

Atech
UDS 上位机使用手册
By Software Team
Version 2.9

uds.exe

Version: 2.8.4.0

ATECH Engineering BCM
2024-10-31

Code: 2017-05-DOC001		Content:	
Scope of Application: Only for developers of Atech Software system.			
Document Sensitivity: Internal document ,only for developers			
Reference:			
Maintainer:			
Name	Position	Department	Email Address
Hui.Liu	Software engineer	ATECH Engineering BCM	hui.liu@atech-automotive.com
Change Log:			
Time (YYYY/MM/DD)	Versi on	Modifie r	Change Log
2017/05/28	V1.0	Hui.Liu	draft
2017/08/26	V1.1	Hui.Liu	1、 修改 CAN、LIN 选择方式 2、 添加 ECU 快捷命令工具 3、 添加 ECU 输入输出控制工具 4、 添加数据流工具解析信号 5、 添加 C#脚本工具
2017/09/13	V1.2	Hui.Liu	1、 添加 C#脚本支持基本控件 2、 添加 C#脚本示例 3、 添加页码
2018/01/01	V1.3	Hui.Liu	1、 稳定考虑，移除 C#脚本工具 2、 添加 ECU 变量周期读取工具
2018/02/01	V1.4	Hui.Liu	1、 重新添加 C#脚本工具 2、 C#脚本工具添加部分窗体控件支持
2018/04/30	V1.5	Hui.Liu	1、 添加 CAN_FD 支持 2、 配置文件添加传输层配置 N_AS/N_BS 等
2018/07/11	V1.6	Hui.Liu	1、 配置文件增加传输层 BlockSize 配置
2018/08/22	V1.7	Hui.Liu	1、 增加安全算法添加和编译方法

2020/06/12	V1.8	Hui.Liu	<ul style="list-style-type: none"> 3、增加安全算法支持大于 4 字节 4、Simulation 工具增加 TransmitAppend 功能，用于添加 CheckSum 功能 5、C#脚本工具增加 tx_append 功能 6、修复 LIN 诊断是收到 78 会停发 3D 的 bug 7、修复流控帧在功能寻址模式下显示为 7DF 的 bug 8、增加变量读写工具支持 MPC5744 GHS 9、添加对 64 字节 CANFD 传输层的支持
2020/12/11	V1.9	Hui.Liu	<ul style="list-style-type: none"> 1、增加 2A、2C 服务 2、增加 memory 工具支持解析 elf 格式文件 (*.elf*.out) 3、优化脚本工具，删除编辑器 4、修复传输层发送 64 字节失败下次重发会发送 8 字节的 bug 5、增加驱动选择功能，支持 Kvaser Pro、Vector CANoe、CANalyst-II
2021/12/18	V2.0	Hui.Liu	<ul style="list-style-type: none"> 6、修改 IO 控制界面放大缩小未同步的问题 7、修改 Memory 工具读写支持设置地址和长度格式固定为 4 字节 8、增加驱动支持 PCAN、Value CAN3(仿 581)、TOSUN
2022/02/23	V2.1	Hui.Liu	<ul style="list-style-type: none"> 1、添加 DoIP 支持 2、.net 运行库从 4.0 修改为 4.6
2022/07/13	V2.2	Hui.Liu	<ul style="list-style-type: none"> 1、release 2.4.1 2、修复 23 工具配置成 4 字节对齐地址发送的数据错误的问题 3、修复同星通道使用错误导致崩溃的问题 4、修改 CANOE 驱动支持多窗口同时打开一个通道 5、修改 CANOE 驱动支持 VN5620 6、修改 CANOE 驱动 XL_API 升级到 11.0 7、修改 CANOE 驱动通道配置 4 个 8、修复 CANFD DLC 错误问题 9、修复收到长度小于 8 字节的诊断响应可能会导致工具异常的问题 10、修复 CAN 显示时间戳不对的问题
2022/08/27	V2.3	Hui.Liu	<ul style="list-style-type: none"> 1、release 2.4.3 2、修复前几个版本 LIN 诊断无法显示数据流的问题 3、修复 Doip 模式下 27 不能自动回复的问题 4、优化 Doip 数据流显示和添加 Doip 断线自动重连功能 5、添加 LIN 传输层支持功能寻址

			6、LIN 添加 KeepOnline 功能 7、修改配置文件中 LIN 的 Tx Id、Tx Fun Id、Rx Id 配置为 NAD
2023/08/07	V2.4	Hui.Liu	1、修复 23 工具配置成 4 字节对齐地址发送的数据错误的问题 2、修复同星通道使用错误和临界区问题导致崩溃的问题 3、修复前几个版本 LIN 诊断无法显示数据流的问题 4、添加 CANOE 驱动支持 VN5620 5、修改.net 运行库版本为 4.8 6、添加 Doip 支持 7、添加 LIN 传输层支持功能寻址 8、LIN 添加 KeepOnline 功能 9、增加 Obd 相关配置和支持 10、修复传输层收到 block 不等于 0 的流控制未回复流控的问题 11、修复 LIN 诊断 27 自动认证无效的问题 12、修复多次开关 TP 后，3E80 重复发送的问题 13、添加解析 hex 文件（Version_MemMap）
2024/03/04	V2.5	Hui.Liu	1、添加 ZLG CAN 驱动支持 CANFD 模式 2、修复上个版本 LIN 无法发送多帧的问题 3、修复安全算法 dll 收到的 key 超过 16 字节会报载入失败的问题 4、传输层增加 CANFD 模式下 8 字节配置，默认 64 字节，兼容之前的传输层 5、Uds 工具增加\$Fd 64 Byte 配置选项
2024/04/05	V2.6	Hui.Liu	1、修复周立功驱动问题，接收超过 100 帧会导致内存越界访问的问题 2、添加 Doip 支持 V3 版本（\$DoIP ProtocolVersion） 3、添加配置文件签名功能 4、添加 uds 自动检查更新和下载功能
2024/04/05	V2.7	Hui.Liu	1、release 2.8.0 2、添加支持 OemSpecific 字段 3、添加接入认证 dll 接口 4、增加上位机和安全算法 dll 自动更新功能 5、增加算法 dll 调用验证，拒绝执行非法的 dll 文件 6、增加 Doip 版本支持 03（ISO13400-2019） 7、优化连接时兼容性，先读取车辆声明，再进行 doip 连接 8、优化 Memory 工具解析后加载到界面的速度

			9、优化升级检测时间 10、优化周立功驱动，增加终端电阻和采样点配置 11、优化同星驱动，将同星 Api 升级到 2023.10.978，需要安装 TSMaster_Setup_alpha+_4_12.exe 12、修复 2E 配置工具超过 64 字节报错的问题 13、修复 Doip 模式下，空消息不显示错误 14、修复 Doip 模式下，27 不自动回复的问题 15、修复传输层发送 7 字节的功能寻址会报错的问题 16、修复周立功驱动 bug
2024/07/23	V2.8	Hui.Liu	1、更新同星使用注意事项
2024/10/31	V2.9	Hui.Liu	1、release 2.8.4 2、添加 SecurityConstant 配置支持，/// Constant: + Value (hex) 3、添加授权即将过期提醒和提前重新更新 license（仅在内网自动获取） 4、修复 1903 读取快照号解析错误的问题 5、更新同星使用注意事项（针对 TsMaster 版本和部署域控无管理员权限使用说明）

Index

Chapter I 文档说明.....	8
Chapter II 软件安装.....	9
1. 安装说明:	9
2. 安装方法:	9
Chapter III 驱动安装.....	10
1. UDS 驱动安装方法:	10
2. 支持的 UDS 驱动:	10
2.1. Kvaser Pro.....	10
2.2. Vector CANoe.....	10
2.3. CANalyst-II.....	12
2.4. PEAK PCAN.....	12
2.5. Value CAN3	12
2.6. TOSUN TsMaster.....	13
2.7. Zlg USBCAN.....	15
2.8. iTEK USBCAN-II.....	15
Chapter IV UDS 上位机.....	16
1. 主界面.....	16
1.1. 菜单界面.....	16
1.2. 诊断设置.....	17
1.3. 诊断服务.....	17
1.4. 诊断数据流.....	17
1.5. 主窗体设置.....	18
2. 工具箱.....	19
2.1. ECU 快捷命令工具.....	19
2.2. ECU 变量读写工具.....	19
2.3. ECU 软件配置工具.....	22
2.4. ECU 输入输出控制工具.....	23
2.5. CAN 信号模拟工具	23
2.6. CAN 信号监视工具	25
2.7. CAN 数据流工具	26
2.8. C# 脚本工具	27
Chapter V 安全算法	30
1. 算法说明.....	30
1.1. 安全算法不大于 4 字节.....	30
1.2. 安全算法任意字节.....	30
2. 添加方法:	30
2.1. cFree5 编译.....	30
2.2. VC++编译.....	30
Chapter VI config.ini 文件配置.....	32
1. 文件说明:	32
2. 格式说明:	32
Chapter VII message.ini 文件配置.....	38

1. 文件说明:	38
2. 格式说明:	38
Chapter VIII DBC 生成 message.ini	40
Chapter IX 联网和授权	41
1. 联网功能:	41
2. 授权功能:	41
2.1. 未检测到授权	41
2.2. 添加授权	42

Chapter I 文档说明

本文档主要介绍 UDS 上位机功能和使用方法，上位机绝大部分功能均建立在 ISO14229 和 ISO15795 诊断协议之上。默认上位机使用者已具备相关知识，故诊断细节方面不再描述。

参考规范如下：

ISO 14229-2006.pdf

ISO 14229-1-2020.pdf

ISO 15765-2 2004.pdf

ISO 13400-2 2012.pdf

ISO 13400-2 2019.pdf

J1979_201702.pdf

J1979DA_201702.xlsx

Chapter II 软件安装

1. 安装说明:

本软件采用 C# .net 编写，所以运行需要安装.net 4.8 运行环境，如果未安装此环境请自行下载安装。

[下载 .NET Framework | 免费官方下载 \(microsoft.com\)](#)

2. 安装方法:

无需安装，解压压缩包到任意目录，可以直接运行 uds.exe

Chapter III 驱动安装

1. UDS 驱动安装方法：

将 UDS 驱动文件夹放到 `uds.exe` 相同目录即可。

2. 支持的 UDS 驱动：

2.1. Kvaser Pro

2.1.1 Kvaser Pro 驱动安装方法：

安装 `kvaser_drivers_setup.exe`，推荐版本：5.22.392.0

注意事项：kvaser 驱动需要依赖 VC++ 运行库（2010vcredist_x86.exe），地址为
\\192.168.8.20\软件安装包\CAN\UDS Test\2010vcredist_x86.exe

