

USBCAN-OBDD

工业级USB-CAN转换器

用户手册



文档版本：V4.10 （2020/12/13）

修订历史

版本	日期	原因
V1.00	2013/6/16	创建文档
V2.01	2013/12/20	修正设备工作参数
V3.01	2015/04/22	添加部分参数
V3.50	2016/07/16	添加 OBDII 功能
V4.01	2017/01/13	添加 CANopen 功能
V4.10	2020/12/13	修正文档

目 录

1. 功能简介.....	3
1.1 功能概述.....	3
1.2 性能特点.....	3
1.3 典型应用.....	4
2. 设备安装.....	5
2.1 设备尺寸.....	5
2.2 接口定义及功能.....	5
2.3 驱动及软件安装.....	6
3. 设备使用.....	7
3.1 与 PC 连接.....	7
3.2 与 CAN-bus 连接.....	7
3.3 CAN 总线终端电阻.....	8
3.4 系统状态指示灯.....	8
4. ECANTools 软件使用.....	10
4.1 软件启动.....	10
4.2 数据收发.....	11
4.3 总线分析功能.....	11
4.4 汽车数据解析功能.....	12
4.5 曲线 Curver 功能说明.....	15
4.6 其他功能.....	17
5. Linux 系统使用说明.....	19
6. 二次开发.....	23
7. 技术规格.....	24
8. 常见问题.....	25
9. 免责声明.....	28
附录 1: CAN2.0B 协议帧格式.....	29
附录 2: ISO15765 协议数据与 PID 对应关系.....	31
销售与服务.....	34

1. 功能简介

1.1 功能概述

USBCAN-OBDD 是集成 1 路 CAN 接口的便携式 CAN-bus 总线通讯接口卡。该型号 CAN 卡是我公司专为汽车电子开发或汽车故障诊断用户设计,使用 USBCAN-OBDD 便携式 CAN 接口卡,PC 可以通过 USB 接口快速连接至汽车 OBD 接口中的 CAN-bus 网络,使 PC 可以构成汽车 CAN-bus 网络中数据处理、数据采集的 CAN-bus 网络控制节点。

USBCAN-OBDD 便携式 CAN 接口卡是 CAN-bus 产品开发、CAN-bus 数据分析的强大工具;同时具有体积小、即插即用等特点,也是便携式系统用户的最佳选择。USBCAN-OBDD 接口卡上自带 USB 连接线缆,集成 CAN 接口电气隔离保护模块,使其避免由于瞬间过流/过压而对设备造成损坏,增强系统在恶劣环境中使用的可靠性。

USBCAN-OBDD 便携式 CAN 接口卡支持 WindowsXP/Win7/Win8/Win10 等 32 位/64 位操作系统,还可支持各版本 Linux 操作系统。我公司为用户提供统一的应用程序编程接口和完整的应用示范代码,含 VC、VB、.Net、Delphi7、Labview 和 C++Builder 等开发例程示范,方便用户进行应用程序开发。

USBCAN-OBDD 接口卡可使用我公司自主开发的 ECANTools 通用测试软件,可执行 CAN-bus 报文的收发和监测等功能。

1.2 性能特点

- PC接口符合USB2.0全速规范,兼容USB1.1及USB3.0;
- 集成1路CAN-bus接口, OBD II 接口方式;
- 支持CAN2.0A和CAN2.0B帧格式,符合ISO/DIS 11898规范;
- CAN-bus通讯波特率在5Kbps~1Mbps之间任意可编程;
- 使用USB总线电源供电 (DC+5V, 130mA);
- CAN-bus接口采用电气隔离,隔离模块绝缘电压: DC 1500V;
- 最高接收数据流量: 14000 fps;
- CAN端接收报文时间戳精度可达1us;

- 支持WinXP、Win7、Win8、Win10等Windows操作系统；
- 支持Linux操作系统；
- 支持ECANTools测试软件；
- 工作温度范围：-40℃~+85℃；

1.3 典型应用

- CAN-bus网络诊断与测试；
- 汽车数据解码工具；
- OBD等汽车电子设备开发；
- 汽车故障诊断分析；
- ECU数据模拟。

2. 设备安装

本章介绍了 USB-CAN 接口卡与电脑连接的方法及初次使用电脑连接 USB-CAN 接口卡时的注意事项。

2.1 设备尺寸

设备外形尺寸：（长）88mm * （宽）43 mm * （高）20mm，其示意图如图 2.1 所示。

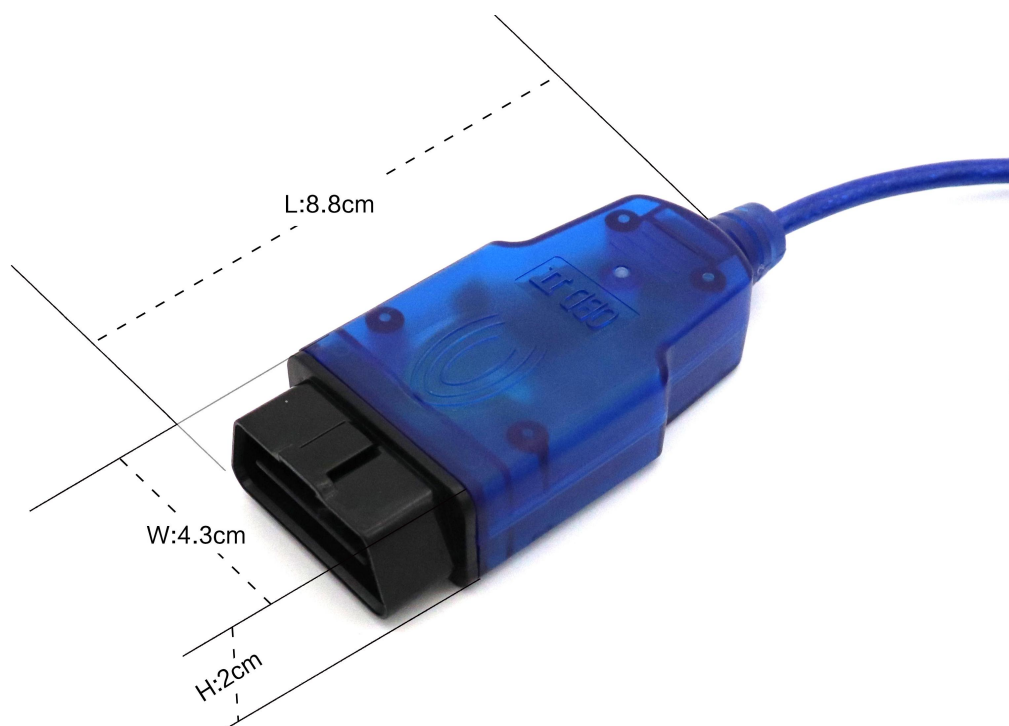


图 2.1 USBCAN-OBDD 外形尺寸

2.2 接口定义及功能

USBCAN-OBDD 口卡集成 1 路 CAN 通道，由 1 个 OBD II 接口引出，可以用于连接 1 路 CAN-bus 网络或者 CAN-bus 接口的设备。

