



# TOSUN-TX1000

## 用户手册

## 产品功能接口速览

产品名称	通道
TX1000	CAN FD *2
	差分线缆 * 1
	POD 板 * 1

### 版权信息

上海同星智能科技有限公司

上海市嘉定区嘉松北路 1288 号 9 号楼（总部）

曹安公路 4849 弄 14-17 栋（上海研究院）

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 2024-2025，上海同星智能科技有限公司。保留所有权利。

## 目录

产品功能接口速览 .....	2
1. 介绍 .....	4
1.1. 产品特性概览 .....	4
1.2. 电气参数 .....	5
1.3. 引脚定义 .....	6
1.4. LED 指示灯说明 .....	7
1.5. 系统要求 .....	9
1.6. 发货清单 .....	9
2. Windows 下的应用示例 .....	11
2.1. 硬件连接 .....	11
2.1.1. CAN .....	11
2.1.2. 系统连接 .....	12
2.2. TX1000 XCP 功能使用示例 .....	12
2.2.1. TX1000 STATION APP .....	13
2.2.2. TX1000 STATION GUI APP .....	15
2.2.3. TX1000 硬件配合 TSMaster 使用 .....	19
2.3. TX1000 XCP ON ECU 模式代码适配 .....	21
3. 附录 .....	24
3.1. CAN 浪涌保护器 .....	24
3.2. 软件安装 .....	24
4. 检查和维护 .....	29

# 1. 介绍

TX1000 是同星智能推出的高速标定设备，产品采用 RJ45 以太网接口与 PC 连接，Windows 系统免驱设计使得设备具备系统兼容性。配合功能强大的 TSMaster 软件，TX1000 支持连接 ARM/AURIX 系列芯片；支持的调试协议为 DAP（JTAG/SWD 暂未适配，后续增加）；支持修改 ECU 的 PC 指针跳转，程序暂停，恢复等；支持连接 ARM/AURIX 系列芯片，能够通过调试接口对 ECU 进行高速内存访问操作，不影响 ECU 代码运行。此外，TX1000 还可用于 XCP 协议标定 ECU。

产品配套资源包含：

- ✓ CAN FD 监控软件 TSMaster
- ✓ 可跨平台的二次开发库（提供单独编程手册）

## 1.1. 产品特性概览

### ➤ 设备

参数	说明
PC 接口	USB2.0 接口、RJ45 以太网
时间戳精度	微秒级别高精度时间戳
驱动	跨平台免驱动设计
接口针脚	标准 D-Sub, 9Pin
License	可加载 TSMaster 所有付费 License
继电器类型	磁保持继电器
供电	USB 供电，外部 DC (9~36 V)
功耗	4W
ESD 防护等级	接触放电等级 4 kV，空气放电等级 8 kV
外壳材质	金属
尺寸	约 178*113*38 mm
重量	约 420 g (无包装) /约 775 g (含包装)

工作温度范围	-40 摄氏度 ~ 80 摄氏度
工作湿度范围	10% ~ 90% (无凝露)

### ➤ CAN

参数	说明
CAN 连接标准	高速 CAN 连接 (符合 ISO11898-2 规范)
协议支持	全面支持 CAN 及 CAN FD 协议 (符合 ISO11898-1 规范)
CAN 波特率范围	从 125 Kbps ~ 1Mbps 可调
CAN 数据长度	最大长度支持 8 字节数据帧
CAN FD 波特率范围	从 125 Kbps ~ 8Mbps 可调
CAN FD 数据长度	最大长度支持 64 字节数据帧, 支持 BRS 帧类型
最大报文收发速率	最大发送帧率: 20000 帧/秒, 最大接收帧率: 20000 帧/秒 (单通道 1Mbps 0 字节远程帧)

## 1.2. 电气参数

### ➤ 电源特性参数

参数	测试条件	最小值	典型值	最大值	单位
工作电压	USB 供电	4.8	5.0	--	V
工作电流	USB 供电	--	0.6	--	A
功耗	USB 供电	--	3	--	W

### ➤ CAN 接口特性参数

参数	测试条件	最小值	典型值	最大值	单位
引脚耐压	CAN_H, CAN_L 对地	-58	--	58	V

参数	测试条件	最小值	典型值	最大值	单位
隔离耐压	漏电流 < 1mA	2500	--	--	VDC

### ➤ EMC（电磁兼容性）性能参数

测试项目	标准	测试条件	等级	单位
ESD	IEC 61000-4-2	接触放电	±4	kV
		空气放电	±8	kV
EFT	IEC 61000-4-4	群脉冲	±2	kV

