法正常通信：

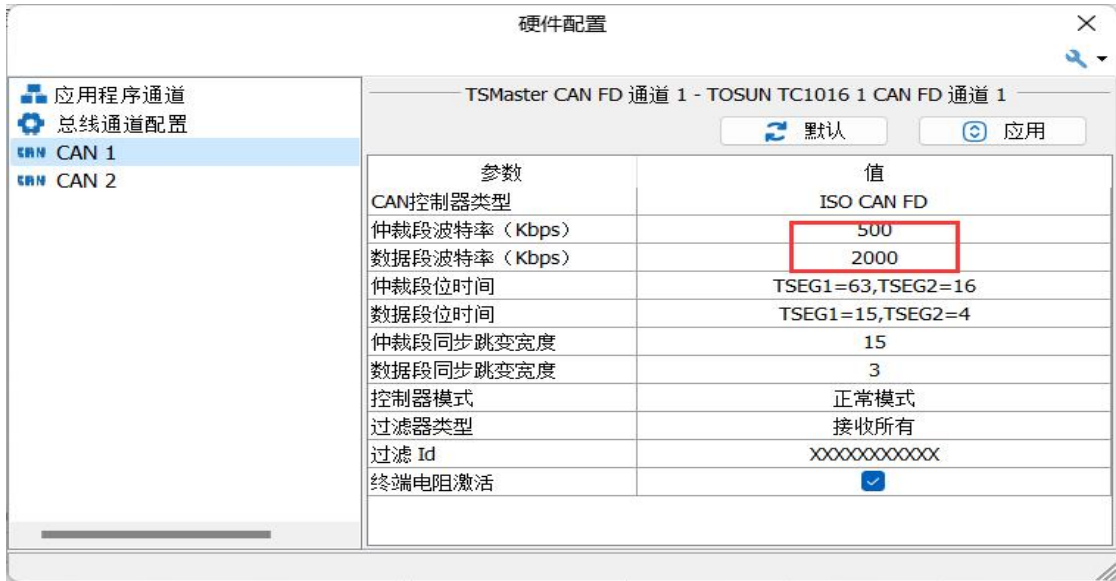
解决方法：检查一下是否设置了通道数。如果 CAN Channel Count = 0，当然无法显示在线硬件。且软件默认配置了虚拟通道，**需要选择硬件真实通道。**