2.1.2 Kvaser Pro 驱动配置方法：

不需要配置

2.1.3 UDS 驱动文件列表：

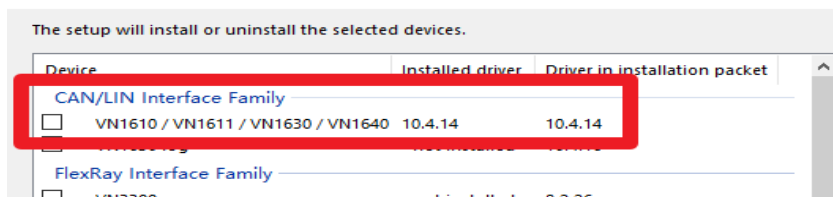
文件名	版本号
canlibCLSNET.dll	8.22.392.0
linlib.dll	8.22.392.0
linlibCLSNET.dll	8.22.392.0
uds_driver.dll	2.6.0.0
uds_driver.ini	

2.2. Vector CANoe

2.2.1 Vector CANoe 驱动安装方法：

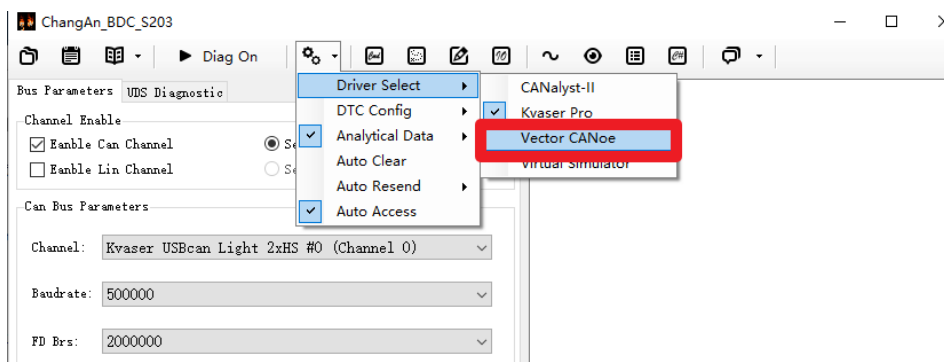
安装 VectorDriverDisk 中 CAN/LIN Interface Family 驱动

Driver Selection

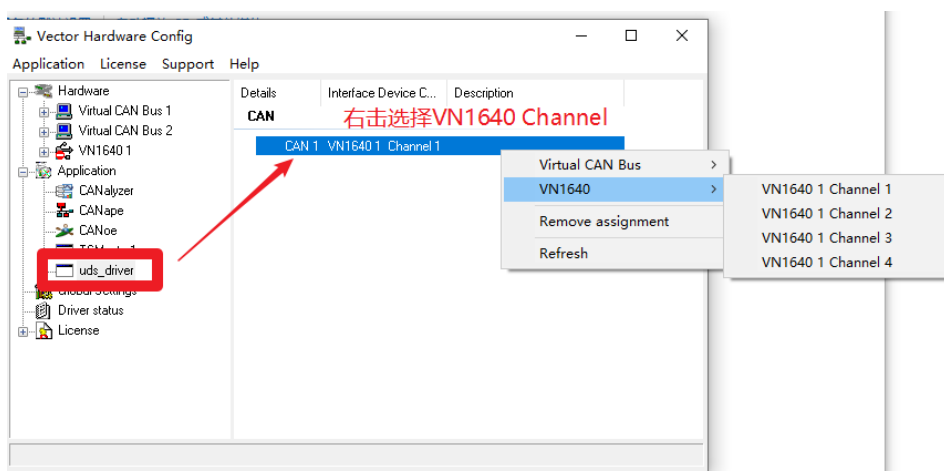


2.2.2 Vector CANoe 驱动配置方法:

- 打开 uds 上位机，选择 Vector CANoe，重启上位机后点关闭



- 打开控制面板->设备和声音->Vector Hardware



2.2.3 UDS 驱动文件列表:

文件名	版本号
vxlapl NET.dll	11.0.14.0
uds_driver.dll	1.0.7.0

2.3. CANalyst-II

2.3.1 CANalyst-II 驱动安装方法:

无

2.3.2 CANalyst-II 驱动配置方法:

无

2.3.3 UDS 驱动文件列表:

文件名	版本号
ControlCAN.dll	1.7.9.7
uds_driver.dll	2.1.0.0

2.4. PEAK PCAN

2.4.1 PEAK PCAN 驱动安装方法:

安装 PeakOemDrv.exe, 版本 4.2.3.353

2.4.2 PEAK PCAN 驱动配置方法:

无

2.4.3 UDS 驱动文件列表:

文件名	版本号
PCANBasic.dll	4.5.4.508
uds_driver.dll	1.0.0.0

2.5. Value CAN3

2.5.1 Value CAN3 驱动安装方法:

安装 Install_ES581_USB_Drivers.exe, 版本 1.0.0.0

2.5.2 Value CAN3 驱动配置方法:

无

2.5.3 UDS 驱动文件列表:

文件名	版本号
icsneo40.dll	3.6.0.45
uds_driver.dll	1.0.0.0

2.6. TOSUN TsMaster

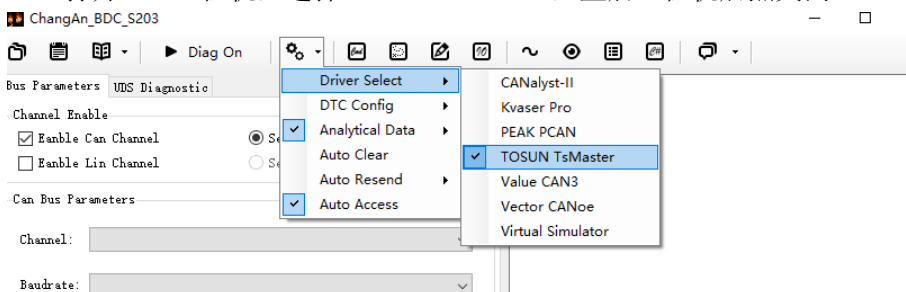
2.6.1 TOSUN TsMaster 驱动安装方法:

安装\\192.168.8.20\软件安装包\CAN\TSMaster\ TSMaster_Setup_alpha+_4_12.exe

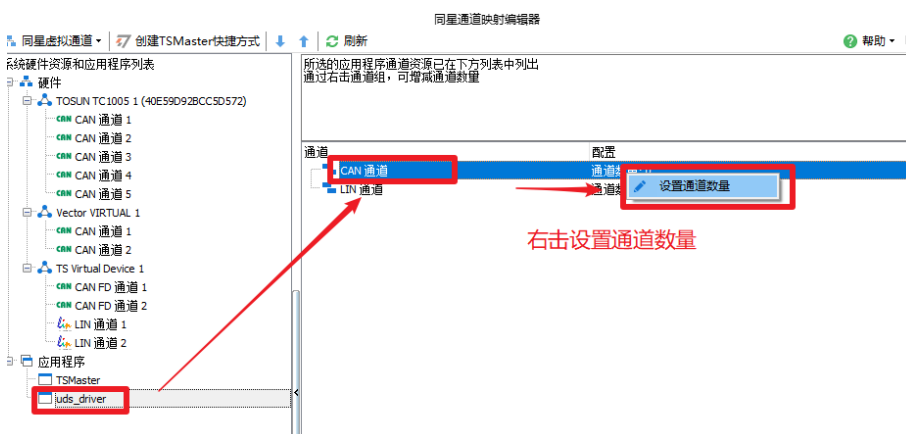
注: 安装其他版本同星软件可能会因为同星的驱动版本不一致, 导致无法正常驱动同星工具

2.6.2 TOSUN TsMaster 驱动配置方法:

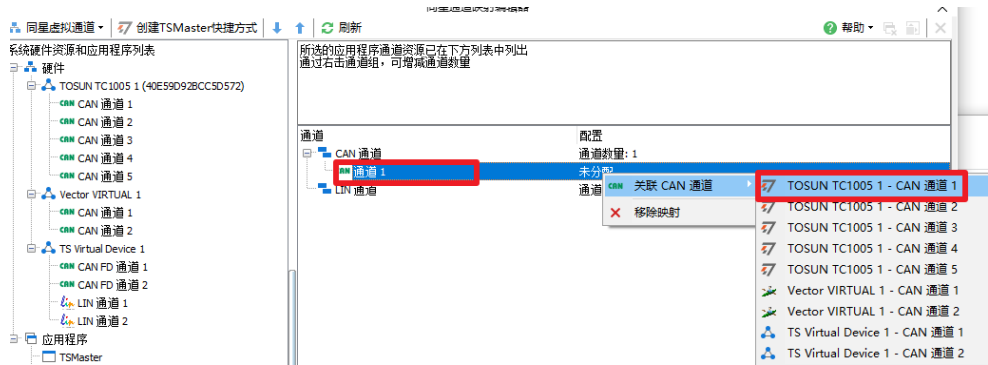
- 打开 uds 上位机, 选择 TOSUN TsMaster, 重启上位机后点关闭



- 在同星通道映射编辑器中设置 CAN 通道数量为 1 (也可以是其他数目)



- 在同星通道映射编辑器中设置通道映射 (必须全部关联!)

