受　控

# USBCAN-Modul 8

工业级USB-CAN转换器

用户手册




文档版本：V2.12 （2024/09/02）

## 修订历史

| 版本 | 日期 | 原因 |
|---|---|---|
| V1.00 | 2016/07/16 | 创建文档 |
| V1.50 | 2017/01/13 | 修改产品参数 |
| V1.60 | 2018/07/18 | 调整文档结构 |
| V2.00 | 2020/07/18 | 修改产品外观 |
| V2.10 | 2022/06/06 | 添加保存数据文件格式 |
| V2.11 | 2022/07/06 | 修正文档 |
| V2.12 | 2024/09/02 | 尺寸修正 |

# 目 录

1. 功能简介 ..... 4
1.1 功能概述 ..... 4
1.2 性能特点 ..... 4
1.3 典型应用 ..... 5
2. 设备安装 ..... 6
2.1 设备尺寸 ..... 6
2.2 接口定义及功能 ..... 6
2.3 驱动及软件安装 ..... 8
2.3.1 驱动及软件安装 ..... 8
2.3.2 驱动及软件卸载 ..... 8
3. 设备使用 ..... 8
3.1 与 PC 连接 ..... 9
3.2 与 CAN-bus 连接 ..... 9
3.3 CAN 总线终端电阻 ..... 10
3.4 系统状态指示灯 ..... 10
4. ECANTools 软件使用 ..... 12
4.1 软件启动 ..... 12
4.2 数据收发 ..... 13
4.3 总线分析功能 ..... 14
4.4 中继功能、脱机功能 ..... 15
4.5 其他功能 ..... 18
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 CAN2.0B 协议帧格式 ..... 29
销售与服务 ..... 31

# 1. 功能简介

## 1.1 功能概述

USBCAN-Modul 8 是集成 8 路 CAN 接口的高性能型 CAN-Bus 总线通讯接口卡。该型号 CAN 卡可兼容 USB2.0 总线全速规范，采用 USBCAN-Modul 8 高性能 CAN 接口卡，PC 可以通过 USB 接口快速连接至 CAN-Bus 网络，构成现场总线实验室、工业控制、智能小区、汽车电子网络等 CAN-Bus 网络领域中数据处理、数据采集的 CAN-Bus 网络控制节点。

USBCAN-Modul 8 高性能 CAN 接口卡是 CAN-Bus 产品开发、CAN-Bus 数据分析的强大工具，同时具有体积小巧、即插即用等特点，也是便携式系统用户的最佳选择。USBCAN-Modul 8 接口卡上自带 USB 接口，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-Modul 8 高性能 CAN 接口卡支持 WindowsXP/Win7/Win10 等 32 位/64 位操作系统，还可支持 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、Delphi、Labview 和 C++Builder 等开发例程示范，方便用户进行应用程序开发。

USBCAN-Modul 8 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8路CAN-Bus接口，使用DB9接线方式；
- 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DC+24V电源供电；
- 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、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
- 汽车电子应用
- 电力通讯网络
- 工业控制设备
- 高速、大数据量通讯

# 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)246mm * (宽)132mm * (高)45mm，其示意图如图 2.1 所示。




图 2.1 USBCAN-Modul 8 外形尺寸

## 2.2 接口定义及功能

USBCAN-Modul 8 接口卡集成 8 路 CAN 总线通道，使用 8 个 DB9 接口引出。其各接口位置及定义如图 2.2、图 2.3、表 2.1、表 2.2 所示。




图 2.2 USB 及电源接口位置

| 端口 | 名称 | 功能 |
|---|---|---|
| DC +24V | DC +24V | USBCAN 供电接口 |
| USB | USB | USBCAN 通讯接口，与电脑连接 |

表 2.1 USB 及电源接口定义




图 2.3 DB9 接口位置

| 引脚<br>（由左至右） | 端口<br>(各个 DB9) | 名称 | 功能 |
|---|---|---|---|
| 1 | CAN/DB9 | NC/+5V | +5V 可选 |
| 2 | | CAN_L | CAN_L 信号线（CAN 低） |
| 3 | | GND | CAN_GND 接地 |
| 4 | | NC | 无定义 |
| 5 | | NC | 无定义 |
| 6 | | GND | CAN_GND 接地 |
| 7 | | CAN_H | CAN_H 信号线（CAN 高） |
| 8 | | NC | 无定义 |
| 9 | | SHIELD | 屏蔽线（FG） |

表 2.2 USBCAN-Modul 8 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”（XP）或“程序和功能”（win7）中找到 ECAN Tools、CAN Test 和 CAN Pro 软件对其进行卸载。

客户也可以通过安装 ecantools 软件来安装驱动，软件安装完成后会自动安装驱动的。

驱动、软件正常安装后，将设备插入 PC 的 USB 接口，即可在 PC 设备管理器中发现新 USBCAN 设备，驱动名称:“GCAN USBCAN Device”，如无黄色叹号或问号即表示设备驱动正常，USBCAN 设备已与 PC 正常连接。安装完成后会显示 4 个驱动。