可自动映射或者手动点击选择硬件真实通道：



若通道选择正确，则需要确保两个通道配置波特率一致才能正常通信，如下图所示：



## 5.2 报文观察不便以及信号过滤：

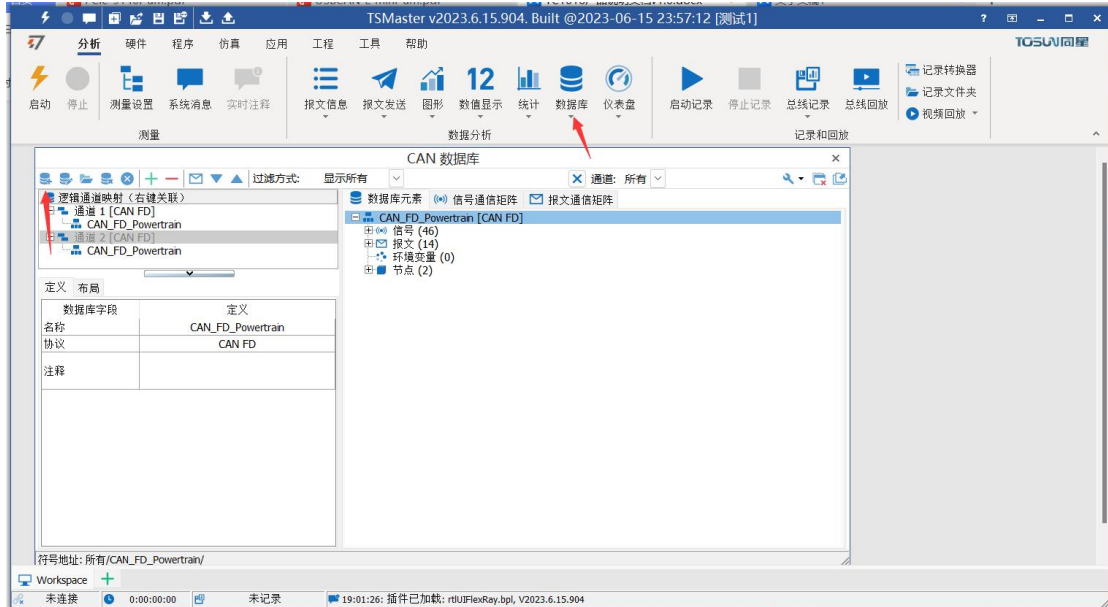


解决办法：按照固定显示或者时间顺序显示，展开或者折叠信号显示，以及过滤字符串，点击如下图标进行操作：



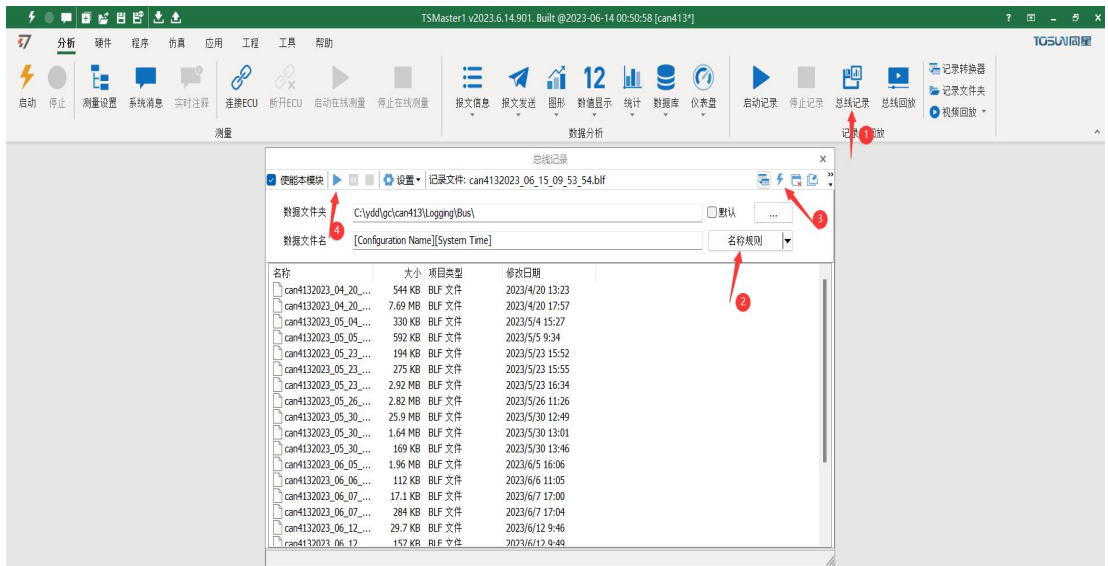
## 5.3 如何加载数据库:

数据库选择 can/lin/flexray 数据库，点击左上角图标添加数据库文件，或者把文件直接拖入此窗口自动加载，再点击左侧通道关联数据库即可。



## 5.4 如何自动记录报文:

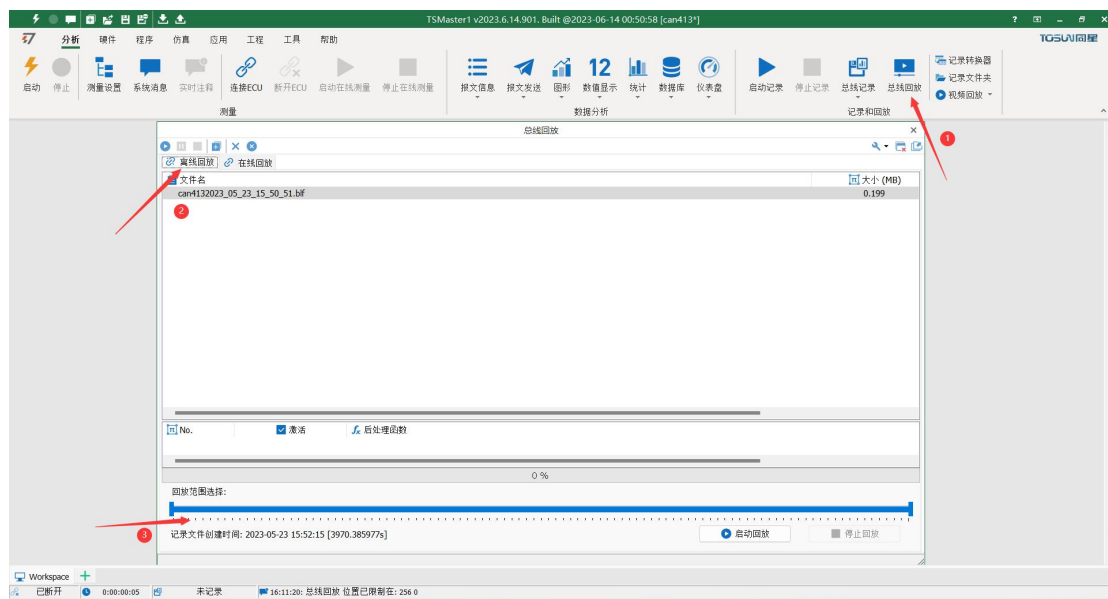
操作步骤:



- 分析-总线记录
- 添加名称规则以区别不同的保存文件
- 添加自启动功能
- 启动记录

## 5.5 如何回放报文（离线回放和在线回放）：

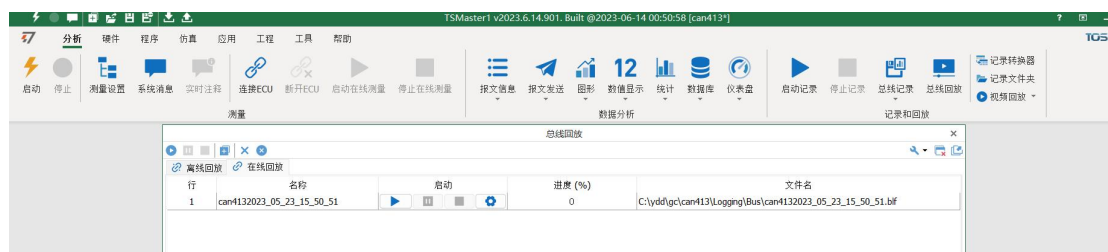
操作步骤：



a.总线回放

b.离线回放，添加需要回放的报文，可直接拖拽文件添加

c.选择报文回放范围，因报文显示窗口有数量限制，可选择自己所需的时间段报文



d.总线回放-在线回放-添加记录文件

e.在线回放可按照采集时间戳回放报文，且对回放数据进行设置



## 6. 附录 CAN2.0B 协议帧格式

### 6.1 CAN2.0 标准帧：

CAN 标准帧信息为 11 个字节，包括两部分：信息和数据部分。前 3 个字节为信息部分。

	7	6	5	4	3	2	1	0
字节 1	FF	RTR	x	x	DLC(数据长度)			
字节 2	(报文识别码)ID.10-ID.3							
字节 3	ID.2-ID.0		x	x	x	x	x	
字节 4	数据 1							
字节 5	数据 2							
字节 6	数据 3							
字节 7	数据 4							
字节 8	数据 5							
字节 9	数据 6							
字节 10	数据 7							
字节 11	数据 8							

字节 1 为帧信息。第 7 位 (FF) 表示帧格式，在标准帧中，FF=0；第 6 位 (RTR) 表示帧的类型，RTR=0 表示为数据帧，RTR=1 表示为远程帧；DLC 表示在数据帧时实际的数据长度。

字节 2、3 为报文识别码，11 位有效。

字节 4~11 为数据帧的实际数据，远程帧无效。

## 6.2 CAN2.0 扩展帧:

CAN 扩展帧信息为 13 个字节，包括两部分，信息和数据部分。前 5 个字节为信息部分。

	7	6	5	4	3	2	1	0
字节 1	FF	RTR	x	x	DLC(数据长度)			
字节 2	(报文识别码)ID.28-ID.21							
字节 3	ID.20-ID.13							
字节 4	ID.12-ID.5							
字节 5	ID.4-ID.0					x	x	x
字节 6	数据 1							
字节 7	数据 2							
字节 8	数据 3							
字节 9	数据 4							
字节 10	数据 5							
字节 11	数据 6							
字节 12	数据 7							
字节 13	数据 8							

字节 1 为帧信息。第 7 位 (FF) 表示帧格式，在扩展帧中，FF = 1；第 6 位 (RTR) 表示帧的类型，RTR=0 表示为数据帧，RTR=1 表示为远程帧；DLC 表示在数据帧时实际的数据长度。

字节 2~5 为报文识别码，其高 29 位有效。

字节 6~13 为数据帧的实际数据，远程帧无效。

## 7. 免责声明

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



## 汽车电子工具链，国产领导品牌

同星智能成立于2017年，一直专注于研发国产自主可控的汽车电子基础工具链产品，也是该领域国产领导品牌。

同星智能的核心软件TSMaster及配套硬件设备，具备嵌入式代码生成、汽车总线分析、仿真、测试及诊断、标定等核心功能，覆盖了汽车整车及零部件研发、测试、生产、试验、售后全流程。

全球企业用户超4000家，用户覆盖：汽车整车厂、零部件供应商、芯片厂商、设备/服务供应商、工程机械、航空航天及舰船军工等领域。



扫码关注  
获取软件下载链接

### 软件

- UDS诊断
- ECU刷写
- CCP/XCP标定
- 嵌入式代码生成
- 应用发布/加密发布
- 记录与回放
- 图形化编程
- 剩余总线仿真
- C/Python脚本
- 总线监控/发送
- SOMEIP和DoIP

### 硬件

- 1/2/4/8/12通道CAN FD/CAN转USB工具
- 1/2/6通道LIN转USB工具
- 10通道CAN FD/CAN转以太网工具
- 多通道Flexray/CAN FD转USB工具
- 多通道车载以太网/CAN FD转USB工具
- 车载以太网介质转换工具(T1转Tx)
- 多通道CAN FD/Ethernet/LIN记录仪



### 解决方案

- EOL测试设备
- FCT测试设备
- 汽车“四门两盖”试验解决方案
- 线控底盘测试解决方案
- 电机性能/耐久试验解决方案
- 新能源产线设备解决方案
- 总线一致性测试解决方案
- 信息安全解决方案