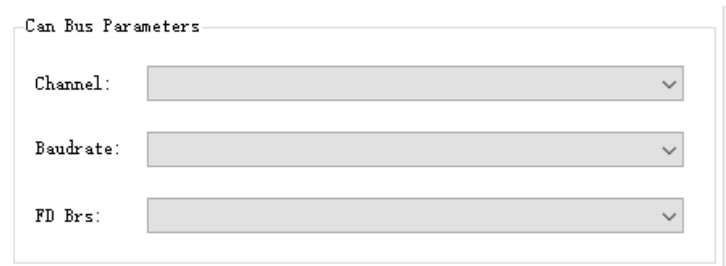


- 关闭同星通道映射编辑器窗口

注意事项 1: 所有配置的通道, 必须分配有效的硬件/虚拟通道, 否则会导致无法设备无法打开, 入下图 LIN 通道未分配, 最终将无法打开同星设备:



注意事项 2: 同星驱动由于部分版本 sdk 升级了 api, 导致不同版本的 TsMaster 和 uds 的驱动无法共用, 表现现象如下图, 因此使用 uds 需要升级到 TSMaster_Setup_alpha+_4_12.exe, 地址为\\192.168.8.20\软件安装包\CAN\TSMaster\TSMaster_Setup_alpha+_4_12.exe



注意事项 3: 同星驱动在运行时需要用管理员权限才能运行, 如果出现选择同星后未显示通道 (如上图), 可以尝试运行 uds.RunAsAdmin.exe 来提升权限 (仅公司部署域控, 无管理员权限的电脑, 有管理员权限的可以尝试右击 uds.exe 以管理员权限运行)

2.6.3 UDS 驱动文件列表:

文件名	版本号
-----	-----

Interop.TSMasterAPI.dll	2023.10.20.978
TSMaster.dll	2024.4.12.1103
uds_driver.dll	1.5.0.0

2.7. Zlg USBCAN

2.7.1 Zlg USBCAN 驱动安装方法:

安装 USBCAN_AllInOne_x86_x64_2.0.0.1.exe

安装 2013vcredist_x64.exe

安装 2013vcredist_x86.exe

注意事项: Zlg 驱动需要依赖 VC++运行库 (2013vcredist_x86.exe 和 2013vcredist_x64.exe), 地址为\\192.168.8.20\软件安装包\CAN\UDS Test\驱动程序\zlg_usbcan-II.zip

2.7.2 UDS 驱动文件列表:

文件名	版本号
zlgcan.dll	1.1.0.8
kerneldlls	
uds_driver.dll	2.6.0.0

2.8. iTEK USBCAN-II

2.8.1 iTEK USBCAN-II 驱动安装方法:

安装 iTEK USBCAN 驱动.rar

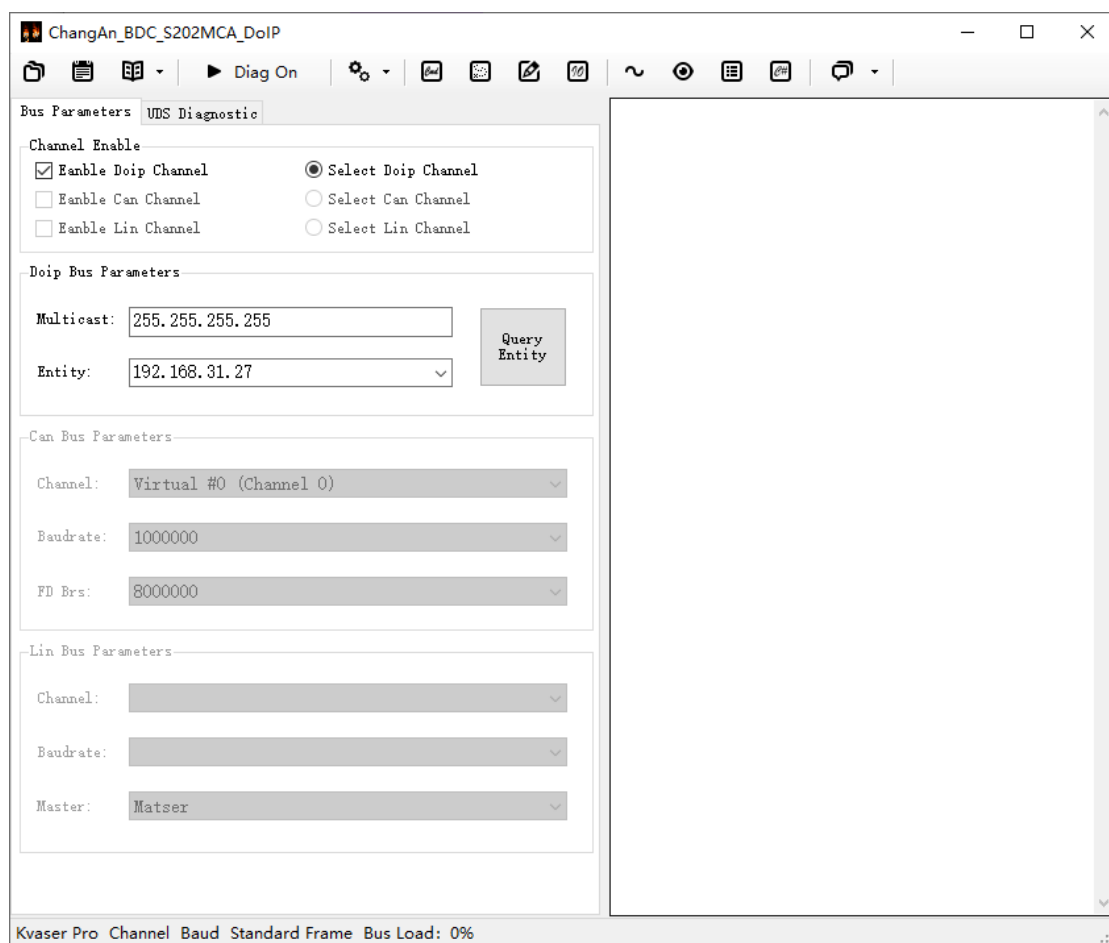
2.8.2 UDS 驱动文件列表:

文件名	版本号
ControlCAN.dll	1.0.1.1
kerneldlls	
uds_driver.dll	2.1.0.0



Chapter IV UDS 上位机

1. 主界面

1.1. 菜单界面



- 载入配置文件：配置文件中包含项目名称、CAN 或 LIN 设置和 UDS 服务配置
- 保存配置模板文件：
- 配置文件历史记录：下拉菜单中可以左击载入，右击删除
- 主窗体设置：见[主窗体设置](#)
- ECU 快捷命令工具：见[ECU 快捷命令工具](#)
- ECU 变量读写工具：见[ECU 变量读写工具](#)
- ECU 软件配置工具：见[ECU 软件配置工具](#)
- ECU 输入输出控制工具：见[ECU 输入输出控制工具](#)
- CAN 信号模拟工具：见[CAN 信号模拟工具](#)
- CAN 信号监视工具：见[CAN 信号监视工具](#)

-  CAN 数据流工具： 见 [CAN 数据流工具](#)
-  帮助： 显示软件的版本和最新支持的[安全算法](#)

1.2. 诊断设置

- Channel Enable: 设置使能 CAN 或者 LIN 通道和选择 DoIP、CAN 或者 LIN 诊断
- Multicast: 广播寻找 DoIP Entity 选择 255.255.255.255, 否则选择 DoIP Entity 的 IP
- Entity: 点击 QueryEntity 后查找到的 Entity, 或者手动输入 DoIP Entity 的 IP
- Channel: 设置 CAN 或者 LIN 通道 (自动识别)
- Baudrate: 设置 CAN 或者 LIN 波特率 (可以通过 [config.ini 配置文件](#) 设置)
- Master: 设置 LIN 通道选择 MASTER 或者 SLAVE
- Diag On/Off: 开启或关闭 CAN、LIN 通道, 本软件所有功能均需要在 Bus On 状态下执行

1.3. 诊断服务

- Session \$10: 选择诊断会话, 可以点击 Session 按键发送进入该会话命令, 如下拉选项选择\$01 Default Session, 按下 Session 按键后, 上位机将会发送 02 10 01
- Level \$27: 选择诊断安全等级, 可以点击 Access 按键进入该安全等级, 如下拉选项选择\$01 Extended, 按下 Access 按键后, 上位机将会发送 02 27 01, 如果此时 ECU 正常回复 27 02 指令后, 上位机将会自动回复 27 03 进行安全认证。
安全等级可以通过 [config.ini 配置文件](#) 中\$27 Level:配置。
上位机支持自适应 2Byte、4Byte 或其他字节的安全算法, 由于不同项目的安全算法可能有所不同, 所以需要通过配置文件 config.ini 选择安全算法, 详细见 [config.ini 配置文件](#)\$Security Access:。
- PhysicalAddr: 选择使用物理寻址还是功能寻址, 选中后为物理寻址, 软件默认为选中。
- KeepOnLine: 选择是否保持诊断模式, 选中后上位机将会每 3s 发送一次功能寻址且不需要回复的 Test Peasant (3E 80), 通知 ECU 保持诊断在线, 软件默认选中。
- ShowOnLine: 选择是否显示诊断保持指令, 软件默认未选中。
- Cmd: 快捷指令: 该快捷指令可以通过 [config.ini 配置文件](#) 中\$Cmd:配置, 选择后点击右侧 Start 按键发送。
- SID: 诊断服务的服务 ID。部分可以通过 [config.ini 配置文件](#) 配置
- FUN: 诊断服务的子功能。部分可以通过 [config.ini 配置文件](#) 配置
- PID: 诊断服务的参数 ID。部分可以通过 [config.ini 配置文件](#) 配置
- TextBox: 更改 SID、FUN、PID 后自动更新, 也可以手动输入, 按回车键或右侧 Start 按键发送
- PARM: 例程控制里的参数

1.4. 诊断数据流

用于显示诊断的数据流, 可以通过右键菜单保存、复制、清除。

1.5. 主窗体设置

- **Driver Select:** 选择 CAN/LIN 设备的驱动
- **DTC Code Convert:** DTC 转义设置，选中后将自动将进行转义，如 910087 将会显示为 U110087，默认关闭。
- **DTC Code Analyze:** DTC 解析设置，选中时自动解析 19 服务的 DTC 相应 Bit，默认开启。
- **Analytical Data:** 选择是否解析所有服务的数据流，选中时自动解析，默认开启。
- **Show ASCII:** 选择是否显示读取数据的 ASCII，选中时自动解析，默认开启。
- **Each Row x Byte:** 选择解析数据中每行显示多少字节，默认 8 Byte。
- **Auto Clear:** 选择是否开启自动清除数据流，开启时每次发送一帧诊断消息之前先清除所有数据流，默认关闭。
- **Auto Resend:** 选择是否开启自动重发功能，开启后上位机会周期发送上次发送的诊断消息，默认关闭。
- **Int Time x.x s:** 选择自动重发间隔时间，默认 2.0s