### ➤ 机械尺寸

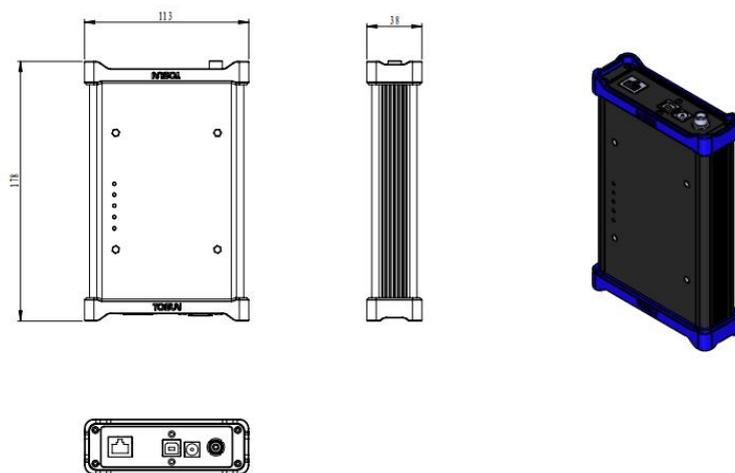


图 1-1 机械尺寸

## 1.3. 引脚定义



图 1-2 硬件接口

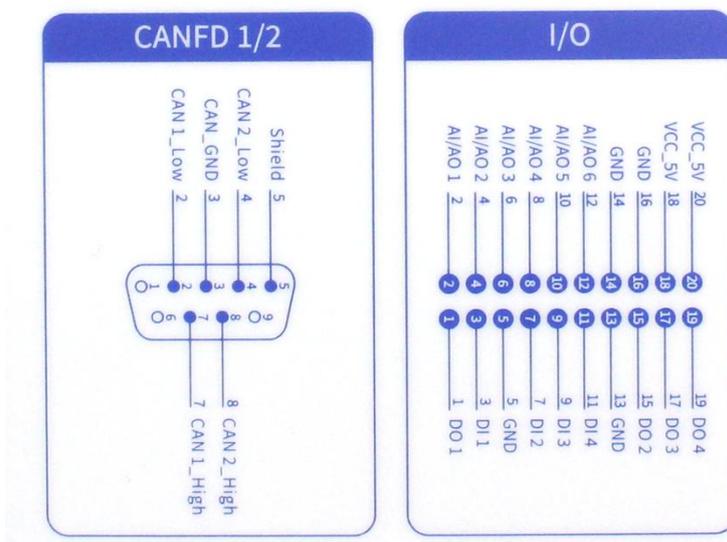


图 1-3 引脚定义

## 1.4. LED 指示灯说明



图 1-4 产品正面贴膜

### ➤ 指示灯说明

指示灯	定义
Power	电源指示灯
MCD	高速标定指示灯
Link	TX1000 设备连接上位机指示灯
CAN FD1	CAN FD 通道 1
CAN FD2	CAN FD 通道 2

### ➤ 指示灯颜色说明

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

## 1.5. 系统要求

### ➤ 计算机配备

- 操作系统 Windows 或 Linux
- 一个空闲的支持 USB 端口（支持 USB2.0），或支持自供电的 USB 扩展坞

### ➤ 驱动安装

- TX1000 设备采用免驱设计，具备极佳的系统兼容性，无需安装驱动即可直接使用

### ➤ 下载内容

- TSMaster 软件
- PDF 格式用户手册
- 可用于二次开发的编程库

## 1.6. 发货清单

设备/配件	数量	图片	标配/付费选配
TX1000 主设备	1		标配
USB 连接线	1		标配
DB9 母一分二公头信号线 (CAN)	2		标配

六类千兆网线	1		标配
差分线缆	1		标配
POD 板子	1		标配
12v2A 电源适配器	1		标配
CAN 浪涌保护器	--		付费选配



付费配件“CAN 浪涌保护器”相关说明请查看附录。

## 2. Windows 下的应用示例

### 2.1. 硬件连接

#### 2.1.1. CAN

使用设备配件“2 \* DB9 母一分二公头信号线 (CAN)”线束，可以通过两个独立的“D-Sub, 9Pin”连接器来访问两个通道。

下图为“DB9 母一分二公头信号线 (CAN)”线束的引脚对应图：

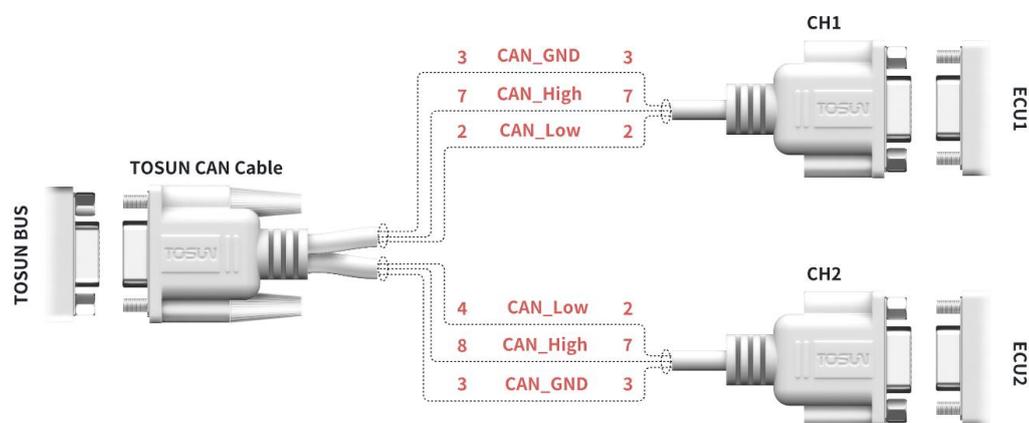


图 2-1 DB9 母一分二公头信号线 (CAN)

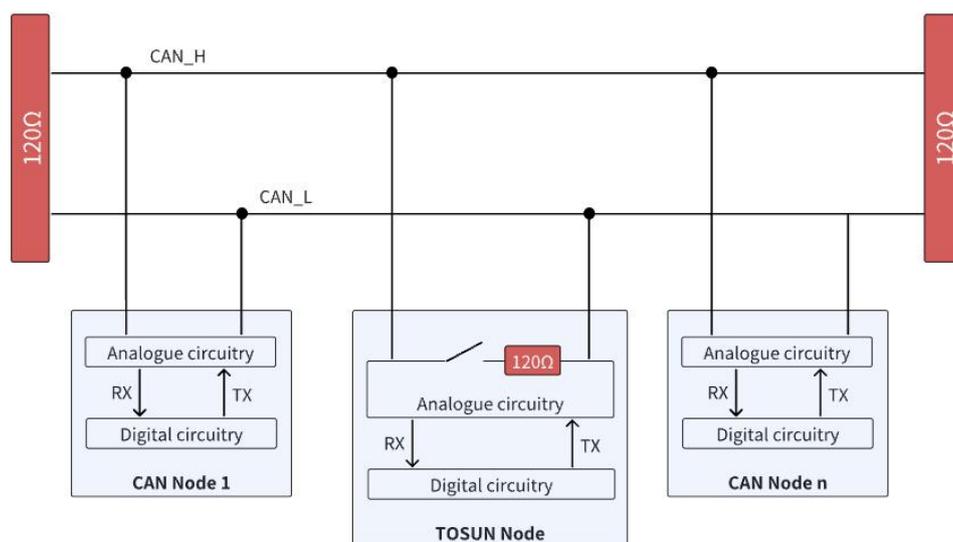


图 2-2 接入 CAN 总线

## 2.1.2. 系统连接

TX1000 设备通过以太网（RJ45）和电脑连接，TX1000 通过差分线缆和 POD 板连接，ECU 通过 JTAG/SWD/DAP 调试线缆和 POD 板连接。在 PC 端，TX1000 设备配合功能强大的 TSMaster 软件即可控制 TX1000 进行 XCP 测量和内存读取等操作。系统连接示意图：