USBCAN-OBDD 各接口位置及定义如图 2.2、表 2.1 所示。



图 2.2 USBCAN-OB 接口位置

引脚	端口	名称	功能
6	OBD II	CAN_H	CAN_H 信号线
14		CAN_L	CAN_L 信号线
其他		NC	无连接

表 2.1 USBCAN-OB 接口卡的 CAN-bus 信号分配

2.3 驱动及软件安装

驱动及软件安装之前，请用户确保自己登陆 Windows 的账户是超级管理员，或用户账户有安装驱动及软件的相关权限，否则可能导致安装失败。

确认 Windows 账户权限的方法：控制面板-用户账户。

2.3.1 驱动及软件安装

用户可以通过直接安装 ECANTools 软件的方式，完成驱动及软件的打包安装。如需手动安装驱动，请进入光盘中的“驱动 driver”文件夹，选择与系统对应（32/64 位）的安装文件（DriverSetup.exe/DriverSetup64.exe）进行手动安装。

2.3.2 驱动及软件卸载

用户可以通过运行上方 DriverSetup.exe/DriverSetup64.exe 后点击“卸载”按钮卸载安装好的设备驱动。

用户可通过“添加/删除程序”中找到 ECANTools 软件对其进行卸载。

3. 设备使用

3.1 与 PC 连接

USBCAN-OBDD 接口卡具有即插即用的特点，因此用户可以使用 PC 机的 USB 接口直接向 USBCAN-OBDD 接口卡供电。

3.1.1 USB 总线供电模式

USB 总线供电模式适合于大多数应用场合，例如，当 USBCAN-OBDD 接口卡是 USB 端口连接的唯一设备时。

将 PC 与 USBCAN-OBDD 接口卡的 USB 端直接与电脑的 USB 接口连接，由 USB 接口向 USBCAN-OBDD 接口卡提供+5V 电源；此时，指示灯点亮，表示设备工作正常且处于待连接状态。

3.2 与 CAN-bus 连接

USBCAN-OBDD 接入 CAN 总线连接方式：将 CAN_H 连 CAN_H，CAN_L 连 CAN_L，或直接将插头插入汽车 OBD 接口中即可建立通信。

CAN-bus 网络采用直线拓扑结构，总线最远的 2 个终端需要安装 $120\ \Omega$ 的终端电阻；如果节点数目大于 2，中间节点不需要安装 $120\ \Omega$ 的终端电阻。对于分支连接，其长度不应超过 3 米。CAN-bus 总线的连接见图 3.1 所示。

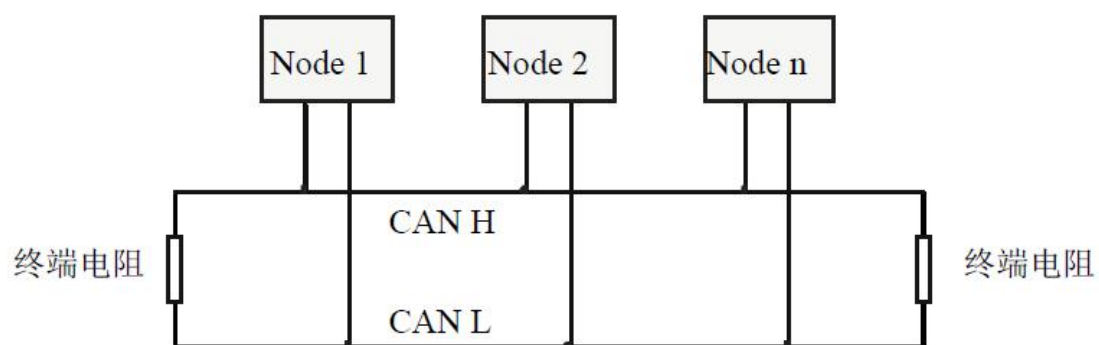


图 3.1 CAN-bus 网络的拓扑结构

注意：CAN-bus 电缆可以使用普通双绞线、屏蔽双绞线。理论最大通信距离主要取决于总线波特率，最大总线长度和波特率关系详见表 3.1。若通讯距离超过 1Km，应保证线的截面积大于 $\Phi 1.0\text{mm}^2$ ，具体规格应根据距离而定，常规是随距离的加长而适当加大。

波特率	总线长度
1 Mbit/s	25m
500 kbit/s	100m
250 kbit/s	250m
125 kbit/s	500m
50 kbit/s	1.0km
20 kbit/s	2.5km
10 kbit/s	5km
5 kbit/s	13km

表 3.1 波特率与最大总线长度参照表

3.3 CAN 总线终端电阻

为了增强CAN通讯的可靠性，消除CAN总线终端信号反射干扰，CAN总线网络最远的两个端点通常要加入终端匹配电阻，如图3.2所示。终端匹配电阻的值由传输电缆的特性阻抗所决定。例如双绞线的特性阻抗为 $120\ \Omega$ ，则总线上的两个端点也应集成 $120\ \Omega$ 终端电阻。USBCAN-OBD接口卡采用82C251收发器，如果网络上其他节点使用不同的收发器，则终端电阻须另外计算。

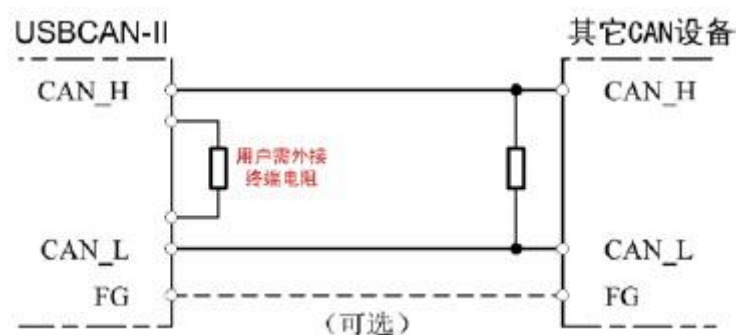


图 3.2 USBCAN 与其他 CAN 节点设备连接

注意：USBCAN-OBD内部未集成 $120\ \Omega$ 终端电阻，如需在此节点增加终端电阻，则需在连接线上集成终端电阻即可。

3.4 系统状态指示灯

USBCAN-OBD 接口卡具有 1 个双色指示灯用来指示设备的运行状态。这个指示灯的具体指示功能见表 3.2，这个指示灯处于各种状态下时，CAN 总线的状态

指示灯	颜色	指示状态
SYS	红/绿	电源指示及系统、数据指示

表 3.2 USBCAN-OBDDiv data-bbox="145 255 850 318" data-label="Text">

USBCAN-OBDDiv data-bbox="145 333 848 374" data-label="Text">

USB 接口连接正常后，软件打开设备后，SYS 灯会慢闪，当 USB 总线有数据在传输时，USB 信号指示灯 SYS 会快速闪烁。

指示灯	状态	指示状态
SYS	红	电源供电正常，系统初始化
	绿	设备驱动正常，等待连接
	绿灯慢闪	设备与软件正常连接
	绿灯快闪	CAN总线有数据收发
	红灯闪烁	CAN总线上有错误帧