2. 工具箱

2.1. ECU 快捷命令工具

2.1.1 工具简介

ECU 快捷命令工具：可以点击相应按钮发送一条或顺序发送一条以上诊断消息

2.1.2 工具界面



2.1.3 工具配置方式

配置方法见 [Config.ini](#) 中的\$Cmd

2.2. ECU 变量读写工具

2.2.1 工具简介

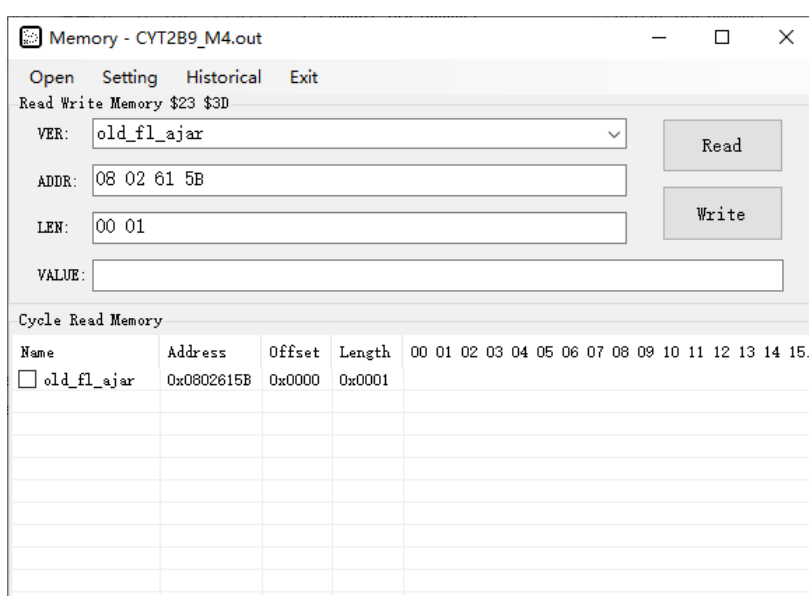
ECU 变量读写工具：采用 UDS 的\$23 服务完成对 ECU 的 RAM 读写或者 FLASH 的读取，工具支持解析编译生成的 MAP 文件，自动解析变量的地址和长度。且支持周期读

取数个变量。

目前该工具支持解析如下 MAP 文件：

- 1、 Freescale G128（CodeWarrior 5.1）系列
- 2、 ST PowerPC（SPCStudio）系列
- 3、 ST STM8（STVD）系列（仅支持解析地址）
- 4、 Renessa RH850（CS+）系列
- 5、 MPC5744（GHS）系列
- 6、 Elf 格式文件（*.elf*.out）

2.2.2 工具界面



- **Open:** 打开 MAP 文件，打开后上位机将会从 MAP 文件中解析函数、全局变量的名称、地址和长度，并将所有函数和变量名称存放到 VAR 下拉列表中。
- **Show Function:** 显示函数，默认关闭。
- **Show Read Only:** 显示只读变量，默认关闭。
- **Encrypt:** 读取自动解密，目前最新 UDS 库中通过 23 服务读取变量会加密后再传输，故需要上位机解密还原，默认开启。
- **Address Length 4Byte:** 设置 23 和 3D 服务发送的地址和长度固定为 4 字节，否则默认按照自动计算。
- **VER:** 存放所有解析出的函数名和变量名。
- **TYPE:** 用于区分变量的类型。
- **ADDR:** 用于设置变量的地址，当 VER 选择改变时，会自动更新地址，也可手动修改，用于读取未支持的芯片。
- **LEN:** 用于设置变量的长度，当 VER 选择改变时，会自动更新长度，也可手动修改，用于读取未支持的芯片。
- **Read:** 按下 Read 按键时，会自动向 ECU 发送读取变量命令，命令中包含读取的地址和长度，ECU 收到指令后，会回复读取的内容。
- **VALUE:** 待写入的值。必须也只允许为十六进制数字。
- **Write:** 按下写入按键时，会自动向 ECU 发送写入变量命令，命令中包含写入的地

址、长度和数据，写入时必须保证 VALUE 的长度与 LEN 相等。

2.2.3 周期读取变量

- Step1: 载入软件对应版本的 MAP 文件或 elf 文件。
- Step2: 在 VER 中选择需要读取的变量。
- Step3: 右击菜单中选择 Add 添加变量。
- Step4: 鼠标点击最左侧选择框。
- Step5: 上位机将自动读取变量，并将变量值更新到名称右侧。
- Step6: 右击菜单中 Delete 删除变量

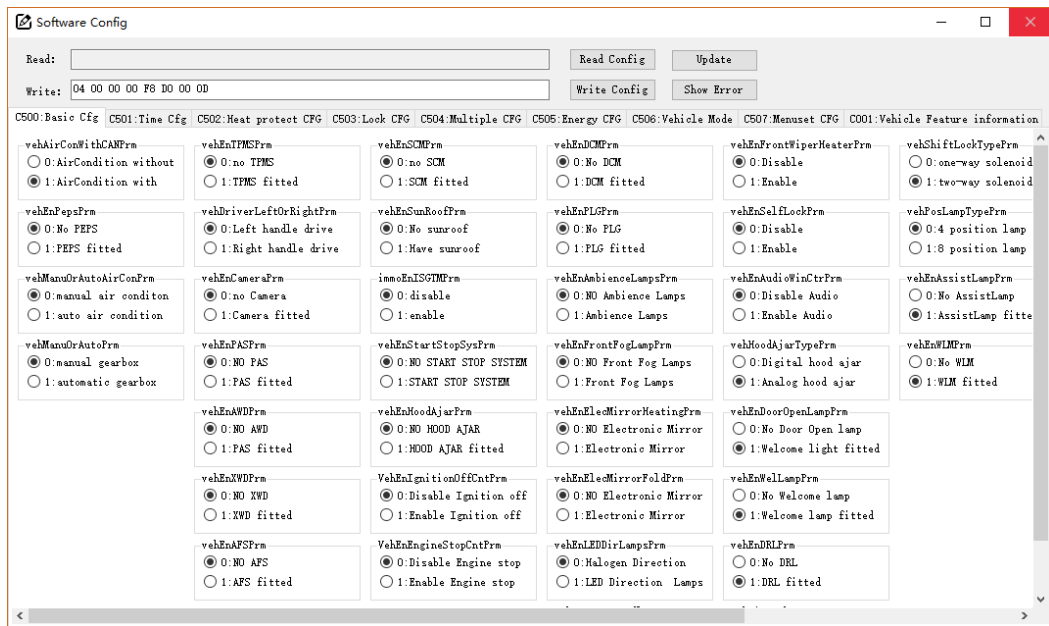
如果没有 MAP 文件可以直接在对应文本框中输入相应地址和长度进行读取

2.3. ECU 软件配置工具

2.3.1 工具简介

ECU 软件配置工具：采用 UDS 的\$22、\$2E 服务读写软件 E2 配置字节。该工具可以解析读取并解析配置字节的每个 Bit 功能定义，需要从主界面载入 [config.ini 文件](#)。

2.3.2 工具界面



- **Read Config:** 通过\$22 服务读取 ECU 软件配置字节，读取到的内容将显示在 Read: 右侧的文本框中，该文本框只读，不可编辑。
- **Update:** 将 Read Config 读到的配置字节，解析并更新到下方显示界面中。
- **Write Config:** 通过\$2E 服务写 ECU 软件配置字节，将 Write: 右侧的文本框中的数字写入 ECU 的相应配置中，文本框仅支持也只允许填写十六进制数字。当下方解析界面选项被改变时，会将更改位自动更新到文本框对应位中。注意：上位机不判断填写的文本的长度，需要用户自己注意长度是否正确，因此不建议用户自行修改文本框中数字。
- **Show Error:** 显示配置文件中的错误信息。

2.3.3 配置方法

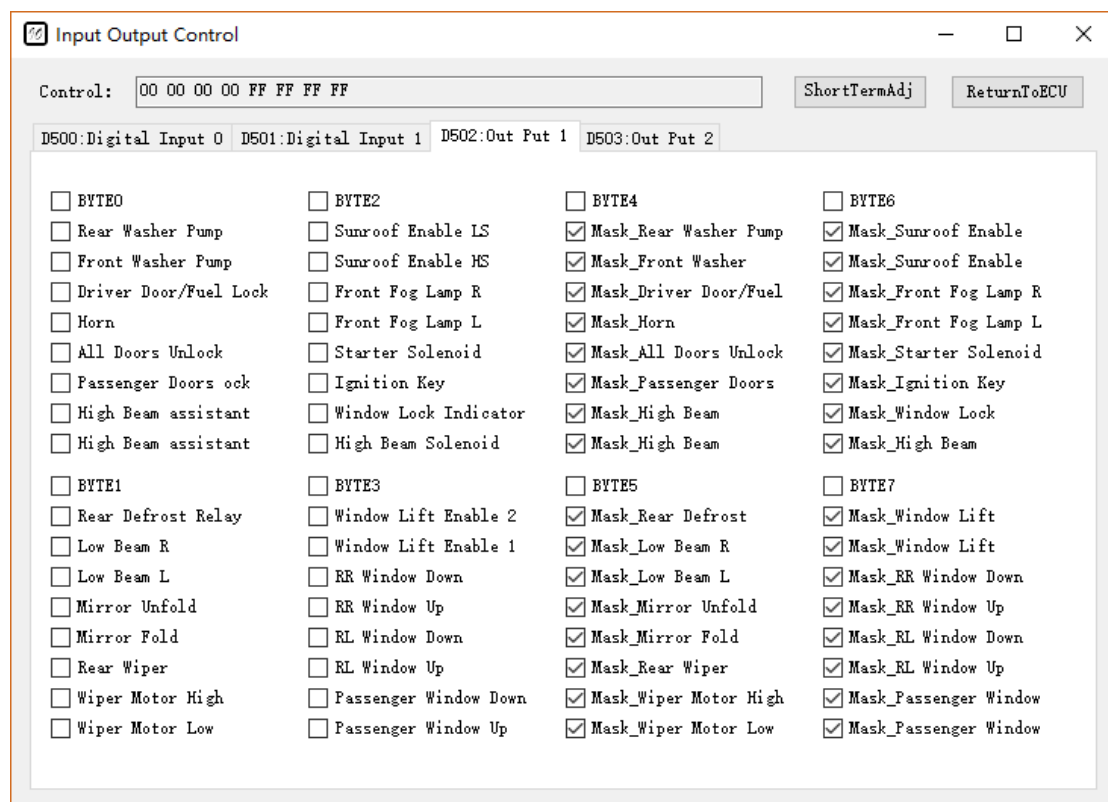
配置方法见 [config.ini 文件](#)配置中\$2E Config:

2.4. ECU 输入输出控制工具

2.4.1 工具简介

ECU 输入输出控制工具：使用 UDS 的 2F 服务，控制输入和出口的状态

2.4.2 工具界面



- Control 文本框显示当前的输入输出控制设置
- ShortTermAdj 发送命令进行输入输出的短时控制
- ReturnToECU 退出控制状态，将控制权交还给 ECU

2.4.3 工具配置

工具配置方法见 [Config.ini](#) 文件的\$2F Control

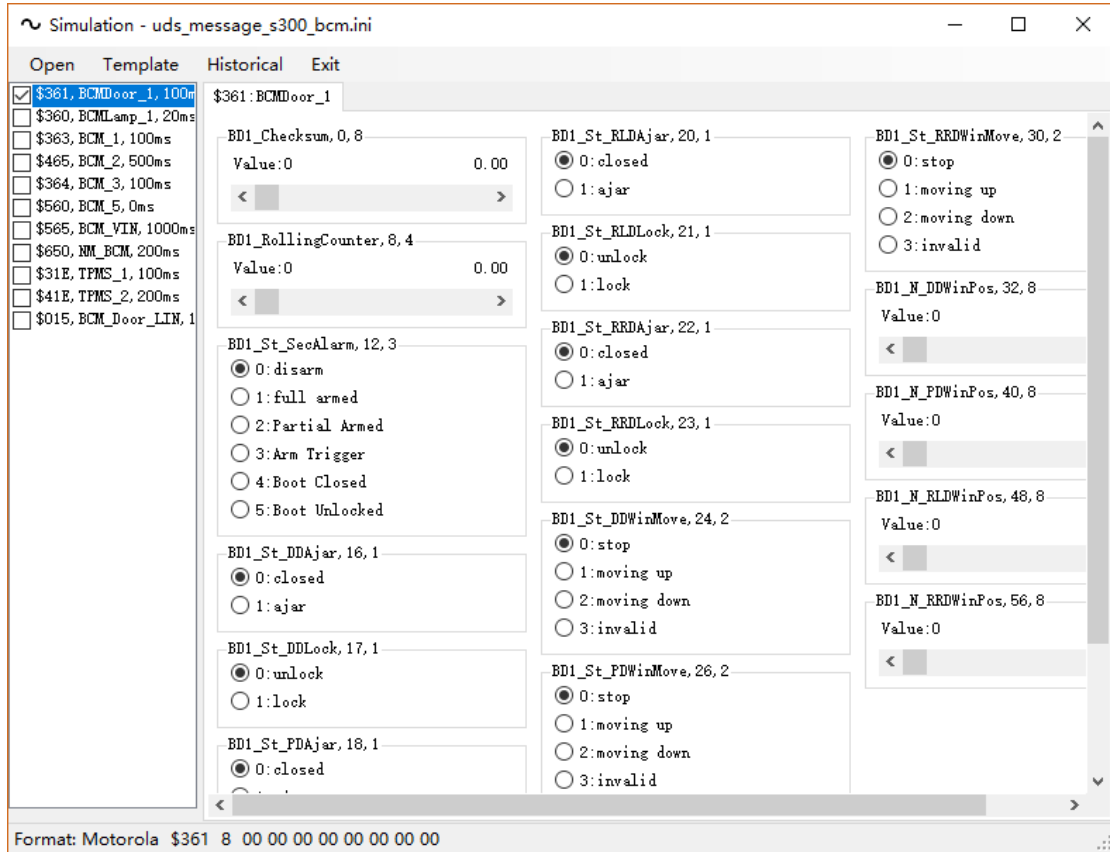
2.5. CAN 信号模拟工具

2.5.1 工具简介

CAN 信号模拟工具：用于台架模拟其他节点周期 CAN 消息，方便软件测试。可以通过载入 message.ini 配置文件适应不同的项目，该工具支持设置每个 CAN 信号帧的格式（Intel 或 Motorola）、长度、周期等。

目前该工具支持 Value 和 PhyValue 两种类型。

2.5.2 工具界面



- 左侧使能栏用于使能 CAN 信号的发送，仅当选项框被选中时，该帧开始按设定周期发送，发送周期最小支持 10ms。
- 右侧选项页中可以单独选择每个信号的当前值，对于 Value 值类型，可以通过鼠标点选其中某个 Value 进行设置；对于 PhyValue 值类型，可以通过鼠标拖动选择，或者通过右击填入数值（只支持十进制）之后按回车设置。

2.5.3 配置方法

配置方法见 [message.ini 文件配置](#)

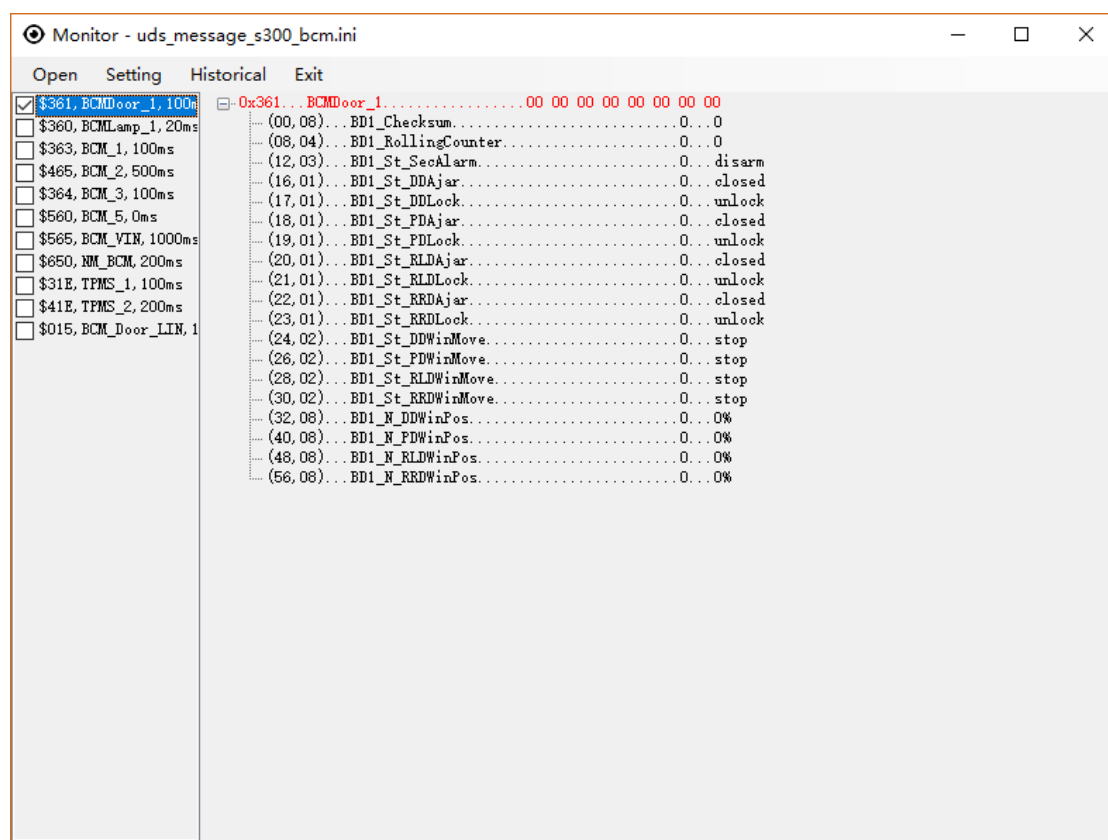
2.6. CAN 信号监视工具

2.6.1 工具简介

CAN 信号监视工具：用于台架监视其他节点周期 CAN 消息，并解析 CAN 消息中每个信号的值，方便软件测试。可以通过载入 message.ini 配置文件适应不同的项目，该工具支持设置每个 CAN 信号帧的格式（Intel 或 Motorola）、长度、周期等。

目前该工具支持 Value 和 PhyValue 两种类型。

2.6.2 工具界面



- 左侧使能栏用于使能 CAN 信号的监视。
- 右侧窗口用于显示 CAN 信号的解析结果。

2.6.3 配置方法

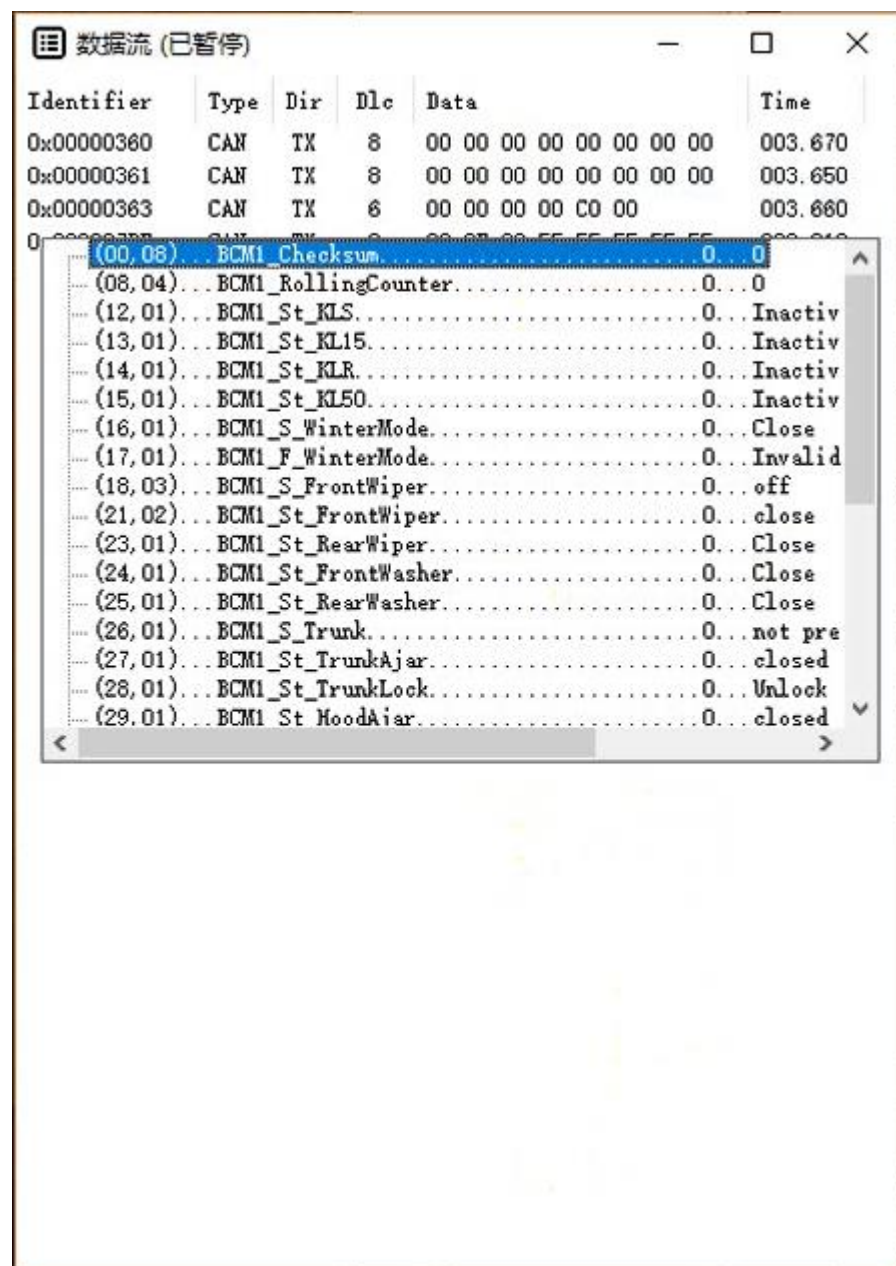
配置方法见 [message.ini 文件配置](#)