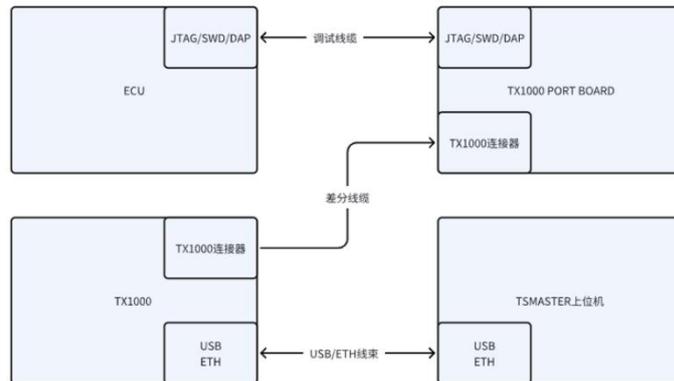


图 2-3 系统连接示意图

系统连接的实物图：



图 2-4 系统连接实物图

## 2.2. TX1000 XCP 功能使用示例

TX1000 XCP 功能一共分为两种模式：XCP ON ECU 和 XCP ON TX1000。

XCP ON ECU 模式是将同星的 XCP 协议栈代码嵌入客户 ECU 中编译，在 ECU 中配置协议栈参数、事件等。该模式相对稳定，并且可以有效保证数据的因果性。

XCP ON TX1000 模式是在 TX1000 内部运行 XCP 协议栈，优点是不需要客户 ECU 嵌

入任何代码，在 TX1000 内部就可以实现 XCP 标定测量等功能。启动该模式需要根据 A2L 文件编辑 xcp\_config.ini 文件，或者代码直接配置到 TX1000 中。此外，由于 XCP 代码运行在 TX1000 内部，线路延时、内存访问延时可能会导致部分情况数据不连续和周期不稳定等问题。

### 2.2.1. TX1000 STATION APP

“TX1000 STATION APP”是一个调用同星 TSdev 设备的 DLL 库实现的 STATION 示例程序，将 TX1000 的 XCP 报文转换为 XCP ON ETH 的报文，通过 TCP 作为 XCP 的传输层。STATION 程序启动后可以直接使用 TSMaster 中的 XCP ON ETH 连接 ECU。

```
SI_Error err = SI_FAIL;
err = ini.LoadFile("xcp_config.ini");
if (err != SI_OK) {
    printf("Failed to load config file! Error: %d, use default config value!\n", err);

    for (int i = 0; i < 6; ++i) {
        event_cycle_ms[i] = 0;
    }

    xcp_pars.addr_bits = 0;
    xcp_pars.checksum_type = 0;
    xcp_pars.daq_timestamp_unit = 0;
    xcp_pars.timestamp_size = 4;
    xcp_pars.enable_cal_pag = 1;
    xcp_pars.enable_daq = 1;
    xcp_pars.enable_get_comm_mode_info = 1;
    xcp_pars.enable_pgm = 1;
    xcp_pars.enable_slave_block_mode = 1;
    xcp_pars.enable_stim = 1;
    xcp_pars.id_type = 1;
    xcp_pars.max_cto_len = 255;
    xcp_pars.max_dto_len = 0xffff;
    xcp_pars.max_daq_cnt = 6;
    xcp_pars.max_event = 6;
    xcp_pars.max_odt_cnt = 20;
    xcp_pars.max_unlock_attempts = 3;
    xcp_pars.timestamp_size = 4;
}
else {
    for (int i = 0; i < xcp_pars.max_event; i++) {
        event_cycle_ms[i] = 0;
    }
}
```

图 2-5 XCP ON TX1000 配置参数 Demo

首先，初始化部分需要加载.ini 文件，加载失败则会使用默认配置，注意这个配置是我们提供的 Demo 工程的配置，如果没有.ini 文件，请注意手动配置。

```

uint32_t addr = (uint32_t)0x70004a64;
if (tsdev_err_ok == tsdev_api_tx1000_init(handle_p, 200000)) {
    printf("tx1000 init success\r\n");
    if (tsdev_err_ok == tsdev_api_tx1000_target_set_sync(handle_p, 0,
        tsdev_TC39x, 50000000, tsdev_DapProtocol, tsdev_DapStandardMode)
        )
    {
        puts("target set ok");
    }
    if (tsdev_err_ok == tsdev_api_tx1000_connect_sync(handle_p, 0))
    {
        puts("DebugPort Connect Finish");
    }
    if (1)
    {
        if (tsdev_err_ok == tsdev_api_tx1000_set_xcp_host_mode(handle_p, event_cycle_ms, &xcp_pars))
        {
            puts("DebugPort Connect Finish");
        }
    }
    else
    {
        uint32_t version;
        if (tsdev_err_ok == tsdev_api_xcp_set_mcu_ring_addr_sync(handle_p, 0x7000412c, &version))
        {
            printf("DebugPort Connect Finish version %x\r\n", version);
        }
    }
    if (tsdev_err_ok == tsdev_api_xcp_rx_event_register(handle_p, tsdev_xcp_rx_cmd_data_func))
    {
        puts("rx reg Finish");
    }
}

```

图 2-6 初始化 TX1000 设备为 XCP ON ECU /XCP ON TX1000

在 TSdev 设备扫描成功后，配置目标为 TC397，时钟频率 50 MHz，DAP 协议标准模式。然后使用 Connect API 连接 TC397。

如果使用 XCP ON TX1000 模式，运行 `tsdev_api_tx1000_set_xcp_host_mode` 函数配置 TX1000 内部 XCP 协议栈。

如果使用 XCP ON ECU 模式就运行 `tsdev_api_xcp_set_mcu_ring_addr_sync` 函数读取 ECU 内部 TOSUN XCP 协议栈的版本号，版本号有效后就可以进行标定/测量。

接下来，注册接收回调函数，这个已经实现。

```

std::cout << "客户端已连接: " << inet_ntoa(clientAddr.sin_addr) << " " << ntohs(clientAddr.sin_port) << std::endl;

struct _XCP_TRANS_LAYER_t data;
uint8_t* buffer = (uint8_t*)malloc(65536);
if(buffer != NULL)
{
    while (1) {
        iResult = sread(clientSocket, &data, 4);
        if (iResult != 1) {
            break;
        }
        iResult = sread(clientSocket, buffer, data.len);
        if (iResult != 1) {
            break;
        }
        else {
            if (tsdev_err_ok != tsdev_api_xcp_tx_cmd_data(handle_p, 0, buffer, data.len))
            {
                printf("tx xcp cmd failed\r\n");
            }
            else {
                printf("tx xcp cmd %x successful\r\n", buffer[0]);
            }
        }
    }
}
}