# 3.设备使用

## 3.1 与 PC 连接

USBCAN-Modul 8 接口卡使用 DC+24V 直流电源供电，USB 接口作为通讯接口，因此在使用时需将 DC+24V 和 USB 一起接入。

## 3.2 与 CAN-bus 连接

USBCAN-Modul 8 接入 CAN 总线连接方式为将 CAN_H 连 CAN_H,CAN_L 连 CAN_L 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 120Ω 的终端电阻；如果节点数目大于 2，中间节点不需要安装 120Ω 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。

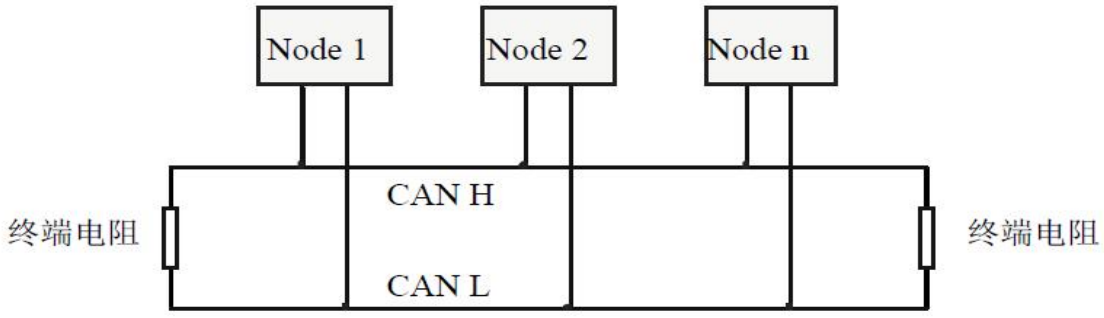



图 3.1 CAN-Bus 网络的拓扑结构

**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Φ1.0mm2，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 | 5.0km |
| 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 120Ω，则总线上的两个端点也应集成 120Ω终端电阻。

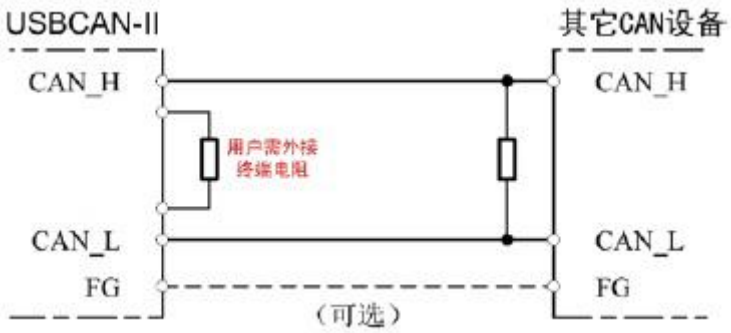



图 3.2 USBCAN-Modul 8 与其他 CAN 节点设备连接

**注意：USBCAN-Modul 8 内部已集成 120Ω终端电阻，可通过拨码开关选择是否将电阻接入总线，拨码开关在 DB9 接口侧，四条通道中间的拨码开关 R1~R8 分别控制对应的八条通道。**

## 3.4 系统状态指示灯

USBCAN-Modul 8 接口卡具有 1 个 PWR 指示灯、1 个 USB 指示灯，8 个 ERR 指示灯、8 个 RUN 指示灯来指示设备的运行状态。指示灯的具体指示功能见表 3.2，指示灯处于各种状态时的含义如表 3.3 所示。

| 指示灯 | 颜色 | 指示状态 |
|---|---|---|
| PWR | 绿 | 电源指示 |
| USB | 绿 | USB指示 |
| RUN | 绿 | CAN通信指示 |
| ERR | 红 | CAN通信错误指示灯 |

表 3.2 USBCAN-Modul 8 接口卡指示灯

USB 接口及电源接口连接，打开供电开关，供电正常后，PWR 指示灯会亮；当 CAN 端有数据传输时相应通道的 RUN 灯会亮；当 CAN 总线产生错误时 ERR 会亮。

| 指示灯 | 状态 | 指示状态 |
|---|---|---|
| PWR | 亮 | 24V供电正常 |
| | 不亮 | 24V供电故障 |

| | | |
|---|---|---|
| USB | 亮 | USB供电正常 |
| | 不亮 | USB供电故障 |
| RUN | 闪烁 | CAN总线有数据传输 |
| | 不亮 | CAN总线无数据传输 |
| ERROR | 亮 | CAN总线错误 |
| | 不亮 | CAN总线正常 |

表 3.3 USBCAN-Modul 8 接口卡指示灯状态

# 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。本章将对 ECANTools 主要功能做介绍。使用 Modul 8 产品打开所有通道需要依次打开 4 个 Ecantools 软件，Ecantools 软件每次启动只能打开 2 个 CAN 通道。