2.7. CAN 数据流工具

2.7.1 工具简介

CAN 数据流工具：用于显示总线上所有节点消息，可以点击表头进行节点消息筛选，未被选中的消息将不会更新显示到该工具上。

2.7.2 工具界面



2.7.3 工具使用

- 消息过滤：可以左击表头的 Identifier、Type、Dir 进行对应的过滤，未被选中的将全部被过滤，不显示在列表中
- 消息解析：可以在暂停时，双击消息行显示该帧消息的解析数据（必须先打开 Monitor 窗口载入 message.ini 文件后才能使用）

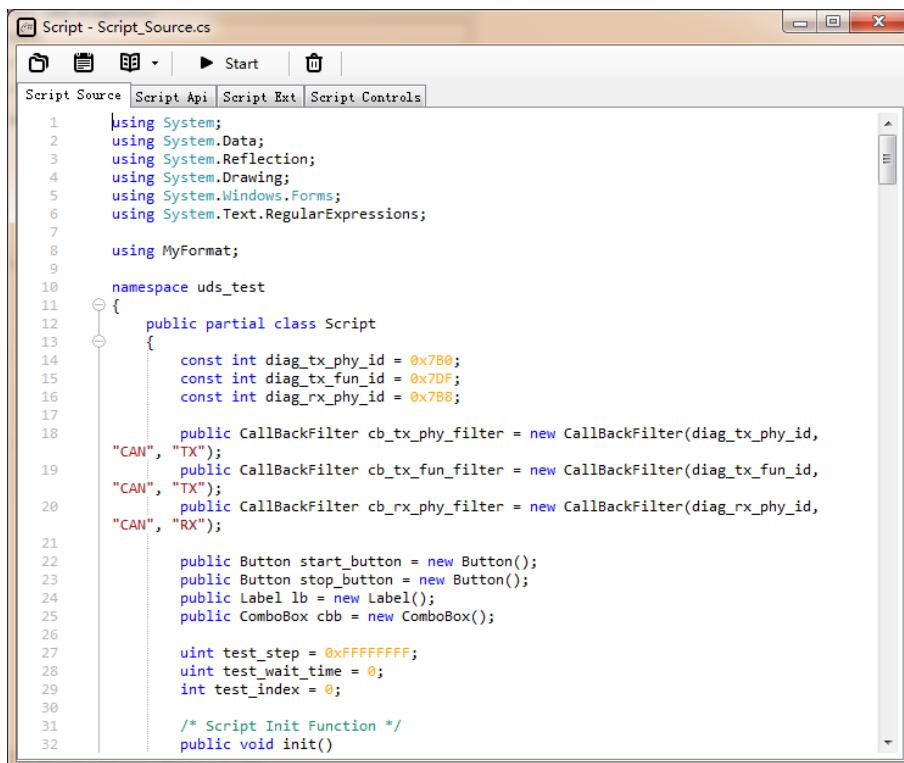
2.8. C# 脚本工具

2.8.1 工具简介

C#脚本工具：使用 C#语言编写简单脚本，载入后直接运行，无需重新编译上位机程序。目前该脚本支持以下功能：

- 常用窗体控件（Button、Label、TextBox、ComboBox 等）
- Task 10ms 任务
- 收到 CAN 或 LIN 信号执行 callback
- Callback 过滤
- 发送 CAN 或 LIN 信号
- 发送 CAN 或 LIN 诊断消息
- 发送 CAN 或 LIN 信号执行 tx_append
- 打印文本到脚本工具下方文本框或者主窗体数据流窗体

2.8.2 工具界面



2.8.3 工具使用

- 常用控件

接口: `public Button start_button = new Button();`
`public Button stop_button = new Button();`
`public Label lb = new Label();`
`public ComboBox cbb = new ComboBox();`

例:

```
public Label lb = new Label();
/* Script Init Function */
public void init()
{
    lb.Text = "BORGWARD S300 BCAN TP Test";
    lb.AutoSize = true;
    lb.Location = new Point(6, 6);
}
```

- init 初始化函数

启动脚本时会首先调用此函数

接口: `public void init()`

- Task 10ms 任务

载入脚本文件后, 点击 Start 按钮, 脚本将会以 10ms 周期执行 tick_10ms()函数

接口: `public void tick_10ms()`

- Callback 过滤

Callback 可以根据 Id、Type 和 Dir 进行过滤 (可以添加多个过滤规则)

接口: `public CallbackFilter(int id, string type, string dir);`
`public CallbackFilter(int start, int end, string type,`
`string dir);`

过滤方法如下:

例 1: 仅接收 0x7DF 的 CAN 信号, 类型为 RX 可以定义

```
public CallbackFilter call_back_7df = new CallbackFilter(0x7DF,
"CAN", "RX");
```

例 2: 接收 0x700~0x7DF 的 CAN 信号, 类型为 RX 可以定义

```
public CallbackFilter call_back_700_7DF = new
CallbackFilter(0x700, 0x7DF, "CAN", "TX");
```

- 收到 CAN 或 LIN 信号后执行 callback

callback 函数会传入收到的 Id、Type ("CAN" or "LIN")、Dir ("TX" or "RX")、Dlc、Dat 和 Time

接口: `public void call_back(int id, string type, string dir, int`
`dlc, byte[] dat, long time)`

- 收到一包诊断消息后执行 diag_callback

收到的诊断消息会以 byte 数组的形式传入函数

接口: `public void diag_call_back(byte[] msg)`

- 发送 CAN 或 LIN 信号
接口: `void TxFrame(int id, string type, int dlc, byte[] dat);`
例: `TxFrame(0x7df, "CAN", 0, new byte[0]);`
- 发送 CAN 或 LIN 诊断消息
接口: `private void TxDiagMsg(string strings);`
`private void TxDiagMsg(string[] strings);`
`private void TxDiagMsg(byte[] data);`
 例 1: `TxDiagMsg("1003");`
 例 2: `TxDiagMsg(new string[]{"1003", "2703"});`
 例 3: `TxDiagMsg(new byte[2] { 0x27, 0x01 });`
- 发送 CAN 或 LIN 信号回调函数
接口: `public void tx_append(int id, string type, int dlc, ref byte[] dat)`
 例:

```
/* Script Can Or Lin Msg Transmit Append Function */
public void tx_append(int id, string type, int dlc, ref byte[] dat)
{
    if(id == 0x256 && dlc == 8)
    {
        int checksum = dat[0]^dat[1]^dat[2]^dat[3]^
            dat[4]^dat[5]^dat[6]^dat[7];
        dat = dat.SetBitMotorola(56, 8, (uint)checksum);
    }
}
```
- 打印文本到脚本工具或主窗体数据流窗体
接口: `void ShowString(string strings);`
`void ShowInfo(string strings);`
 例: `ShowInfo("收到功能寻址诊断帧");`

Chapter V 安全算法

1. 算法说明

由于每个项目的安全算法都有所不同，上位机为了适应所有项目，将安全算法打包成 SecurityAccess.dll 文件，以方便修改。为与 ECU 端安全算法最小修改，dll 采用 C 编写，由 cFree5 编译。

1.1. 安全算法不大于 4 字节

工程 SVN 地址: <https://xxx.xxx.xxx.xxx:8443/svn/SecurityAccess>

目前支持的项目可以通过上位机主界面→更新记录进行查看

该工程仅支持 2Byte 和 4Byte 安全算法，大于 4Byte 请使用 C#脚本工具实现。

1.2. 安全算法任意字节

工程 SVN 地址: https://xxx.xxx.xxx.xxx:8443/svn/branches/uds_seed2key

目前支持的项目可以通过上位机主界面→更新记录进行查看

该工程支持 256 字节以内 Seed 和 Key 的安全算法。

2. 添加方法:

2.1. cFree5 编译

- 在添加安全算法函数,接口为 UINT32 SecurityAccess_XX_YY_ZZ(UINT32 seed, UINT32 level), 添加安全算法 (XX 为主机厂名称, YY 为项目类型, ZZ 为项目代号)
- 将安全算法添加到 SecurityAccessTable 表中, 如 { &SecurityAccess_ZOTYE_BDM_B21, "众泰 BDM_B21" }, (格式为安全算法的函数地址和安全算法的描述)
- 在 char* __cdecl About(void)函数中将 build version 加一, 用于区分不同版本的 dll
- 打开 Cfree5 工程 SecurityAccess.cfp
- 菜单->Build->Build SecurityAccess(project)
- 生成的 dll 在 MinGW\SecurityAccess.dll
- 将修改上传至 SVN

2.2. VC++编译

用 VS2017 可以采用 C++编写安全算法, 实现以下接口编译生成 dll 文件替换即可使用

```
SECURITYACCESS_API
```

```
UINT32 __cdecl SecurityAccess(UINT32 project, UINT32 seed, UINT32 level)
```

```
{  
  
    return Result;  
}
```

```
char* __cdecl About(void)
{
    return Result;
}
```

Chapter VI config.ini 文件配置

1. 文件说明:

config.ini 文件是 uds 上位机主界面和 ECU 软件配置工具所使用的配置文件，文件以固定格式的文本进行存储，以方便手工修改。

2. 格式说明:

///配置说明:

///每行以\$xx yyy:起始的为有效行，\$xx yyy:请勿修改

///每行\$xx yyy:之后的为该服务的数据，数据中以','分开

///每行以///起始的为注释行，说明之后的数据存放格式，请勿删除

///注意：每行除空白行、注释行和有效行之外任何数据均会报错

///配置文件中所有标点符号必须使用英文符号

- **\$Project**:配置文件项目名称，载入配置文件后将会在主窗体的标题栏显示该名称。

格式 ///\$Project: + Name (string)

例：宝沃 S300 项目

\$Project:UDS_DIAG_S300

- **\$Tx Id**:物理寻址地址，在 CAN 诊断中用于填写物理地址，在 LIN 诊断中用于填写待诊断的从节点 NAD。

格式 ///\$Tx Id: + Value (hex)

例：物理寻址地址 0x7B0

\$Tx Id:0x7B0

- **\$Tx Fun Id**:功能寻址地址，在 CAN 诊断中一般使用 0x7DF，在 LIN 诊断中一般填写 0x7E。

格式 ///\$Tx Fun Id: + Value (hex)

例：功能寻址地址 0x7DF

\$Tx Fun Id:0x7DF0

- **\$Rx Id**:诊断接收地址，在 LIN 诊断中用于填写待诊断的从节点 NAD。

格式 ///\$Rx Id: + Value (hex)

例：接收地址 0x7B8

\$Rx Id:0x7B8

- **\$Baud**:诊断使用的波特率，在 CAN 诊断中可以填写 1000000、500000、250000、125000 等，在 LIN 诊断中可以填写 19200、9600。

格式 ///\$ Baud: + Value (dec)

例：波特率 500K

\$Baud:500000

- **\$Fd Baud**:诊断使用的波特率 (CAN_FD)，可以填写 8000000、4000000、2000000、1000000 等。

格式 ///\$ Fd Baud: + Value (dec)

例：波特率 1000K

\$Fd Baud:1000000

- **\$Fd Frame**:是否使用 CAN_FD, false 为标准 CAN 帧, true 为 CAN FD 帧。

格式 /// \$Fd Frame: + Value(bool)

例：标准帧

\$Fd Frame:false

- **\$Fd Iso**:是否使用 ISO 模式 (CAN_FD 下有效), false 为非 ISO CANFD, true 为 ISO CANFD。

格式 /// \$Fd Iso: + Value (bool)

例：ISO 模式

\$Fd Iso:true

- **\$Fd Brs**:数据场是否改变比特率 (CAN_FD 下有效, 比特率为 Fd Baud), false 为使用 Baud, true 为 Fd Baud。

格式 /// \$Fd Brs: + Value (bool)

例：变比特率模式

\$Fd Brs:true

- **\$Extended Frame**:是否使用扩展帧, 0 为标准帧, 1 扩展帧。

格式 /// \$Extended Frame: + Value (bool)

例：标准帧

\$Extended Frame:0

- **\$N_AS**:传输层参数 N_AS。

格式 ///\$N_AS: + Value (dec)

例：N_AS 设置为 1000ms

\$N_AS:1000

- **\$N_BS**:传输层参数 N_BS。

格式 ///\$N_AS: + Value (dec)

例：N_BS 设置为 1000ms

\$N_BS:1000

- **\$BlockSize**:传输层参数 BlockSize。

格式 ///\$BlockSize: + Value (dec)

例：BlockSize 设置为 8

\$BlockSize:8

- **\$Stmin**:传输层参数 stmin。

格式 ///\$Stmin: + Value (dec)

例：Stmin 设置为 10ms

\$Stmin:10

- **\$FillByte**:传输层参数 FillByte。

格式 ///\$FillByte: + Value (hex)

例：FillByte 设置为 55

\$FillByte:0x55

- **\$Fd 64 Byte**:CANFD 传输层按照 64 字节发送。

格式 ///\$Fd 64 Byte: + Value (bool)

例：Fd 64 Byte 设置为 false (按照 8 字节发送)

\$Fd 64 Byte:false

- **\$DoIP MulticastAddress**: DoIP 查找 Entity 的广播地址，广播时填 255.255.255.255，否则填写 Entity 的 IP 地址。
格式 `///$DoIP MulticastAddress: + Value (IP 地址)`
例：DoIP MulticastAddress 设置为 192.168.31.27
`$DoIP MulticastAddress: 192.168.31.27`
- **\$DoIP EntityAddress**: DoIP Entity 的默认地址，可以通过广播查找，也可以手动填写。
格式 `///$DoIP EntityAddress: + Value (IP 地址)`
例：DoIP EntityAddress 设置为 192.168.31.27
`$DoIP EntityAddress: 192.168.31.27`
- **\$DoIP LogicalAddress**: DoIP Tester 的逻辑地址。
格式 `///$DoIP LogicalAddress: + Value (hex)`
例：DoIP LogicalAddress 设置为 0x0E01
`$DoIP LogicalAddress: 0x0E01`
- **\$DoIP ActivationType**: DoIP 路由激活的类型，可以选择 0x00(Default)、0x01(WWH-OBD)、0xE0(Central Security)。
格式 `///$DoIP ActivationType: + Value (hex)`
例：DoIP ActivationType 设置为 0x00 (Default)
`$DoIP ActivationType: 0x00`
- **\$DoIP SourceAddress**: DoIP 诊断消息的源地址。
格式 `///$DoIP SourceAddress: + Value (hex)`
例：DoIP SourceAddress 设置为 0x0E01
`$DoIP SourceAddress: 0x0E01`
- **\$DoIP TargetAddress**: DoIP 诊断消息的目的地址。
格式 `///$DoIP TargetAddress: + Value (hex)`
例：DoIP TargetAddress 设置为 0xE000
`$DoIP TargetAddress: 0xE000`
- **\$DoIP ProtocolVersion**: DoIP 协议版本，仅支持 0x00-0x03。
格式 `///$DoIP ProtocolVersion: + Value (hex)`
例：DoIP ProtocolVersion 设置为 0x02 (iso13400-2012)
`$DoIP TargetAddress: 0x02`
- **\$DoIP OemSpecific**: DoIP OEM 字段设置。删除此字段时，发送的路由激活请求中不会包含 OEM 字段。
格式 `///$DoIP OemSpecific: + Value (hex)`
例：DoIP OemSpecific 设置为 FFFFFFFF
`$DoIP OemSpecific: 0xFFFFFFFF`
- **\$Security Access**: 使用的安全算法，可以根据不同项目更改，目前支持的安全算法见安全算法章节
格式 `/// $Security Access: + Value (hex)`
例：宝沃 S300
`$Security Access: 0x00`
注：Security Access 小于 0x10000 使用 SecurityAccess.dll
Security Access 大于等于 0x10000 使用 uds_seed2key.dll
- **\$27 Level**: 使用的安全等级，

格式 ///\$27 Level: + Value (hex)+ ',' + Name(string)

例：需要添加安全等级 0x11，进行 IMMO 认证可以添加

\$27 Level:11,IMMO

- **\$27 Constant:**使用安全算法的常量，用于支持个别项目安全算法中需要 2E 写入常量的特殊需求（如吉利）。

格式 ///\$27 Constant: + Value (hex)

例：\$27 Constant:55555555

- **\$Cmd:**快捷指令，可以通过选择主界面快捷指令下拉选择后发送，该命令允许发送多条诊断指令。

格式 ///\$Cmd: + cmd + ',' + ... + Name(string)

例：快速配置 E2，先进入安全等级 1，之后写入 E2

\$Cmd:1003,2701,2EC50100000000, 快速配置 E2

- **\$01 ObdPid:**OBID Pid 解析(string)，存放每条 Pid 对应的故障名称，用于 01 服务解析数据使用。

格式 ///\$01 ObdPid: + Ident + ',' + Name

例：增加 Pid 01，自诊断故障码清除后的监视状态

\$01 ObdPid:01, 自诊断故障码清除后的监视状态

- **\$03 ObdDtc:**OBID DTC 解析(string)，存放每条 DTC 对应的故障名称，用于 03 服务解析数据使用。

格式 ///\$03 ObdDtc: + Ident + ',' + Name

例：增加 DTC D563，Voltage_High_AB8017_Circuit_Voltage_Above_Threshold

\$03 ObdDtc:D563, Voltage_High_AB8017_Circuit_Voltage_Above_Threshold

- **\$10 Session:**使用的安全会话。

格式 ///\$10 Session: + Value (hex)+ ',' + Name(string)

例：\$10 Session:01,默认会话

\$10 Session:02,编程会话

\$10 Session:03,扩展会话

\$10 Session:04,测试会话

- **\$19 Dtc:**DTC 解析(string)，存放每条 DTC 对应的故障名称，用于 14 和 19 服务解析数据使用。

格式 ///\$19 Dtc: + Dtc(string) + ',' + Name(string)

例：增加 DTC U110087, Lost_BCAN_Communication

\$19 Dtc:C10087,Lost_BCAN_Communication

或者\$19 Dtc:U110087,Lost_BCAN_Communication

- **\$22 Read:**读服务，可以读取 IO 输入输出状态、E2 配置、零部件信息等

格式 ///\$22 Read: + Ident(hex) + ',' + Name(string)

例：ID F183，读取 BootLoad 软件版本号

\$22 Read:F183, BootLoad 软件版本号

- **\$2E Write:**写服务，一般用于写 E2（写软件配置建议使用\$2E Config:）

格式 ///\$2E Write: + Ident(hex) + ',' + Data(hex) + ',' + Name(string)

例：ID F199，写编程日期

\$2E Write:F199,201705270951,Programming Date

- **\$2F IoCtrl:**输入输出控制，一般用于终检测试。

格式 ///\$2F IoCtrl: + Ident(hex) + ',' + Data(hex) + ',' + Name(string)

例：ID D502，参数长度 8 字节，控制输出全部开启

\$2F IoCtrl:D502,FFFFFFFFFFFFFFFF,Out Put All On

- **\$31 Routine:**例程控制

格式 ///\$31 Routine: + Ident(hex) + ',' + StartParameter(hex) + ',' + StopParameter(hex) + ',' + ResultParameter(hex) + ',' + Name(string)

例：ID FDF1 快速睡眠（只需要启动，且无参数）

\$31 Routine:FDF1,,,快速睡眠

例：ID 7501 创建随机 SK（只需要启动，参数 4Byte 的 PIN）

\$31 Routine:7501,AABBCCDD,,,Generate Random Secret Key

- **\$2E Config:**软件 E2 配置，配置中可以包含多个配置\$2E Config:，每个配置可以包含多个信号@Signal:，每个信号可以包含多个#Value:。