```

图 2-7 XCP ON ETH 报文解析并转发

最后，等待 TCP 连接成功后透传 XCP 命令给 TX1000 即可。

## 2.2.2. TX1000 STATION GUI APP

“TX1000 STATION GUI APP”是一个示例程序，能够将 TX1000 STATION APP 的一些配置项通过.ini 文件添加到可视化界面。XCP 的一些配置信息通过.ini 文件设置，确保与 A2L 文件匹配（仅在 XCP ON TX1000 模式下生效）。

TX1000 上位机软件的主要功能是作为 Socket 通信的服务端，监听 TSMaster 客户端的行为，同时也作为 TX1000 的上位机下发各项指令和配置参数。

软件自动扫描网卡，用户选择对应的网卡后连接设备，然后启动服务端，即可监听 TSMaster 的连接，用户在 TSMaster 端启动，即可建立通信。

### 2.2.2.1. 设备连接

进入设备连接界面，选择对应网卡，点击“连接设备”即可连接到 TX1000 设备。

保证上位机和 TX1000 在同一个网段，比如 TX1000 的 IP 为 192.168.1.10，掩码是 255.255.255.0 那么电脑的 IP 应该为 192.168.1.x, x 为 2-255 中的一个数，但是不包括 10。

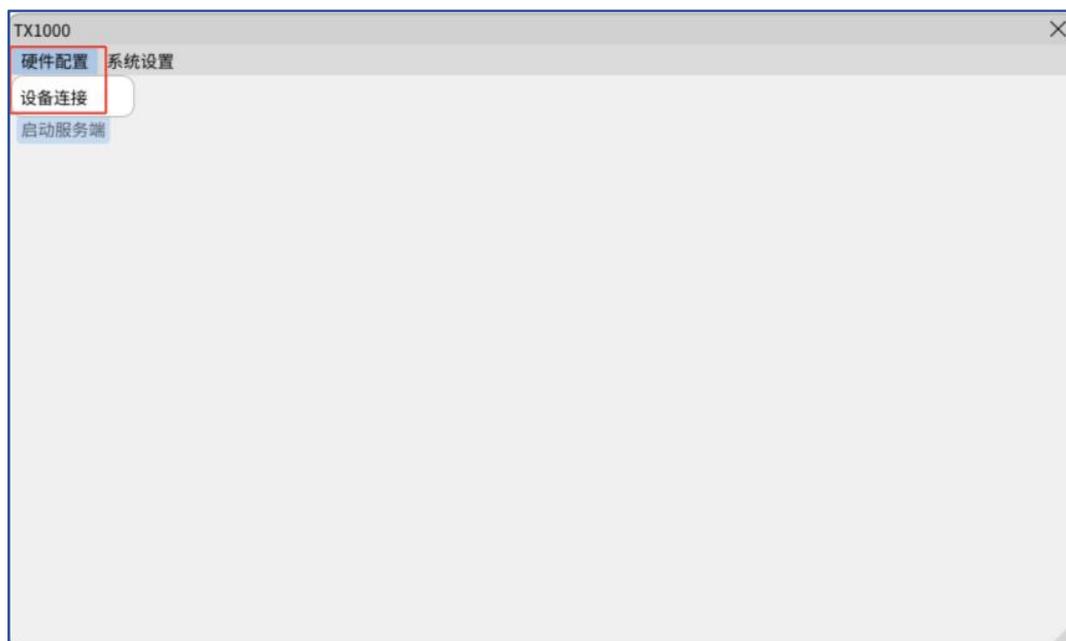


图 2-8 TX1000 主界面

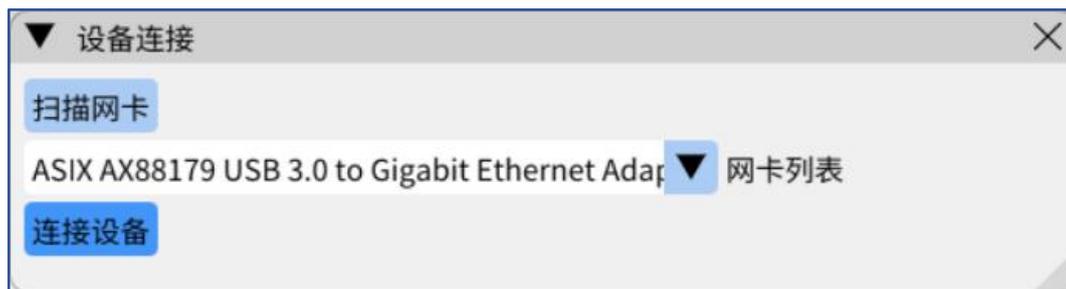


图 2-9 TX1000 连接设备界面

在主界面可以查看设备连接状态，点击“断开设备”，即可断开设备连接。



图 2-10 TX1000 连接成功界面