## 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。

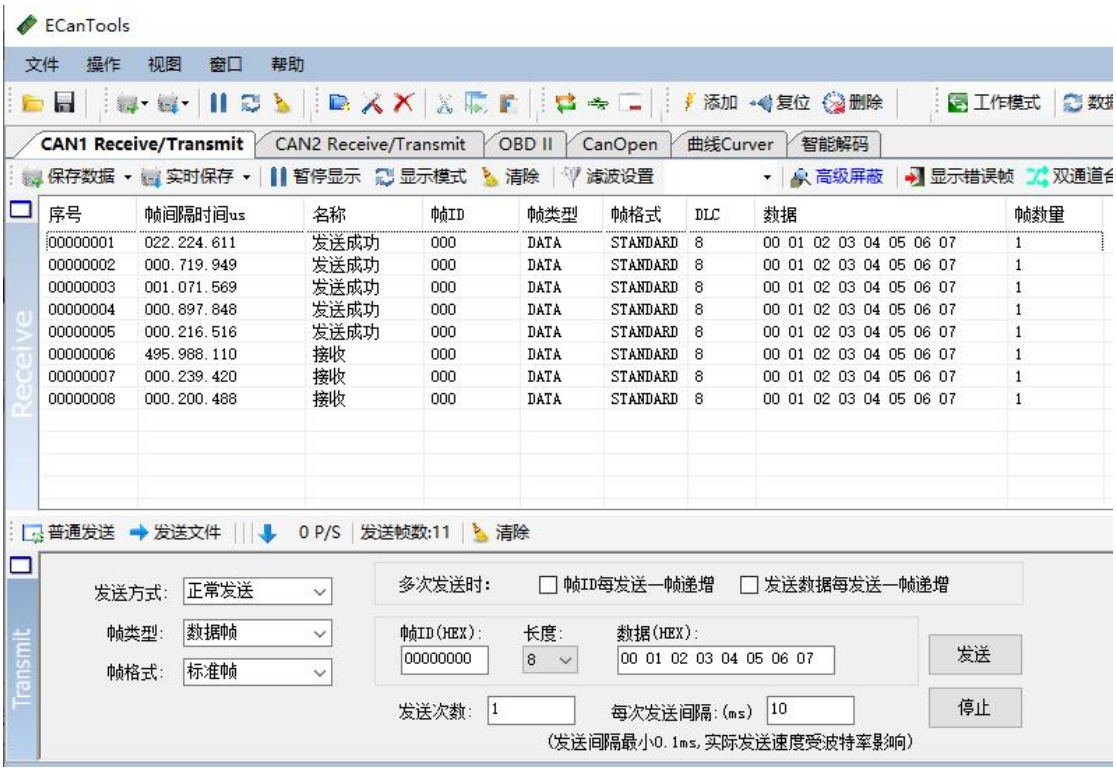
3. 选择工作模式。软件提供 3 种工作模式可选：正常模式、只听模式、自发自收模式。正常模式适用于软件数据收发；只听模式适用于在不干扰总线的情况下监听总线，但不可发送数据，此时 USBCAN 设备不作为 CAN 总线节点，不发送