格式 ///\$2E Config: + Id(hex) + ',' + Length(dec) + ',' + Name(string)

///@Signal: + Start(dec) + ',' + Length(dec) + ',' + Name(string) + ',' + DefaultValue(dec)

///#Value: + Value(dec) + ',' + Name(string)

例：配置 ID C500，长度 8Byte，配置名称 Basic Cfg

信号起始 bit2，长度 1bit，名称 vehAirConWithCANPrm，默认配置 1

值 0 为 AirCondition without CAN

值 1 为 AirCondition with CAN

信号起始 bit3 长度 2bit，名称 RfLongPressTimPrm，默认配置 0

值 0 为 1s

值 1 为 1.5s

值 2 为 2s

值 3 为预留

\$2E Config:C500,8,Basic Cfg

@Signal:2,1,vehAirConWithCANPrm,1

#Value:0,AirCondition without CAN

#Value:1,AirCondition with CAN

@Signal:9,2,RfLongPressTimPrm,0

#Value:0,1s

#Value:1,1.5s

#Value:2,2s

#Value:3,预留

- **\$2F Control:**软件输入输出控制配置，配置中可以包含多个配置\$2F Control:，每个配置可以包含多个信号@Value:。

格式 ///\$2F Control: + Id(hex) + ',' + Length(dec) + ',' + Name(string)

///@Value: + Start(dec) + ',' + Name(string) + ',' + DefaultValue(dec)

例：配置 ID D502，长度 8Byte，配置名称 Out Put 1

信号起始位 bit7，名称 High Beam assistant L，默认配置 0（0->关）

信号起始位 bit6，名称 High Beam assistant R，默认配置 0（0->关）

信号起始位 bit5，名称 Passenger Doors Lock，默认配置 0（0->关）

\$2F Control:D502,8,Out Put 1

@Value:7,High Beam assistant L(Left Assist Light),0

@Value:6,High Beam assistant R(Right Assist Light),0

@Value:5,Passenger Doors ock,0

Chapter VII message.ini 文件配置

1. 文件说明:

message.ini 文件是 CAN 信号模拟工具和 CAN 信号仿真工具所使用的配置文件，文件以固定格式的文本进行存储，以方便手工修改。

配置软件中支持对帧 CAN 消息设置单独的帧格式（Intel or Motorola）

message.ini 文件可以通过记事本等文本编辑器进行编辑，也可以通过通过 dbc 文件直接生成，方法见 [dbc 生成 message.ini](#) 章节

2. 格式说明:

///配置说明:

///每行以\$xx yyy:起始的为有效行，\$xx yyy:请勿修改

///每行\$xx yyy:之后的为该服务的数据，数据中以','分开

///每行以///起始的为注释行，说明之后的数据存放格式，请勿删除

///注意：每行除空白行、注释行和有效行之外任何数据均会报错

///配置文件中所有标点符号必须使用英文符号

///\$Message: + Id + ',' + Length + ',' + Name + ',' + Cycle

///&Format: + Intel or Motorola

///@Signal: + Start + ',' + Length + ',' + Name + ',' + DefaultValue

///#Value: + Value + ',' + Name

///#PhyValue: + MaxValue + ',' + Factor + ',' + Offset + ',' + Unit

配置文件中包含以下几中类型:

- 1) **\$Message**:定义一帧 CAN 消息，包含消息的 ID(hex)、字节长度(dec)、名称(string)、发送周期(dec)等信息，配置文件中可以有多个消息。
- 2) **&Format**:定义消息的格式(string)，可选 Intel 或 Motorola，每个消息都应包含一个格式
- 3) **@Signal**:定义一个信号，包含信号的起始位(dec)、长度位(dec)、名称(string)、默认值(dec)，每个消息中可以包含多个信号，但信号的相关信息不能有重复或无效值。
- 4) **#Value**:定义一个值，包含值(dec)和名称(string)，每个信号中可以包含多个值，但值的相关信息不能有重复。
- 5) **#PhyValue**:定义一个值，包含值的最大值(float)、精度(float)、偏移(float)和单位(string)，该值的解析结果为 $\text{Factor} * \text{value} + \text{Offset} + \text{Unit}$ 。每个信号中只能包含一个 PhyValue 或多个 Value。

例：CAN 消息 0x535，长度 8Byte，名称 AUDIO1，发送周期 500ms

格式 Motorola

信号起始位 28，长度 1 位，名称 Park Light，默认值 0

值 0 为 Inactive

值 1 为 Active

信号起始位 40，长度 15 位，名称 BR1_N_VehicleSpeed，默认值 0

最大值 10000，精度 0.01，偏移 0，单位 km/h

\$Message:535,8,AUDIO1,500

&Format:Motorola

@Signal:28,1,Park Light,0

#Value:0,Inactive

#Value:1,Active

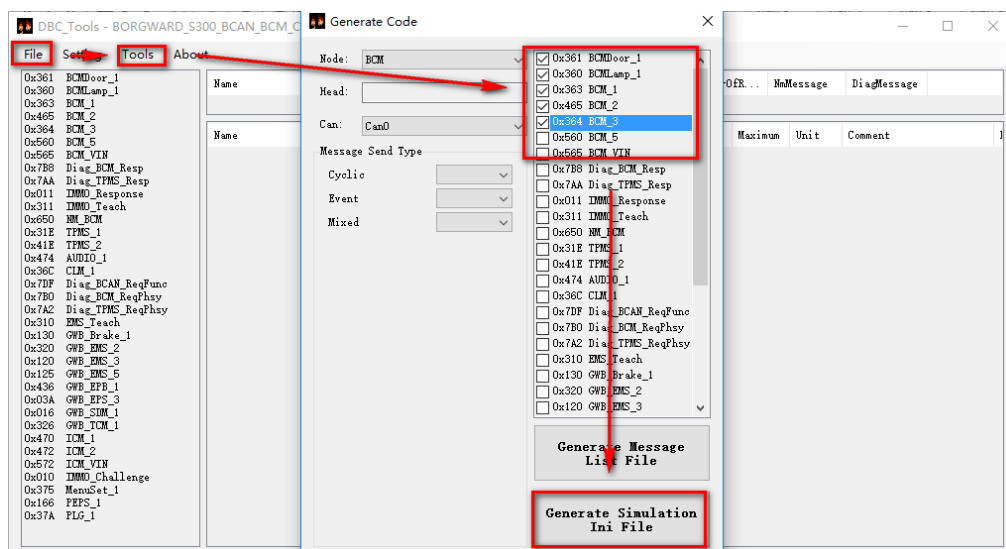
@Signal:40,15,BR1_N_VehicleSpeed,0

#PhyValue:10000,0.01, 0,km/h

Chapter VIII DBC 生成 message.ini

从 DBC 生成 message.ini 文件需要借助 DBC_Tools.exe 工具

步骤:



Step1: 打开 DBC_Tools.exe 工具

Step2: 菜单 File→Load Dbc File 打开 DBC 文件

Step3: 菜单 Tools→Generate Code

Step4: 勾选需要生成的 CAN 消息帧

Step5: 点击 Generate Simulation Init File 按钮

注意: DBC_Tools 暂不支持 CANFD, 解析 bitstart 会有错误

Chapter IX 联网和授权

1. 联网功能：

uds.exe 从 2.7.0.0 版本开始，支持自动联网自动升级功能。

- 1) 每次打开 uds 上位机时，如果发现处于公司内网中，将自动获取最新 uds 版本和最新的安全算法，并弹窗提示是否升级。如果点击升级，则会触发下载并将文件替换为最新版本文件，之后自动重启。
- 2) 如果发现未处于公司内网中，将尝试连接公网服务器获取最新 uds 版本，并提示升级。注：公网下仅升级程序，不会自动升级安全算法。

2. 授权功能：

uds.exe 从 2.8.0.0 版本开始，增加授权功能，未经授权的电脑，将无法使用此工具。

注：首次打开工具时，检查到电脑连接了公司内网，则自动从内网中获取授权。

注：更新上位机后，需要重新获取授权，请谨慎升级。

注：授权和电脑时间强绑定，修改系统时间可能会导致授权过期

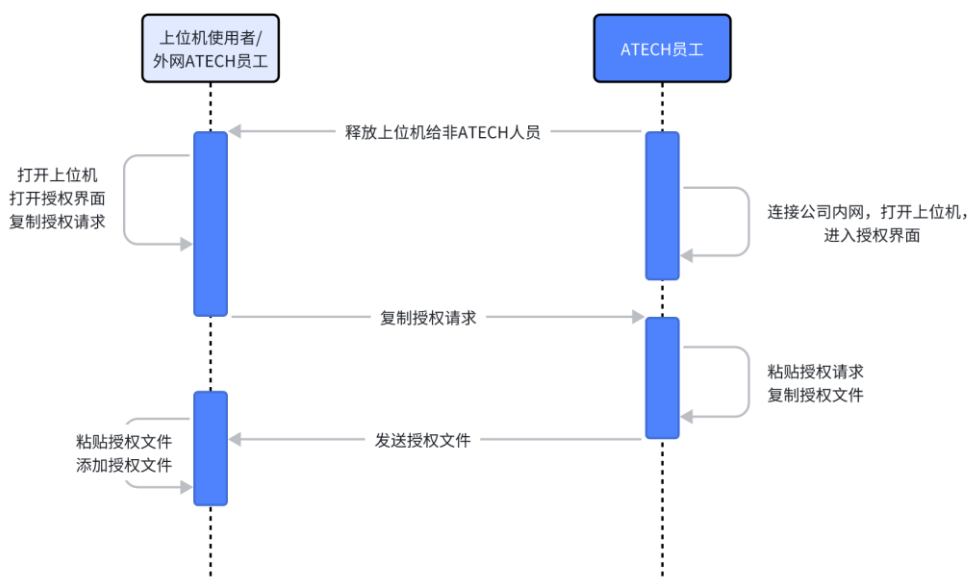
2.1. 未检测到授权

未授权的电脑，打开 uds 上位机时，将会弹窗提示未检测到授权，点击确认后将弹出授权管理界面，如下图。



2.2. 添加授权

添加授权文件过程如下:



下图为打开授权管理的方法:

注：仅在公司内网中，且可以正常连接内网服务器情况下才可以进入下图的授权文件生成界面，否则仅能进入授权请求生成界面