### 2.2.2.2. 配置参数

进入配置参数界面，可以设置各项参数，启动服务端后软件会下发这些配置给 TX1000 设备。

注 1：启动服务端后，中途修改参数不会影响已启动的配置。

注 2：软件在首次启动时，会自动加载 `xcp_config.ini` 文件里的配置参数。如果该文件不存在，则使用系统默认参数。

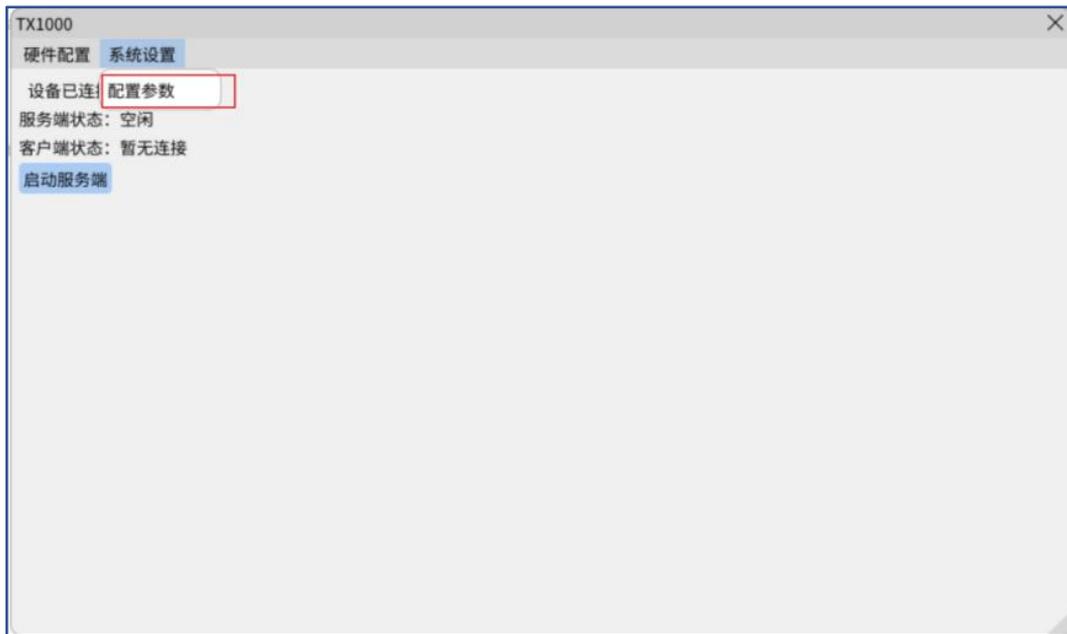


图 2-11 选择配置参数界面

该界面有三个按钮，功能介绍如下：

1. 加载配置文件参数：将 `xcp_config.ini` 文件中的各项参数加载到该界面；
2. 保存配置参数到文件：将该界面的各项参数保存到 `xcp_config.ini` 文件中；
3. 重置为默认参数：将该界面的各项参数重置为系统默认值，但不会自动保存到文件。



事件周期 (ms)	值
事件周期1 (ms)	100
事件周期2 (ms)	100
事件周期3 (ms)	100
事件周期4 (ms)	100
事件周期5 (ms)	100
事件周期6 (ms)	100

图 2-12 配置 XCP ON T1000 参数界面

### 2.2.2.3. 启动服务端

点击“启动服务端”，即可将配置参数下发给 TX1000 设备，并开始监听客户端，主界面会显示服务端和客户端的连接状态。



图 2-13 启动 XCP 服务器

### 2.2.3. TX1000 硬件配合 TSMaster 使用

1. 使用 TSMaster 需要提前启动 STATION APP 或者 STATION GUI APP，启动后打开 TSMaster 的 TX1000 Demo 工程。
2. 在 TSMaster 上测试 XCP:
  - 点击 TSMaster 中的应用，然后点击标定管理器。