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。

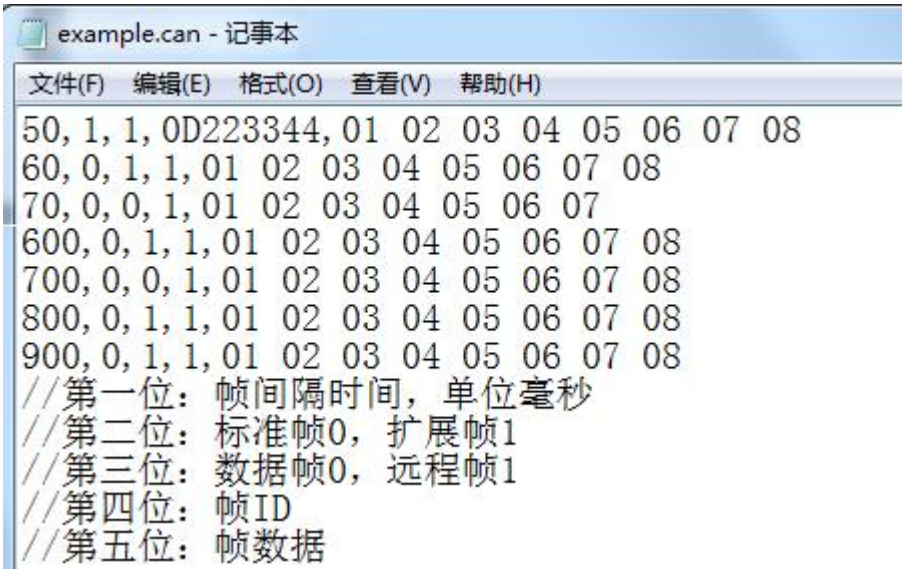
## 4.2 数据收发

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。



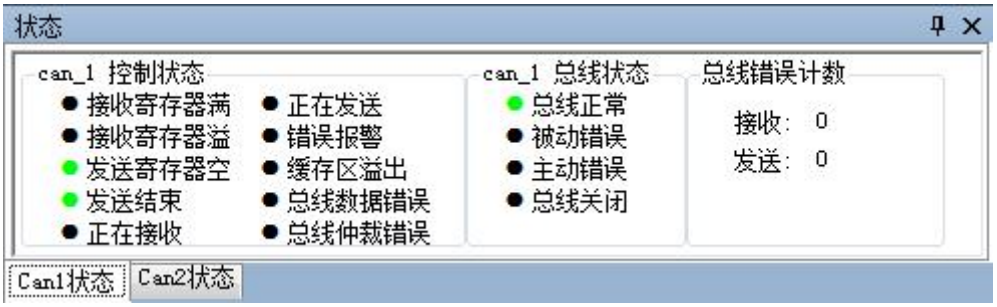

除以上基本功能外，软件还具有发送文件功能。文件发送分为普通文件发送及批处理文件发送。普通文件发送可以向总线上的节点发送文件，用户可以基于此功能开发 CAN 总线程序烧录软件；批处理文件发送可以将用户保存的批处理文件（.can）按保存时的时序及数据重新发送回 CAN 总线，批处理文件保存详见 4.5。

**批处理文件发送应用举例：**




## 4.3 总线分析功能

ECANTools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。

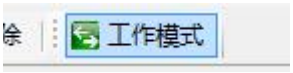
**CAN控制器消极错误：**总线发生多次发送错误或接收错误，导致CAN控制器进入错误消极状态，并能显示错误计数值。

**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。

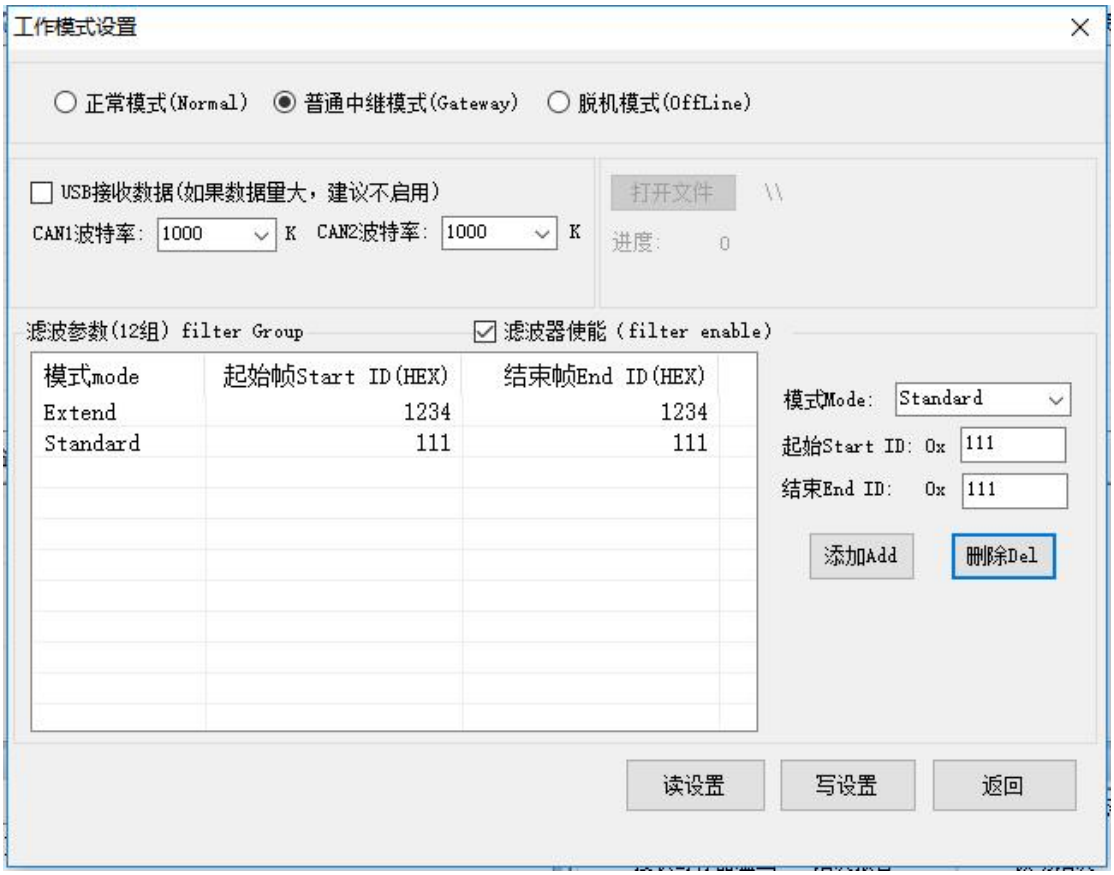
## 4.4 中继功能、脱机功能




在 CAN1、2 通道选项卡上方可以选择设备的工作模式，双通道设备均支持 CAN 中继工作模式。通过分别设置 CAN1、CAN2 的波特率，可以实现两条波特率不同的总线数据的转发中继。**点击“写设置”之后关闭软件，重新对设备上电便可实现脱机中继**。硬件设备提供外接 5VDC 标准工业现场电源插头，方便工业配套使用。