表 3.3 USBCAN-OBDDiv data-bbox="158 909 286 928" data-label="Page-Footer">

产品数据手册

4. ECANTools 软件使用

ECANTools 软件是我公司针对 Windows 平台开发的专用调试分析软件，使用该软件，用户可以直观，快速的进行 CAN 总线数据收发。软件极易使用且扩展功能非常丰富，用户可以很快上手使用。

4.1 软件启动

1. 如用户已安装 ECANTools 软件，可以通过双击桌面上的“ECANTools”图标打开软件。



2. 选择对应的设备类型后，点击“打开设备”即可在设备列表中出现已经插入电脑 USB 接口的 CAN 设备。

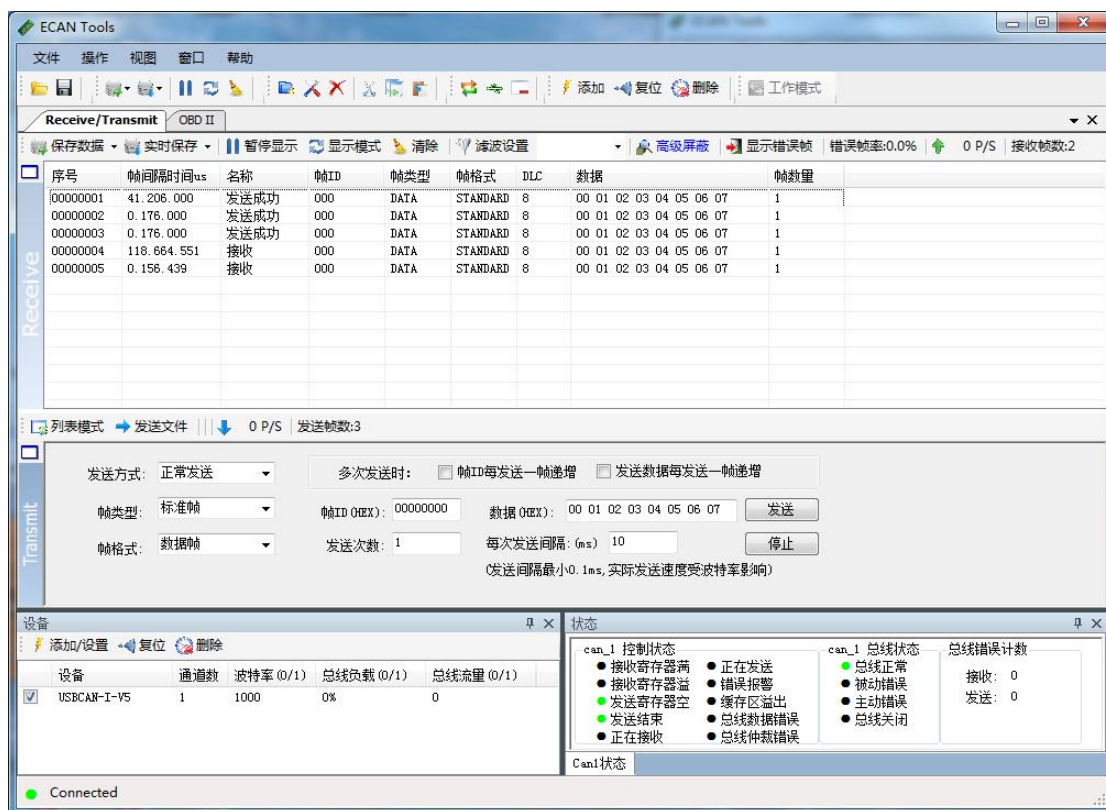
3. 选择工作模式。软件提供 3 种工作模式可选：正常模式、只听模式、自发自收模式。正常模式适用于软件数据收发；只听模式适用于在不干扰总线的情况下监听总线，但不可发送数据，此时 USBCAN 设备不作为 CAN 总线节点，不发送应答和时钟信号；自发自收模式用于测试 USBCAN 设备的 CAN 通信是否正常。

4. 选择波特率。用户需要按实际接入的总线的波特率进行设置，如波特率与总线不匹配，则无法正常通信，甚至干扰总线。如需自定义波特率，请与我公司

联系。当用户并不知道总线波特率时，也可使用自动识别波特率功能进行扫描。

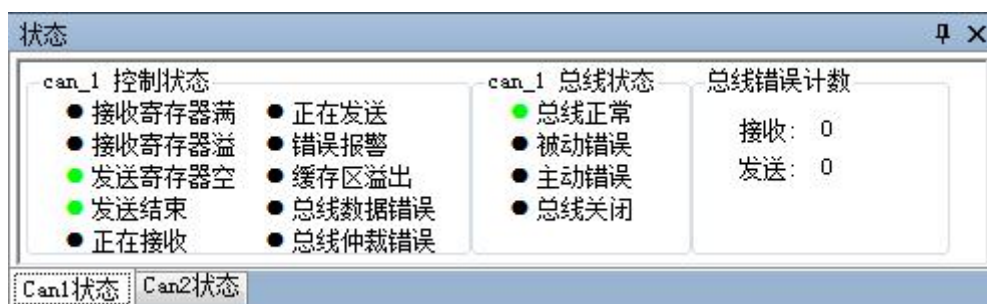
4.2 数据收发

接收和发送界面是ECANTools的主要功能，在此界面，用户可以直观的看到接收到的CAN报文，以及通过普通发送或列表发送的方式，向总线上发送数据。



4.3 总线分析功能

具有总线诊断功能，可以检测到总线错误帧以及总线仲裁丢失位，实现了比较全面的CAN总线分析。



CAN总线状态显示：指示当前CAN总线状态包括：总线正常、被动错误、主动错误、总线挂起。

CAN控制器内部FIFO溢出:某一段时间内的总线报文过于密集,导致USBCAN来不及接收,发生报文丢失。

CAN控制器错误报警:总线发生多次发送错误或接收错误,错误计数器超出报警阈值,并能显示错误计数值。

CAN控制器消极错误:总线发生多次发送错误或接收错误,导致CAN控制器进入错误消极状态,并能显示错误计数值。

CAN控制器仲裁丢失:本节点与其他设备在竞争总线时失败,待发报文将在下一次总线空闲时尝试发送,能捕捉到相应的错误位信息并显示出来。

CAN控制器总线错误:总线发生一次发送错误或接收错误,错误计数器的值将累加,并能捕捉错误的位信息,比如ACK、CRC、位界定错误信息。

CAN数据缓冲区溢出:由于一段时间未与PC交换数据,导致设备存储的报文数目超出硬件容量限制,新接收的报文将覆盖最早的报文。

4.4 汽车数据解析功能

用户可以使用 ECANTools 软件的汽车信号解析功能。使用该设备接入汽车 OBD 接口即可将汽车内部部分传感器的实际数值解析出来,信号解析功能仅支持家用汽油车 ISO15765 协议。车速、转速、水温可通过软件仪表盘显示。