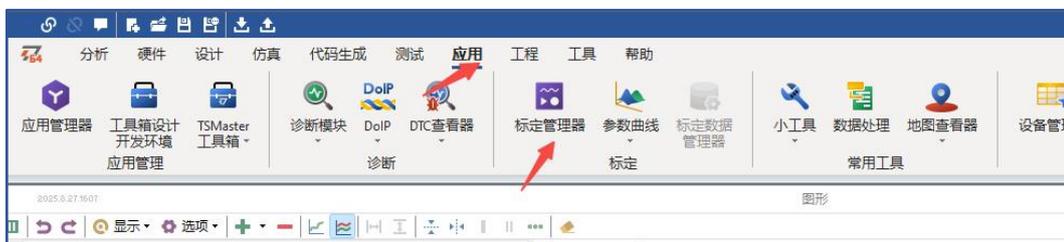


图 2-14 TSMaster 标定管理器

- 导入 A2L 文件并配置传输层为 ETH:ON TCP/IP STACK。



图 2-15 配置 XCP ON ETH

- 配置可选命令，建议取消勾选不使用的命令（未勾选的命令待适配，如有必要可以优先适配）。





图 2-18 配置为 DAQ 模式

- 启动 TSMaster 即可自动执行 XCP 测量, 添加了图形的信号可以直接看到信号的变化。

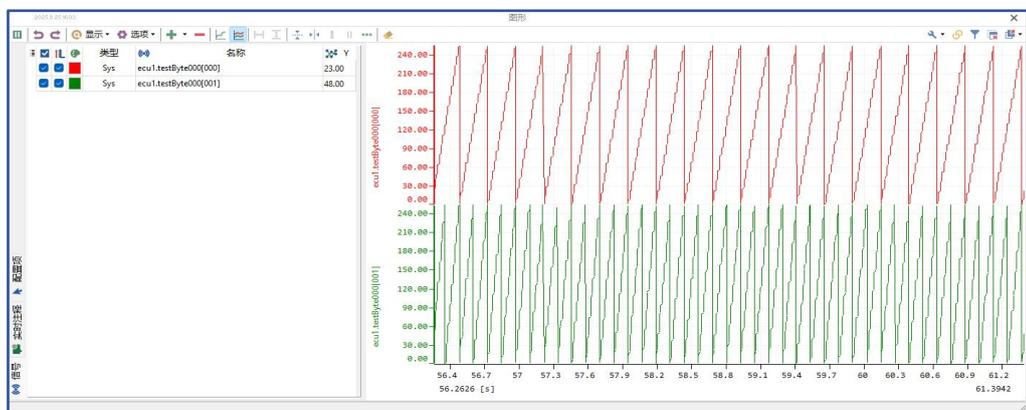


图 2-19 图形显示 DAQ 信号变化

### 2.3. TX1000 XCP ON ECU 模式代码适配

为了方便用户更快地使用 TX1000 设备, 我们提供了 TX1000 的使用说明。

- 首先, 将提供的代码包中的 01\_embedded\_source 添加到 ECU 编译路径中。

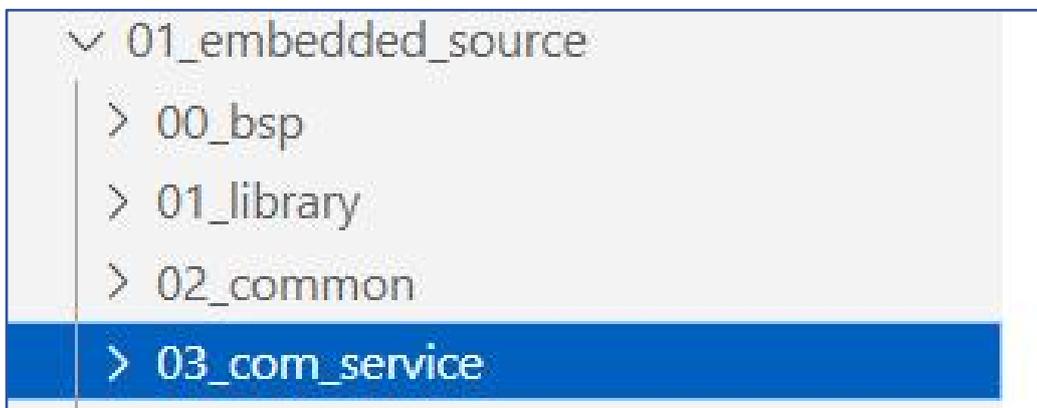


图 2-20 XCP 代码目录

- 然后, 参考提供的 xcp\_init.c 中的代码, 修改初始化配置。先修改 xcp\_ring

结构体的参数，包含 XCP 版本、XCP Tx/Rx 的缓冲区参数，再修改 XCP 协议栈使用的参数，主要包括使用的内存（分配两块连续内存，一块供协议栈使用，一块供 DAQ 使用）、内存、Seed、Key 算法等参数。update\_odt\_entry 函数供上传数据时查看（非必要）。

```
uint8_t tx_ring_buff[4096];
uint8_t rx_ring_buff[4096];
struct tosun_xcp_ring_t xcp_ring = {
    .info = {
        .xcp_version = 0x20252025,
        .xcp_tx_buff_len = sizeof(tx_ring_buff),
        .xcp_rx_buff_len = sizeof(rx_ring_buff),
        .xcp_rx_ring_buff_addr = (uint32_t)rx_ring_buff,
        .xcp_tx_ring_buff_addr = (uint32_t)tx_ring_buff
    },
    .var = {
        0,0,0,0
    }
};
struct tosun_xcp_init_pars_t xcp_pars={
    .tosun_xcp_ring_addr = (uint32_t)&xcp_ring,
    .event_pars = event_pars,
    .daq_max_use_mem = sizeof(daq_mem_pool_0),
    .daq_use_mem = daq_mem_pool_0,
    .ag = xcp_if_addr_granularity_8,
    .get_daq_timestamp = NULL,
    .max_event = 6,
    .xcp_use_mem = mem_pool_0,
    .xcp_max_use_mem = sizeof(mem_pool_0),
    .tosun_xcp_mem_read = tosun_xcp_mem_read,
    .tosun_xcp_mem_write = tosun_xcp_mem_write,
    .get_key = get_key,
    .get_seed = get_seed,
    .max_odt_cnt = 20,
    .max_cto_len = 0xff,
    .max_daq_cnt = 8,
    .max_dto_len = 0xff,
    .update_odt_entry = update_odt_entry
};
```

图 2-21 XCP 模块初始化参数 Demo

- 将初始化的参数传入 tosun\_xcp\_api.h 中的 tosun\_xcp\_setup 函数，函数返回非空即代表初始化完成。返回值指针用于作为 tosun\_xcp\_event\_handle 处

理。

```
},  
void* tosun_xcp_setup(struct tosun_xcp_init_pars_t* init_pars_p);  
void tosun_xcp_handle();  
void tosun_xcp_event_handle(void* xcp_p, uint16_t event_idx);
```

图 2-22 XCP 模块 API

- 初始化结束后即可开始标定和测量。根据需求将 `tosun_xcp_handle()` 函数放在周期性调用函数中。具体调度周期可以根据标定/测量信号自行修改。针对 `xcp_event`，只需要在具体的观测量位置添加 `event_handle` 函数即可实现 DAQ 功能，简易伪代码：

```
void*xcp_p = NULL;  
struct tosun_xcp_init_pars_t init;  
void task_10ms(){  
    uint16_t event_idx=0;  
    uint16_t daq_val=0;  
    xcp_p = tosun_xcp_setup(&init);  
    {  
        while(1){  
            delay_ms(10);  
            tosun_xcp_handle();  
            daq_val++;  
            tosun_xcp_event_handle(xcp_p,event_idx);  
        }  
    }  
}
```

图 2-23 XCP 伪代码示例

## 3. 附录

### 3.1. CAN 浪涌保护器



图 3-1 CAN 浪涌保护器

CAN 浪涌保护装置是用于保护 CAN 总线系统的设备,防止浪涌(瞬间过电压或过电流)对其造成损害。此装置无需外接电源,采用 DB9 接口设计,兼容性好,安装简便,能够即插即用,不影响通信质量。上述产品中,不支持浪涌保护的型号,您可以通过加装同星的 DB9 浪涌保护装置,型号为 TCA00011,来达到 CAN 通道浪涌保护的目。