勾选“USB 接收数据”后，即可使用 ECANTools 软件监控两条通道中继的数据，也可显示中继数据流向，此功能可用于对 CAN 总线数据进行抓包，对抓取握手协议等非常适用。

中继功能可以设置滤波器使能，实现中继滤波功能。滤波是 CAN-ID 段设置，最多可以设置 12 条。滤波必须勾选滤波器使能，设置完成后设备重新上电才会生效。




**脱机模式应用举例：**

用户可以通过加载 TXT 文档的方式对设备进行脱机程序烧写，烧写脱机程序

后的设备可以实现智能发送数据等功能，且烧写后无需连接电脑，上电即运行。具体 TXT 文档编写规则及例子还可参见随货光盘中的“脱机模式使用说明”。

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| 格式说明 | 等待时间 | 模式 | 触发 ID | 替换掩码 | 发送 ID | 帧格式/帧类型 | 数据长度 | 帧数据 | 发送次数 | 间隔时间 |
| 举例 | 10, | 1, | 4000 0000, | 0000, | 144, | 0, | 8, | 01 02 03 04 05 06 07 08, | 100, | 500 |
| 备注 | 单位 ms | | | | | | | | 0 表示无限次 | 单位 ms |

注释：

②模式选择：1=CAN1 收发，2=CAN2 收发，3=CAN1 收到数据后 CAN2 转发，4=CAN2 收到数据后 CAN1 转发。

③触发 ID 设置：如果是 40000000 表示不触发。若设置为 11111111 则表示收到以此帧 ID 开头的数据才转发。

④替换掩码设置：此段为十六进制数据，需转换为二进制。如 0020 可表示为 0000 0000,0010 0000，其中逗号前边两位为帧 ID 和长度，逗号后边八位为帧数据，0 表示不换、1 表示换，如果设置成替换的话，**该数据将会变成帧数据相应位的数据**。

⑤发送 ID：如果设置了③触发 ID 的话，则该部分只有触发的情况下才会参与发送。如果没有设置③触发 ID 的话（即 40000000），这个发送 ID 为正常需要发送的帧 ID。

⑥帧格式/帧类型标识：0=标准数据帧，1=标准远程帧,2=扩展数据帧，3=扩展远程帧。

⑧帧数据：各个字节中间要有空格。

⑨发送次数：0 表示无限次发送。

例如：

**10,1,40000000,0000,144,0,8,01 02 03 04 05 06 07 08,100,500**

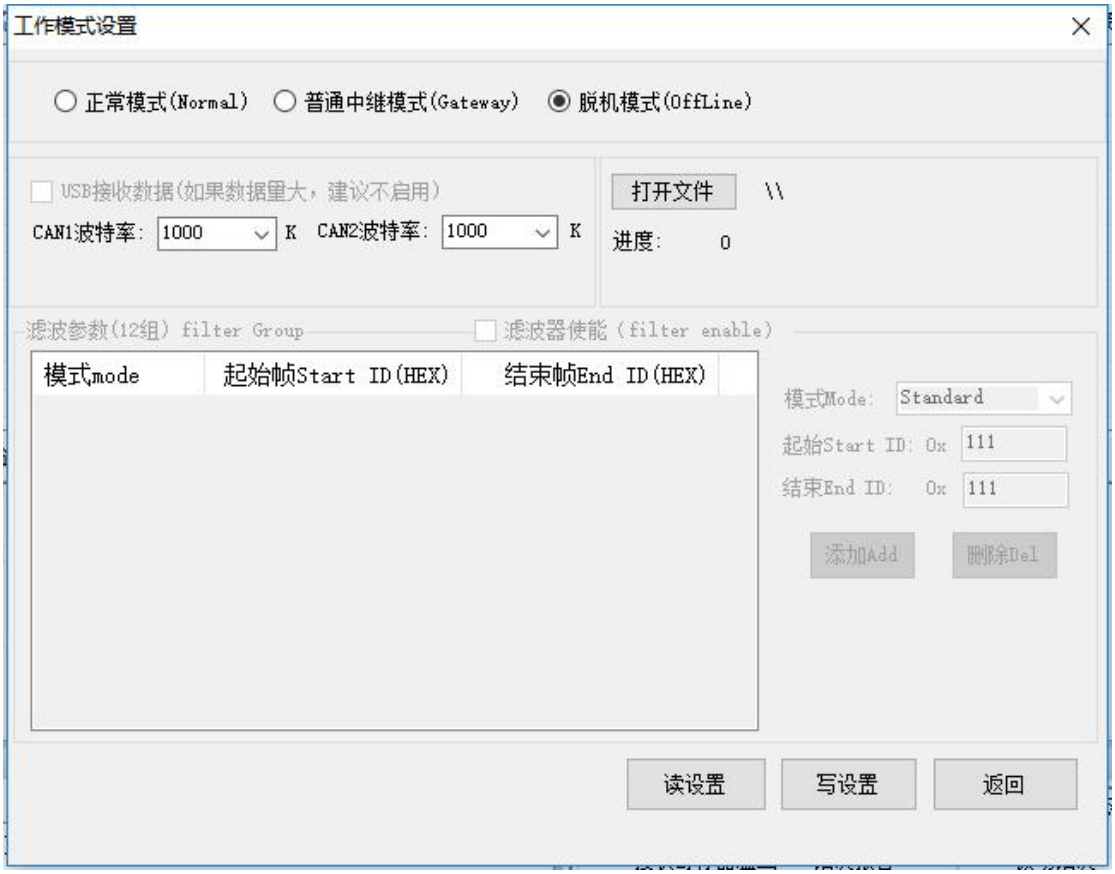
此条数据的功能是：上电等待 10ms，初始化 CAN1 通道收发数据，不设置触发，不替换 ID 和数据，发送 ID 144，标准帧，数据长度 8 位，发送数据 01 02.....08，发送次数 100 次，每次发送时间间隔 500ms。