ECANTools 软件还可以解析 ISO15765 协议中规定的汽车传感器数据, 包括: 发动机转速、冷却液温度、车辆速度、电压、进气歧管压力、进气温度、空气流速、节气门位置、氧传感器电压、燃油压力等。并且以上这些数据的数值变化可以实时保存在电脑中。

使用时, 请首先点选 11Bit 按钮, 勾选“循环获取实时数据”。若此时 USBCAN 分析仪连接到汽车动力总线上, 则将返回车速、转速、水温等参数。若此时无数据, 可切换至 29Bit 重新尝试。若两种模式皆无反馈, 请检查通信波特率及接线。大多数汽车动力总线波特率为 500Kbps。

PID 手动获取功能:

您可通过输入 PID 的值来手动获取车辆的各种参数。请注意, 需要首先选取正确的发送命令类型, 即 11Bit 或 29Bit。详细的 PID 值(十进制)如下表所示。

PID	数据头	最小值	最大值	单位
01	当前故障码数量	0	127	个
04	计算负荷值	0	100	%
05	发动机冷却液温度	-40	215	°C
06	短时燃油修正(气缸列 1 和 3)	-100	99.22	%
07	长期燃油修正(气缸列 1 和 3)	-100	99.22	%
08	短时燃油修正(气缸列 2 和 4)	-100	99.22	%
09	长期燃油修正(气缸列 2 和 4)	-100	99.22	%
010	燃油压力计量	0	765	kPa
011	进气歧管绝对压力	0	255	kPa
012	发动机转速	0	9999	rpm
013	车速	0	255	km/h
014	第一缸点火正时提前角(不包括机械提前)	-64	63.5	°
015	进气温度	-40	215	°C

016	空气流量传感器的空气流量	0	655.35	g/s
017	绝对节气门位置	0	100	%
031	自发动机起动的时间	0	65535	s
033	在 MIL 激活状态下行驶的里程	0	65535	km
034	相对于歧管真空度的油轨压力	0	5177.265	kPa
035	相对于大气压力的油轨压力	0	655350	kPa
044	EGR 指令开度	0	100	%
045	EGR 开度误差 (实际开度-指令开度)/指令开度	-100	99.22	%
046	蒸发冲洗控制指令	0	100	%
047	燃油液位输入	0	100	%
048	自故障码被清除之后经历的暖机循环个数	0	255	N/A
049	自故障码被清除之后的行驶里程	0	65535	km
050	蒸发系统的蒸气压力	-8192	8192	Pa
051	大气压	0	255	kPa
060	催化器温度 B1S1	-40	6513.5	℃
061	催化器温度 B2S1	-40	6513.5	℃
062	催化器温度 B1S2	-40	6513.5	℃
063	催化器温度 B2S2	-40	6513.5	℃

066	控制模块电压	0	65.535	V
067	绝对负荷值	0	25700	%
068	等效比指令	0	2	N/A
069	相对节气门位置	0	100	%
070	环境空气温度	-40	215	°C
071	绝对节气门位置 B	0	100	%
072	绝对节气门位置 C	0	100	%
073	加速踏板位置 D	0	100	%
074	加速踏板位置 E	0	100	%
075	加速踏板位置 F	0	100	%
076	节气门执行器控制指令	0	100	%
077	MIL 处于激活状态下的发动机运转时间	0	65535	min
078	自故障码清除之后的时间	0	65535	min

实时保存 TXT 文件功能:

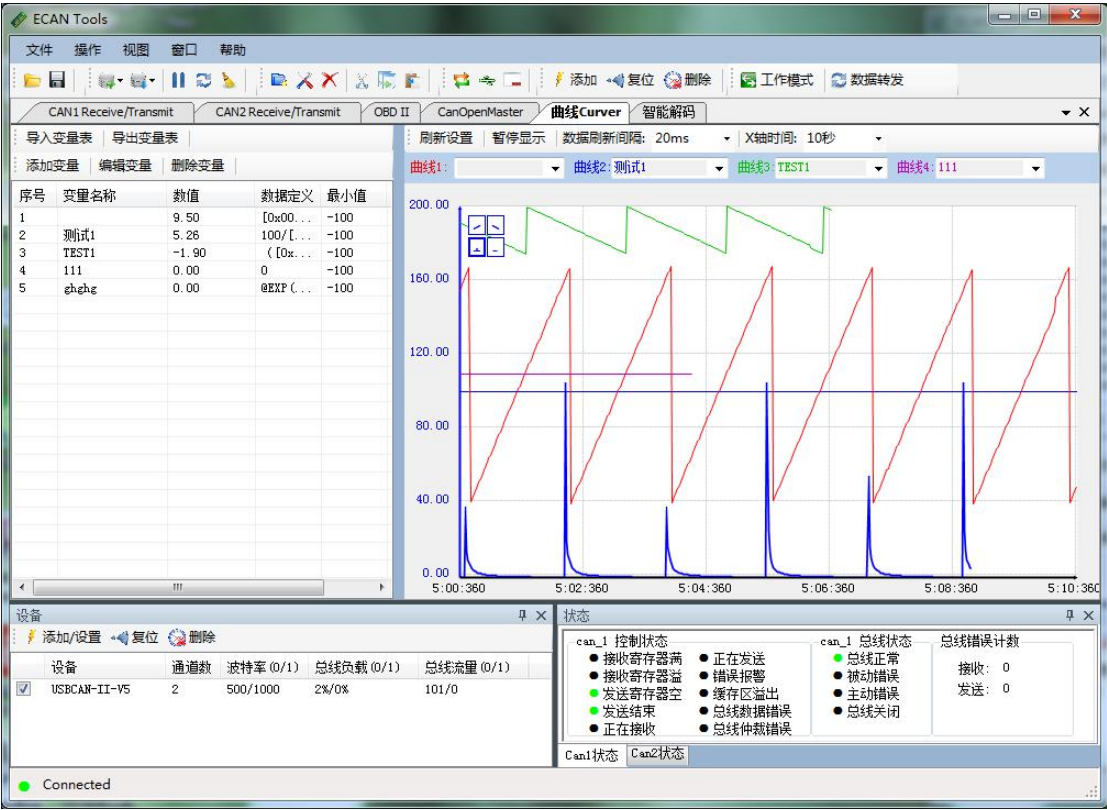
用户可以通过此功能直观的看出汽车内部部分传感器的具体数值并基于这些数据诊断汽车各传感器状态是否正常。用户还可直观的比较某些传感器数值的变化规律和多种传感器数值之间进行比较。

4.5 曲线 Curver 功能说明

数据数值变化也可通过曲线实时显示其变化轨迹，如下图所示。用户最多可选择四个变量在同一界面同时显示。

用户可以通过曲线功能选择最多四个您所需要的变量，并通过观察曲线的方式实时了解每个数据的变化，并且还可直观的比较多个数值之间的相对变化，可用于调整发动机怠速时转速、扭矩等。