#### ➤ CAN 浪涌保护器

参数	参数值
尺寸	约 76 * 38 * 25 mm
重量	约 71 g
浪涌保护等级	±2 kV

### 3.2. 软件安装

本章节介绍在 Windows 下安装 TSMaster 软件到计算机上的步骤。

#### ➤ TSMaster 软件下载

<https://www.tosunai.com/downloads/>

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。



图 3-2 TOSUN 公众号二维码

➤ 软件安装

1. 双击 TSMaster 软件安装包，选择安装语言，点击“确定”。

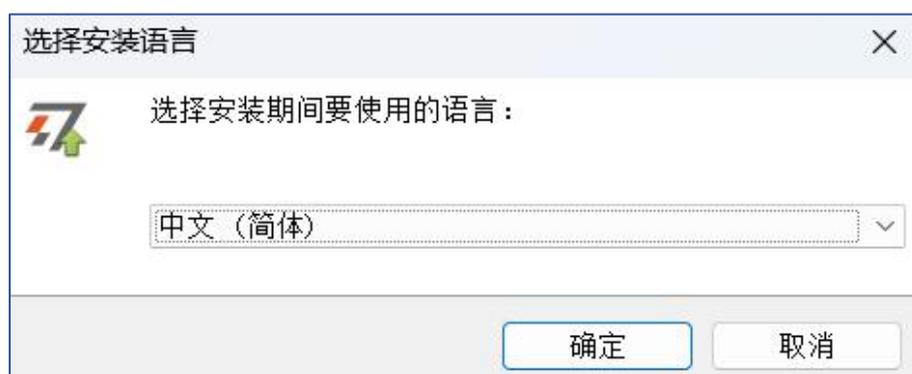


图 3-3 TSMaster 安装

2. 选择“我接收协议”，点击“下一步”。

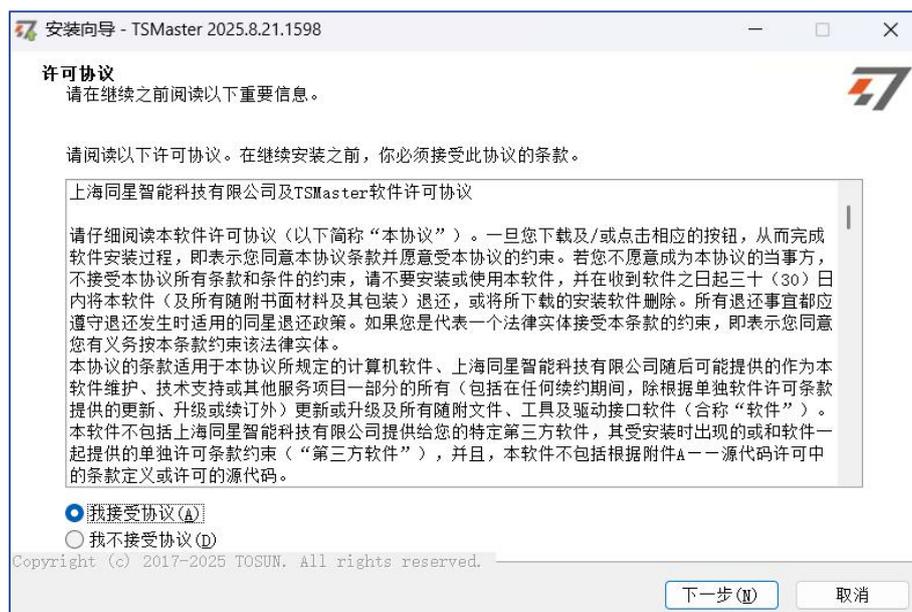


图 3-4 TSMaster 安装

3. 选择安装目录，点击“下一步”。

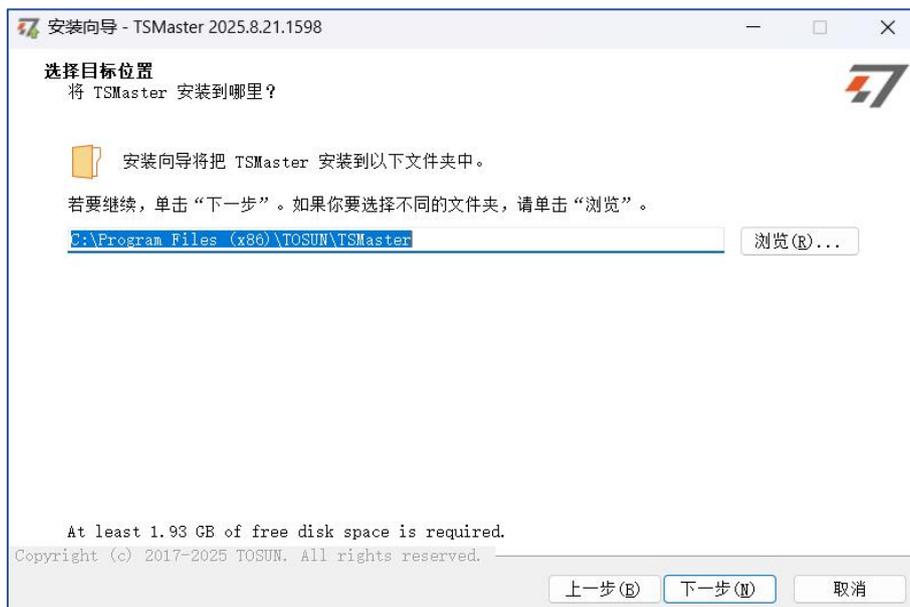


图 3-5 TSMaster 安装

4. 按需选择附加任务，点击“下一步”。

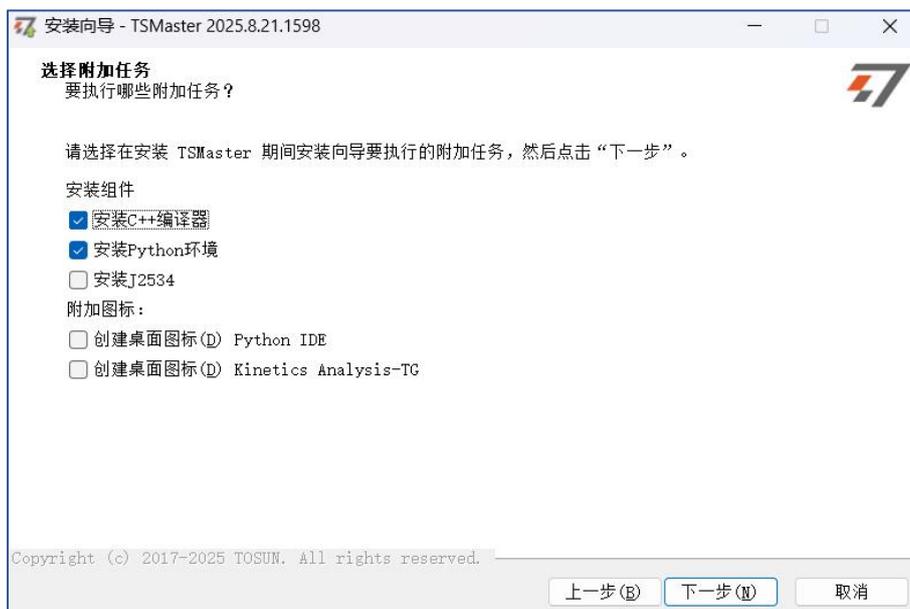


图 3-6 TSMaster 安装

5. 点击“安装”。

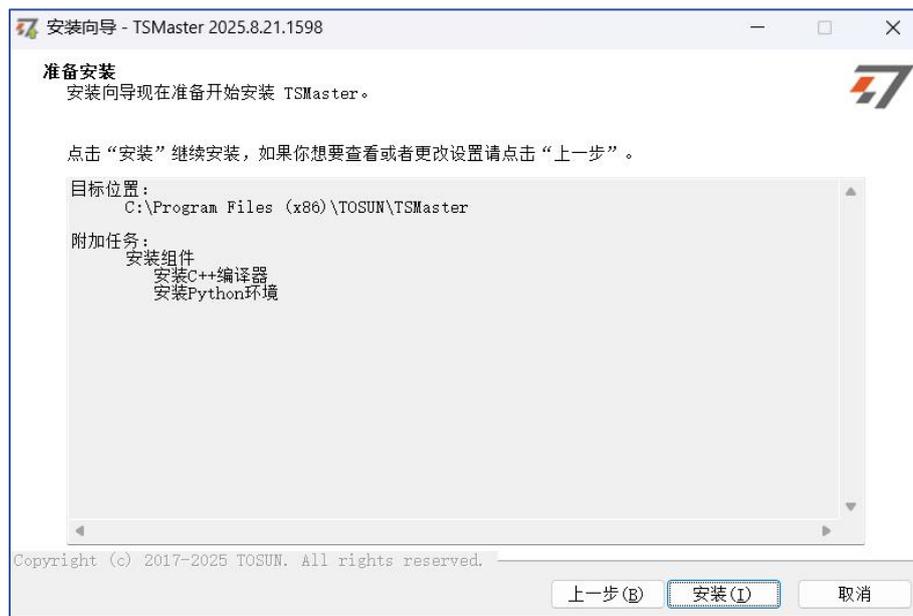


图 3-7 TSMaster 安装

## 6. 等待片刻。

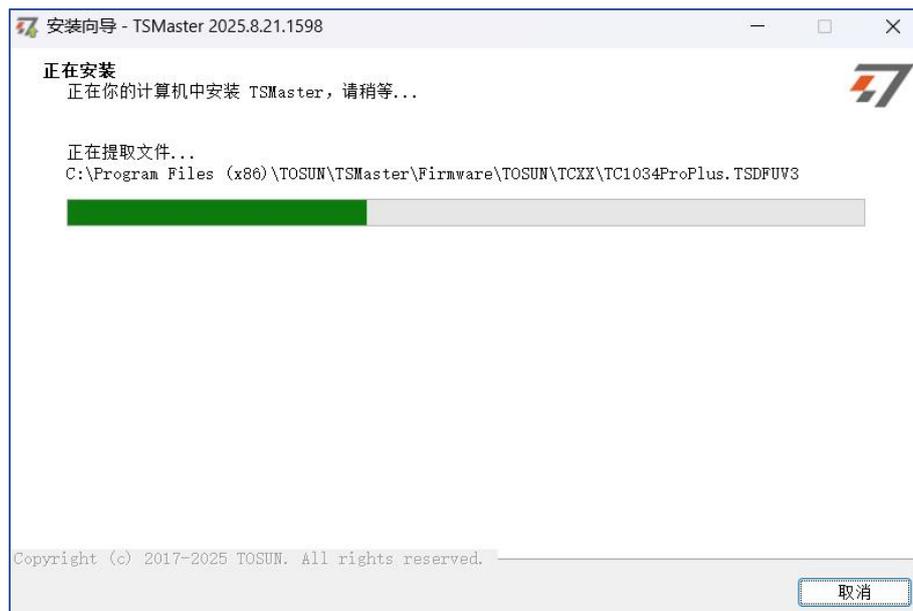


图 3-8 TSMaster 安装

## 7. 点击“完成”完成安装。



图 3-9 TSMaster 安装

## 4. 检查和维护

TX1000 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 1 年，至少检查 1 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。

### ➤ 电源环境检查

项目	检查内容	标准/允许范围	行动/措施
电源 供应	在电源供应端检查电压波动	USB 端口： +5V DC	使用 USB 功耗仪/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
		电源端口： +12V DC	
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的范围内