**10,3,11111111,0020,123,0,8,01 02 03 04 05 06 07 08,1,500**

此条数据的功能是：上电等待 10ms，初始化为 CAN1 收到数据后往 CAN2 转发，触发 ID 11111111 即收到以此 ID 开头的数据才转发，替换掩码 0020 替换第三个字节的数据为 00（此段为十六进制数据，转换为二进制 0000 0000,0010 0000 其中逗号前边两位为帧 ID 和长度，逗号后边八位为帧数据，0 表示不换、1 表示换，设置成替换的将会变成**该数据将会变成帧数据相应位的数据，将第三位数据替换成 03）**，发送 ID 123 如没有设置 ID 替换则此位无意义，标准帧如第三位设置触发了则此位无意义触发数据与原数据帧格式相同，发送数据如已设置触发则此位无意义，发送次数 1 次，间隔 500ms。

**10,3,00000222,0000,123,0,8,01 02 03 04 05 06 07 08,1,500**

此条数据的功能是：上电等待 10ms，初始化为 CAN1 收到数据后往 CAN2 转发，触发 ID 00000222 即收到以此 ID 开头的数据才转发，替换掩码 0000 所有数据均不替换，发送 ID 123 如没有设置 ID 替换则此位无意义，标准帧如第三位设置触发了则此位无意义触发数据与原数据帧格式相同，发送数据没有设置替换则此位无意义，发送次数 1 次，间隔 500ms。此条数据可实现 CAN1 所有以 00000222 帧 ID 开头的数据都将原封不动的转发到 CAN2，类似单方向中继。




## 4.5 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、can、excel、二进制、ASC、BLF。

**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。

**双通道合并**：双通道合并接收数据，可保存在同一文件中，发送文件时根据保存文件的通道同时发送。

**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

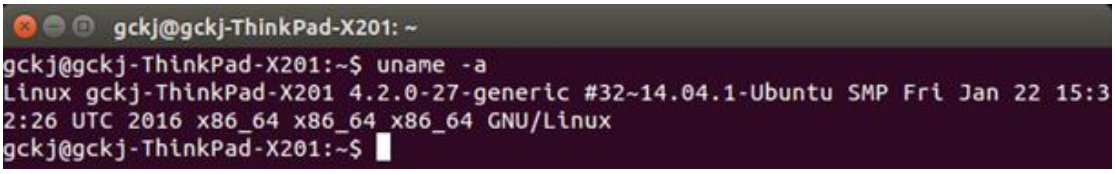
广成科技 USBCAN 分析仪系列产品支持各种版本 Linux 操作系统，我公司会向客户提供 32/64 位 Linux 系统驱动及二次开发相关文档，用户可自行在 Linux 系统中开发使用。

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）




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

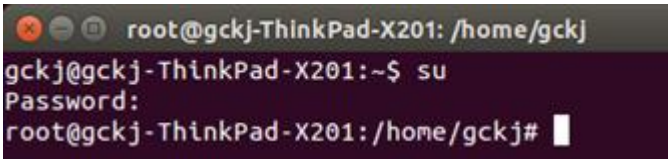
（此例子中我们把驱动文件放到系统桌面）




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）




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 l
ibECanVci.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；