在使用曲线功能时请注意，将变量设置好后再接收数据，数据接收时新建的变量无法在曲线中显示







点击“添加变量”“编辑变量”即可添加和编辑变量，这些变量可以对接接收的 CAN 总线的原始数据进行简单的数学运算，并在以曲线的方式显示出来。可以添加多组变量，变量名称可包含字母，数字，汉字，空格和下划线，方便识别。

曲线功能中可提供的特殊运算如下表所示，**请严格按照运算形式书写！**

运算符	运算功能	书写格式（以正弦为例）： 使用特殊运算符时，运算符括号内需要有至少一则运算。例如： @SIN(数据源) 是错误的书写方式，会提示非数字，正确的书写方式为 @SIN(数据源*1)
sin/cos/tan	正弦/余弦/正切	
sqr	求一个非负实数算数平方根	
log	求一个数以 e 为底的对数 (ln x)	
abs	求一个数的绝对值	
exp	求以 e 为底的指数函数的值 (e ^x)	
Max	变量当前最大值	
Min	变量当前最小值	



“添加数据源”就是将收到的 CAN 总线原始数据中某一帧的某一字节的数据作为数据源添加到变量计算定义中，添加时先点击“添加数据源”，然后依次填充“CAN ID”、“标准帧（扩展帧）”、“数据”等信息，点击确定，会在“变量计算定义”中生成一条相应的语句，该语句也可自行编写，编写时请遵循该格式。

在曲线界面中，“”功能是切换到上一条曲线，“”功能是切换到下一条曲线；“”功能是将当前颜色曲线坐标轴上移，“”功能是将当前颜色曲线坐标轴下移。

4.6 其他功能

ECANTools 软件扩展功能极为丰富，且都很方便易用，灵活运用这些扩展功能可以帮助用户方便快速的分析数据。



保存数据/实时保存：可保存接收列表，保存格式有：txt、dat、can、excel。

暂停显示：可暂停接收列表的滚动。

显示模式：可选滚动模式和列表模式，列表模式可按规则将数据实时分类。

清除：可清除当前接收列表中的数据。

滤波设置：通过编辑滤波 ID 的方式，实现多段滤波。

高级屏蔽：通过屏蔽 ID 的方式，不显示已屏蔽的 ID。

显示错误帧：可显示/隐藏总线上的错误帧。

OBDII 插件：可直观的解析部分汽车参数。

CANopen 主站：方便用户进行 CANopen 协议解析。

请注意：ECANTools 软件更多功能详述请查看“ECANTools 软件使用说明”

5. Linux 系统使用说明

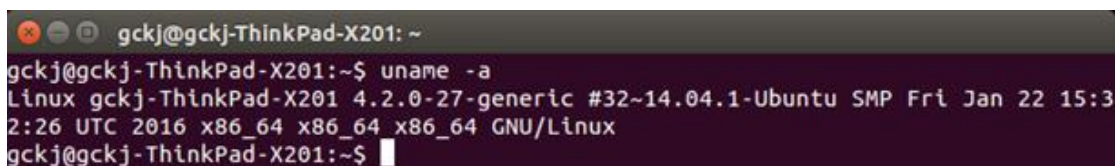
广成科技 USBCAN 分析仪系列产品支持各种版本 Linux 操作系统，我公司会向客户提供 32/64 位 Linux 系统驱动及二次开发相关文档，用户可自行在 Linux 系统中开发使用。

Linux 系统中使用我公司设备的大体方法如下：①获取系统管理员权限；②拷贝必要的文件到系统 GCC 编译目录中；③切换目录到 USBCAN 驱动文件夹进行编译；④运行测试程序。具体操作方法如下：

1. 查询 linux 版本号，确认系统类型（32/64 位）。

输入：uname -a

（由结果可知我们 Linux 系统类型是 64 位的）



```
gckj@gckj-ThinkPad-X201: ~  
gckj@gckj-ThinkPad-X201:~$ uname -a  
Linux gckj-ThinkPad-X201 4.2.0-27-generic #32~14.04.1-Ubuntu SMP Fri Jan 22 15:32:26 UTC 2016 x86_64 x86_64 x86_64 GNU/Linux  
gckj@gckj-ThinkPad-X201:~$
```

2. 确定 Linux 系统类型后，拷贝对应的 USBCAN 驱动文件到系统中。

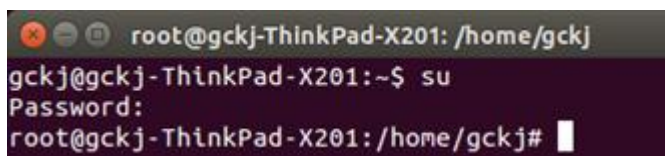
（此例子中我们把驱动文件放到系统桌面）



3. 获取管理员权限，以便于后续安装驱动等操作。

输入：su

（输入 su 指令后要求输入管理员密码，输入正确的密码即可获取管理员权限）



```
root@gckj-ThinkPad-X201: /home/gckj  
gckj@gckj-ThinkPad-X201:~$ su  
Password:  
root@gckj-ThinkPad-X201: /home/gckj#
```

4. 进入 USBCAN 驱动文件夹，拷贝 libusb.so、libusb-1.0.so、libECanVci.so.1 到 gcc 编译库目录下。（默认路径为/usr/lib）

输入：cp libusb.so libusb-1.0.so libECanVci.so.1 /usr/lib

(默认路径为/usr/lib)

```
root@gckj-ThinkPad-X201:/home/gckj# cd Desktop/linux64
root@gckj-ThinkPad-X201:/home/gckj/Desktop/linux64# cp libusb.so libusb-1.0.so 1
libECanVci.so.1 /usr/lib
root@gckj-ThinkPad-X201:/home/gckj/Desktop/linux64#
```

5. 进入 gcc 编译库文件夹, 把 libECanVci.so.1 和 libECanVci.so 链接到一起。

输入: `ln -sv libECanVci.so.1 libECanVci.so`

```
root@gckj-ThinkPad-X201:/usr/lib# ln -sv libECanVci.so.1 libECanVci.so
'libECanVci.so' -> 'libECanVci.so.1'
root@gckj-ThinkPad-X201:/usr/lib#
```

6. 再次进入 USBCAN 驱动文件夹, 编译。

输入: `make`

```
root@gckj-ThinkPad-X201:/# cd /home/gckj/Desktop/linux64
root@gckj-ThinkPad-X201:/home/gckj/Desktop/linux64# make
removed 'test'
gcc -o test test.c -lpthread -lECanVci -lusb
root@gckj-ThinkPad-X201:/home/gckj/Desktop/linux64#
```