	检查环境湿度 (包括封闭环境的内部湿度)	相对湿度： 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

### ➤ 污染与防护检查

项目	检查内容	标准/允许范围	行动/措施
污染	检查灰尘、粉末、盐、金属屑的积累	无积累	清洁并保护设备
	检查水、油或化学喷雾溅射到	无液体溅射	如果需要，清洁并保护设

项目	检查内容	标准/允许范围	行动/措施
	设备		备
危险气体	检查易腐蚀或易燃气体	无此类气体	通过嗅觉或使用传感器检查

### ➤ 机械应力与噪声检查

项目	检查内容	标准/允许范围	行动/措施
机械应力	检查震动和冲击水平	震动和冲击在规定的范围内	如果需要，安装衬垫或其他减震装置
电磁环境	检查设备附近的噪声源	无重要噪声信号源	隔离设备与噪声源，或对设备采取屏蔽保护措施

### ➤ 安装与接线检查

项目	检查内容	标准/允许范围	行动/措施
接线	检查外部接线中的压接连接器	连接器间有足够间隔空间	目视检查，如有必要则进行调整
	检查外部接线的损坏	无损坏	目视检查，如有必要则更换线缆

## 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 / SOME/IP和DoIP / 自动化测试



扫码关注  
获取软件下载链接



## 硬件

1/2/4/8/12通道CAN FD/CAN转USB/PCIe工具  
 1/2/6通道LIN转USB/PCIe工具  
 多通道FlexRay/CAN FD转USB/PCIe工具  
 多通道车载以太网/CAN FD转USB/PCIe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  
 TTS测试系统(通信板卡、数字/模拟量板卡等)



## 解决方案

总线一致性 / 网络自动化测试系统 / 充电测试系统  
 EMB标定测试设备 / 信息安全解决方案  
 FCT/EOL测试设备 / 线控底盘测试解决方案  
 汽车“四门两盖”试验解决方案  
 电机性能 / 耐久试验解决方案



## 关于我们

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。

国际组织  
ASAM, CiA

质量保证  
ISO9001:2015

CE认证

## 愿景

解决一切工程难题!

### 联系我们

021-59560506  
marketing@tosunai.cn

### 访问官网

www.tosunai.com