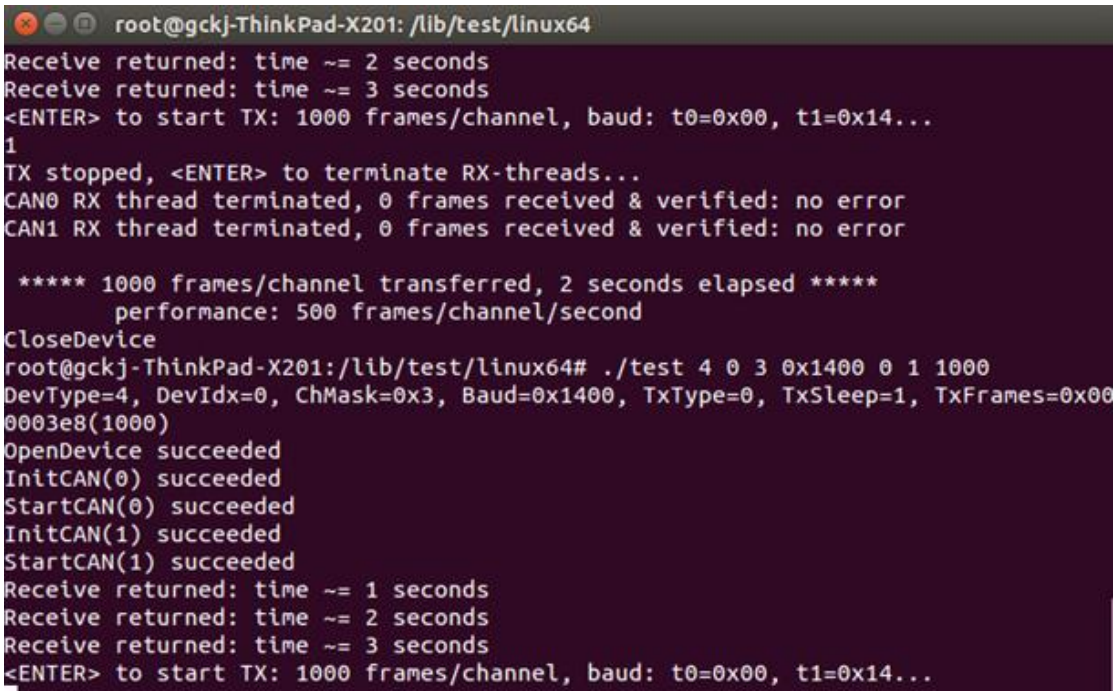
第四位（0x1400）：设置 CAN 总线波特率，0x1400 表示波特率为 1000K，其他波特率的值详见“EcanVCI 动态库使用手册”；

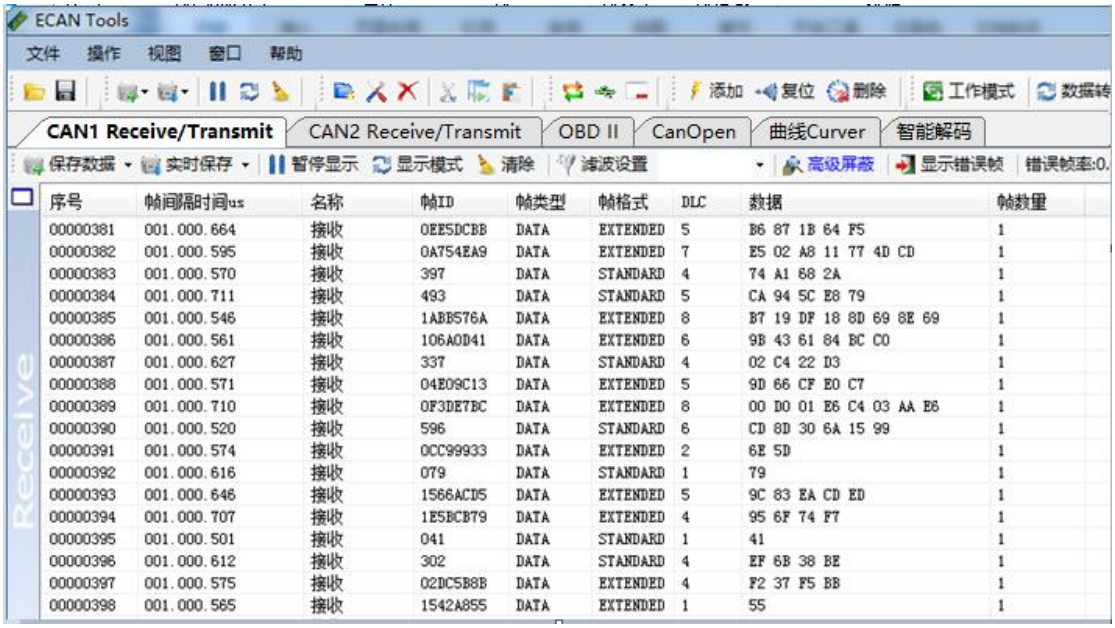
第五位（0）：工作模式，0 为正常模式，其他工作模式详见“EcanVCI 动态库使用手册”；