7. 运行测试程序测试 USBCAN 收发。

输入: `./test`

```
root@gckj-ThinkPad-X201:/home/gckj/Desktop/linux64# ./test
test [DevType] [DevIdx] [ChMask] [Baud] [TxType] [TxSleep] [TxFrames]
example: test 16 0 3 0x1400 0 1 1000
          |  |  |  |  |  |  |
          |  |  |  |  |  |  |1000 frames / channel
          |  |  |  |  |  |  |tx > sleep(3ms) > tx > sleep(3ms) ....
          |  |  |  |  |  |  |
          |  |  |  |  |  |  |0-normal, 1-single, 2-self_test, 3-single_self_
test, 4-single_no_wait....
          |  |  |  |  |  |  |
          |  |  |  |  |  |  |0x1400-1M, 0x1c03-125K, ....
          |  |  |  |  |  |  |
          |  |  |  |  |  |  |bit0-CAN1, bit1-CAN2, bit2-CAN3, bit3-CAN4, 3=CAN1+CAN2,
7=CAN1+CAN2+CAN3
          |  |  |  |  |  |  |
          |  |  |  |  |  |  |Card0
          |  |  |  |  |  |  |
          |  |  |  |  |  |  |1-usbcan, ....
root@gckj-ThinkPad-X201:/home/gckj/Desktop/linux64#
```

输入./test 后会出现系统提示及例子, 其中:

第一位 (16): 设备类型, 单通道设备输入 3, 双通道输入 4;

第二位 (0): 设备索引号, 当只接入一台 USBCAN 时为 0;

第三位 (3): 打开第几路 CAN, 打开 CAN1 输入 1, 打开 CAN2 输入 2 同时打开 CAN1 和 CAN2 输入 3;

第四位 (0x1400): 设置 CAN 总线波特率, 0x1400 表示波特率为 1000K, 其他波特率的值详见“EcanVCI 动态库使用手册”;

第五位 (0): 工作模式, 0 为正常模式, 其他工作模式详见“EcanVCI 动态库使用手册”;

第六位 (1): 发送时间间隔, 单位 ms;

第七位 (1000): 发送次数。

8. 运行测试程序后即可使用其他 USBCAN 设备收到他发出的数据

```
root@gckj-ThinkPad-X201: /lib/test/linux64
Receive returned: time ~= 2 seconds
Receive returned: time ~= 3 seconds
<ENTER> to start TX: 1000 frames/channel, baud: t0=0x00, t1=0x14...
1
TX stopped, <ENTER> to terminate RX-threads...
CAN0 RX thread terminated, 0 frames received & verified: no error
CAN1 RX thread terminated, 0 frames received & verified: no error

***** 1000 frames/channel transferred, 2 seconds elapsed *****
performance: 500 frames/channel/second
CloseDevice
root@gckj-ThinkPad-X201:/lib/test/linux64# ./test 4 0 3 0x1400 0 1 1000
DevType=4, DevIdx=0, ChMask=0x3, Baud=0x1400, TxType=0, TxSleep=1, TxFrames=0x00
0003e8(1000)
OpenDevice succeeded
InitCAN(0) succeeded
StartCAN(0) succeeded
InitCAN(1) succeeded
StartCAN(1) succeeded
Receive returned: time ~= 1 seconds
Receive returned: time ~= 2 seconds
Receive returned: time ~= 3 seconds
<ENTER> to start TX: 1000 frames/channel, baud: t0=0x00, t1=0x14...
```

ECAN Tools										
文件 操作 视图 窗口 帮助										
CAN1 Receive/Transmit CAN2 Receive/Transmit OBD II CanOpenMaster										
保存数据 实时保存 暂停显示 显示模式 清除 滤波设置 高级屏蔽 显示错误帧 错误帧率:0										
序号	帧间隔时间us	名称	帧ID	帧类型	帧格式	DLC	数据	帧数量		
00000381	001.000.664	接收	0EE5DCBB	DATA	EXTENDED	5	B6 87 1B 64 F5	1		
00000382	001.000.595	接收	0A754EA9	DATA	EXTENDED	7	E5 02 A8 11 77 4D CD	1		
00000383	001.000.570	接收	397	DATA	STANDARD	4	74 A1 68 2A	1		
00000384	001.000.711	接收	493	DATA	STANDARD	5	CA 94 5C E8 79	1		
00000385	001.000.546	接收	1AB8576A	DATA	EXTENDED	8	B7 19 DF 18 8D 69 8E 69	1		
00000386	001.000.561	接收	106A0D41	DATA	EXTENDED	6	9B 43 61 84 BC C0	1		
00000387	001.000.627	接收	337	DATA	STANDARD	4	02 C4 22 D3	1		
00000388	001.000.571	接收	04E09C13	DATA	EXTENDED	5	9D 66 CF E0 C7	1		
00000389	001.000.710	接收	0F3DE7BC	DATA	EXTENDED	8	00 D0 01 E6 C4 03 AA B6	1		
00000390	001.000.520	接收	596	DATA	STANDARD	6	CD 8D 30 6A 15 99	1		
00000391	001.000.574	接收	0CC99933	DATA	EXTENDED	2	6E 5D	1		
00000392	001.000.616	接收	079	DATA	STANDARD	1	79	1		
00000393	001.000.646	接收	1566ACD5	DATA	EXTENDED	5	9C 83 EA CD ED	1		
00000394	001.000.707	接收	1E5BCB79	DATA	EXTENDED	4	95 6F 74 F7	1		
00000395	001.000.501	接收	041	DATA	STANDARD	1	41	1		
00000396	001.000.612	接收	302	DATA	STANDARD	4	EF 6B 38 BE	1		
00000397	001.000.575	接收	02DC5B8B	DATA	EXTENDED	4	F2 37 F5 BB	1		
00000398	001.000.565	接收	1542A855	DATA	EXTENDED	1	55	1		

6. 二次开发

我公司为二次开发的用户提供标准的接口函数库，包括：ECANVCI.h、ECANVCI.lib、ECANVCI.dll。该接口函数库均为标准格式，用户可以在 VC、VB 等编程环境中，对这些接口函数声明调用，具体使用方法详见“ECAN 动态库使用手册”。图 5.1 为常用结构体名称及函数库调用流程。

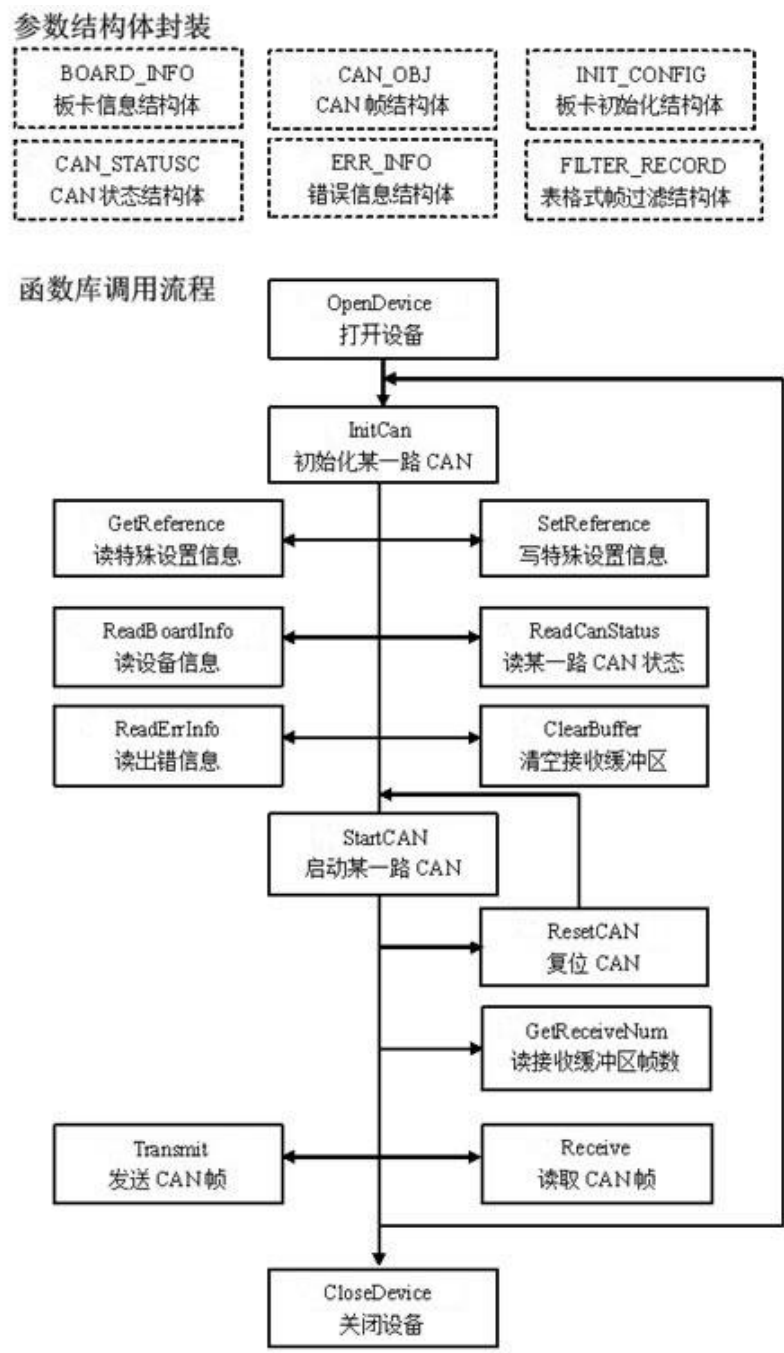


图 5.1 二次开发流程

7. 技术规格

连接方式	
PC端	USB接口
CAN端	OBD II接口
接口特点	
USB接口	USB2.0全速接口，兼容USB1.1，USB3.0
CAN接口	遵循ISO 11898标准，支持CAN2.0A/B
CAN波特率	5Kbit/s~1Mbit/s
电气隔离	1500V，DC-DC
CAN终端电阻	未集成
供电电源	
供电电压	+5V DC （USB接口）
供电电流	最大130mA
环境试验	
工作温度	-40℃~+85℃
工作湿度	15%~90%RH，无凝露
EMC测试	EN 55024:2011-09 EN 55022:2011-12
防护等级	IP 20
基本信息	
外形尺寸	88mm *43mm *22mm，线长1.5m
重量	110g

8. 常见问题

1. 在 ECANTools测试软件中，为何显示“打开设备错误”？

可能产生此类故障的原因是：设备类型选择不正确或USBCAN设备驱动没有正常安装。在PC的设备管理器中检查 USBCAN设备属性，看看有没有“!”或“?”在USBCAN设备前面；若有，则检查硬件/软件冲突，并重新安装 USBCAN设备驱动程序。

2. 是否一定需要使用 120Ω 终端匹配电阻？

建议 120 Ω 终端匹配电阻用于吸收端点反射，提供稳定的物理链路。当进行单节点的自发自收测试时必须连接该 120 Ω 的终端电阻构成回路，否则无法进行自发自收测试。USBCAN 高性能 CAN 接口卡内部已经连接有 120 Ω 的终端电阻。

3. 一台计算机能否安装多块USBCAN接口卡？

旧版的接口不支持多卡同时操作，但是目前的 USBCAN 接口卡，支持多达 8 个同一型号的 USBCAN 接口卡同时操作。

4. USBCAN-OBDD接口卡最高的数据转换率是多少？

USBCAN 接口卡的单一 CAN 通道最高支持 8000 fps 的 CAN 总线数据转换，这里提到的帧是指标准帧 8 个数据的数据帧，如果是小于 8 字节数据或者远程帧可能会更快。另外，最高数据流量会受 PC 性能的限制。

5. 为何CAN状态指示灯不亮？

因为 USBCAN 接口卡的所有操作是受 PC 机控制的，只有 PC 机发送了启动 CAN 通讯的命令后，CAN 状态指示灯才会有意义。

6. 为何调用接口函数时系统非法操作？

首先在使用接口函数时请认真阅读函数说明，保证输入参数合法，特别注意指针(地址)的传递，或参照提供的例子程序，倘若问题还是未能解决，可联系我们的技术支持。

7. USBCAN 接口卡的通讯波特率如何设置？

设备提供了一组常用的波特率的设置值，若要使用其他的波特率，请联系广成科技客服进行计算。需要注意：USBCAN 接口卡的 CAN 控制器使用 24MHz 时钟，用户自定义波特率时要根据该时钟频率进行计算。

8. 系统进入待机或睡眠状态是否影响接收？

会有影响。这时所有处理将停止，最大可能导致硬件接收缓冲溢出错误。若有程序打开设备将尝试阻止系统进入待机或睡眠状态，从而保证系统正常工作。

使用 USBCAN接口卡时，请禁止系统的待机和睡眠功能。

9. 如何处理应用中的错误？

错误主要分为函数调用错误和 CAN-bus 通讯错误两种。函数调用错误一般由参数错误引起，如：设备号超出范围，类型号错误等，用 Win32 函数 GetLastError 返回的错误号是 87，还有的是对未打开的设备进行操作，实际是对一个非法句柄操作，根据具体函数调用情况都有相应的 Win32 标准错误码提供，用户可以使用 GetLastError 进行错误分析，这部分除错工作一般应该在设计时完成。

对于 CAN-bus 通讯错误，一般由 CAN 网络引起，也可能因用户设置不当而引起，如：波特率设置不一致、没有启动 CAN 控制器便调用发送函数等。大部分错误已经在设备驱动中作了简单的处理，如果要进行更深层次的错误分析和处理，可以调用 ReadCANStatus 函数。

另外需要注意的是数据溢出中断错误，它的产生有两种可能：(1) 软件接收缓冲区溢出。这说明应用程序无法及时处理接收到的数据，这时用户应该优化应用程序或更改通讯策略。(2) 硬件接收缓冲区溢出。产生这种错误是由于接收端 PC 中断延迟太大而引起的。只能通过提升计算机性能或协调其余节点适当降低发送速度来解决。

10. 打开关闭设备要注意哪些事项？

USBCAN-OBDD 接口卡提供 1 个 CAN 端口，用户能够在同一程序中操作不同的端口。USBCAN-OBDD 接口卡不允许共享方式打开设备，同一个设备不可被不同进程通过调用 OpenDevice 函数多次打开。OpenDevice 和 CloseDevice 函数一般在应用程序初始化和退出时只需要调用一次。当关闭设备时若能当前端口不再使用，应该先调用 ResetCAN 函数使当前端口脱离 CAN 总线，设备驱动程序只会在最后一个设备句柄关闭时才自动调用 ResetCAN 退出 CAN 总线的连接。