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






| 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 | 1ABB576A | 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 E6 | 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 | 001.000.585 | 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、Labview 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 6.1 为常用结构体名称及函数库调用流程。

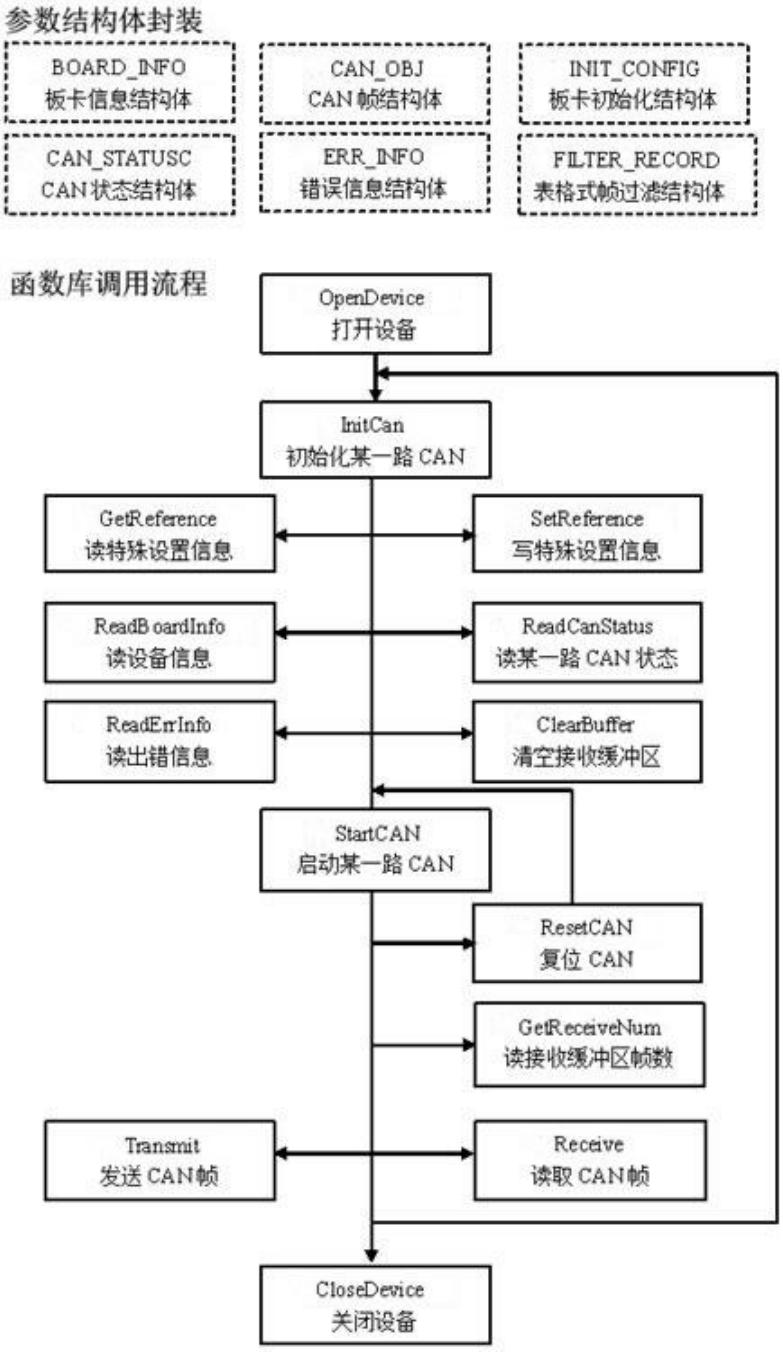



图6.1 二次开发函数调用

# 7. 技术规格

| | |
|---|---|
| **连接方式** | |
| PC端 | USB接口，A类型 |
| CAN端 | DB9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 | 已集成，通过拨码开关选择是否启用 |
| **供电电源** | |
| 供电电压 | +24V DC |
| 供电电流 | 最大500mA |
| **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<br>EN 55022:2011-12 |
| 防护等级 | IP 20 |
| **基本信息** | |
| 外形尺寸 | (长)246mm * (宽)132mm * (高)45mm |
| 重量 | 630g |

# 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 120Ω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 120Ω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 120Ω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-Modul 8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) 软件接收缓区冲溢出。 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-Modul 8 接口卡提供 2 个 CAN 端口，用户能够在同一程序中操作不同的端口。USBCAN-Modul 8 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%。

**12. 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。

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 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。

# 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
淘宝官方店：https://shop72369840.taobao.com/
天猫官方店：https://gcan.tmall.com/
京东官方店：https://mall.jd.com/index-684755.html
官方服务热线：13019325660
售前服务电话与微信号：15712411229
售前服务电话与微信号：18309815706
售前服务电话与微信号：18940207426
售后服务电话与微信号：18609820321
售后服务电话与微信号：13840170070