11. 如何使用中断方式操作通讯卡？

USBCAN 接口卡不提供直接操作中断的接口，因为中断已经在驱动程序中处理了。需要在应用程序中操作中断的多数原因是：程序不知道数据何时能到达设备，需要得到一个接收消息的触发才能从缓冲读取数据。解决这个问题的一般手段是使用多线程(或多任务)。即启动一个新的线程，在线程中循环调用 Receive 函数来查询接收缓冲。Receive 内部已实现了阻塞机制，在缓冲里没有数据时会挂起调用线程，这时不会占用 CPU 的时间，应用程序仍然可以处理其他事务。

12. 如何更好的使用 Transmit 发送函数？

USBCAN 接口卡的驱动提供约 128 帧发送缓冲 FIFO，每次 Transmit 调用最多发送约 128 帧数据。发送设备的发送速度由当前计算机软硬件性能决定，一般连

续发送速度在 2000 fps 左右(标准数据帧 11Bytes, 1Mbps), 若发送速度过快将有可能使远端接收设备数据溢出而失去响应, 这样用户可在应用编程中适当添加延时以降低发送速度。

发送过程中每一帧都有超时限制, 单帧发送时超时时间约 2 秒, 一次发送多帧时最后一帧发送超时为 2 秒, 其余为 1 秒。发送超时一般由于 CAN 总线繁忙且当前节点优先级较低时发生, 并不是函数调用或通讯错误, 用户可以编程实现重发(一般中低速网络极少发生发送超时事件)。因此, 在系统设计时注意保证 CAN 总线占用不应该超过总线容量的 60-70%。

13. 如何更好的使用Receive函数?

设备驱动提供 100000 帧软件接收缓冲区, 这为应用编程人员提供了充足的反应处理时间。当软件接收缓冲数据溢出时设备驱动程序将调用 ResetCAN 复位 CAN 总线, 同时置位 CAN status 的数据溢出中断标志位, 注意软件缓冲溢出和 CAN 控制器硬件缓冲溢出都是使用该标志位。

接收函数提供 Wait 参数适合用于多线程编程, 函数内部封装一个阻塞函数, 其参数 Wait 含义如同 Win32 的 WaitForSingleObject 的 dwMilliseconds 参数(请参考 Win32API 说明), 它为 Receive 指定一个超时返回时间, 单位为毫秒。

当 Wait 为 0 时函数调用时立即返回当前成功读取到的帧数, 若接收缓冲为空则返回 0。当 Wait 非 0 时, 若函数调用时接收缓冲中已经有数据则马上返回成功读取的帧数, 若这时接收缓冲为空, 函数将等待一个指定的超时到达或接收到数据才返回成功接收的帧数。当 Wait 为 0xFFFFFFFF 时为无限等待直到有数据接收到, 建议不要把 Wait 设得过大, 无限等待更应该注意。

nFrames 等于 0 时函数实际是一个通知消息返回, 不要求读接收缓冲区, 是一个特殊的技巧性用法。注意: 若在主线程中调用 Receive 函数并且 Wait 非 0 则有可能引起应用程序暂时性的失去响应。若通过查询方式接收, 一般应该把 Wait 设为 0。

9. 免责声明

感谢您购买广成科技的 GCAN 系列软硬件产品。GCAN 是沈阳广成科技有限公司的注册商标。本产品及手册为广成科技版权所有。未经许可，不得以任何形式复制翻印。在使用之前，请仔细阅读本声明，一旦使用，即被视为对本声明全部内容的认可和接受。请严格遵守手册、产品说明和相关的法律法规、政策、准则安装和使用该产品。在使用产品过程中，用户承诺对自己的行为及因此而产生的所有后果负责。因用户不当使用、安装、改装造成的任何损失，广成科技将不承担法律责任。

关于免责声明的最终解释权归广成科技所有。

附录 1：CAN2.0B 协议帧格式

CAN2.0B 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	×	×	DLC（数据长度）			
字节 2	（报文识别码）				ID.10—ID.3			
字节 3	ID.2—ID.0			×	×	×	×	×
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，远程帧时无效。

CAN2.0B 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	×	×	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					×	×	×
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。

附录 2: ISO15765 协议数据与 PID 对应关系

PID	数据头	最小值	最大值	单位
01	当前故障码数量	0	127	个
04	计算负荷值	0	100	%
05	发动机冷却液温度	-40	215	℃
06	短时燃油修正(气缸列 1 和 3)	-100	99.22	%
07	长期燃油修正(气缸列 1 和 3)	-100	99.22	%
08	短时燃油修正(气缸列 2 和 4)	-100	99.22	%
09	长期燃油修正(气缸列 2 和 4)	-100	99.22	%
010	燃油压力计量	0	765	kPa
011	进气歧管绝对压力	0	255	kPa
012	发动机转速	0	9999	rpm
013	车速	0	255	km/h
014	第一缸点火正时提前角(不包括机械提前)	-64	63.5	°
015	进气温度	-40	215	℃
016	空气流量传感器的空气流量	0	655.35	g/s
017	绝对节气门位置	0	100	%
031	自发动机起动的时间	0	65535	s
033	在 MIL 激活状态下行驶的里程	0	65535	km
034	相对于歧管真空度的油轨压力	0	5177.265	kPa
035	相对于大气压力的油轨压力	0	655350	kPa

044	EGR 指令开度	0	100	%
045	EGR 开度误差 (实际开度-指令开度)/指令开度	-100	99.22	%
046	蒸发冲洗控制指令	0	100	%
047	燃油液位输入	0	100	%
048	自故障码被清除之后经历的暖机循环个数	0	255	N/A
049	自故障码被清除之后的行驶里程	0	65535	km
050	蒸发系统的蒸气压力	-8192	8192	Pa
051	大气压	0	255	kPa
060	催化器温度 B1S1	-40	6513.5	℃
061	催化器温度 B2S1	-40	6513.5	℃
062	催化器温度 B1S2	-40	6513.5	℃
063	催化器温度 B2S2	-40	6513.5	℃
066	控制模块电压	0	65.535	V
067	绝对负荷值	0	25700	%
068	等效比指令	0	2	N/A
069	相对节气门位置	0	100	%
070	环境空气温度	-40	215	℃
071	绝对节气门位置 B	0	100	%
072	绝对节气门位置 C	0	100	%

073	加速踏板位置 D	0	100	%
074	加速踏板位置 E	0	100	%
075	加速踏板位置 F	0	100	%
076	节气门执行器控制指令	0	100	%
077	MIL 处于激活状态下的发动机运转时间	0	65535	min
078	自故障码清除之后的时间	0	65535	min

销售与服务

沈阳广成科技有限公司



地址：辽宁省沈阳市浑南区长青南街 135-21 号 5 楼

邮编：110000

网址：www.gcgd.net

全国销售与服务电话：400-6655-220

售前服务电话与微信号：13889110770

售前服务电话与微信号：18309815706

售后服务电话与微信号：13840170070