

USBCAN-Modul 16

16通道USB-CAN转换器

用户手册



文档版本： V1.60 （2018/07/18）

修订历史

版本	日期	原因
V1.00	2016/07/16	创建文档
V1.50	2017/01/13	修改文档参数
V1.60	2018/07/18	调整文档结构

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.....	4
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.....	7
3. 设备使用.....	8
3.1 与 PC 连接.....	8
3.2 与 CAN-bus 连接	8
3.3 CAN 总线终端电阻	9
3.4 系统状态指示灯	9
4. ECANTools 软件使用.....	11
4.1 软件启动.....	11
4.2 数据收发.....	12
4.3 总线分析功能.....	12
4.4 中继功能、脱机功能.....	13
4.5 其他功能.....	14
5. Linux 系统使用说明	16
6. 二次开发.....	19
7. 技术规格.....	20
8. 常见问题.....	21
附录 CAN2.0B 协议帧格式	24
销售与服务.....	26

1. 功能简介

1.1 功能概述

USBCAN-Modul 16 是集成 16 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 16 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 16 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、即插即用等特点，也是便携式系统用户的最佳选择。USBCAN-Modul 16 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 16 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、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 16 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 16 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；
- 支持各版本 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)482mm * (宽)119mm * (高)44mm，其示意图如图 2.1 所示。



图 2.1 USBCAN-Modul 16 外形尺寸

2.2 接口定义及功能

USBCAN-Modul 16接口卡集成16路CAN通道，使用16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

引脚 (由左至右)	端口	名称	功能
1	DC +24V	DC +24V	USBCAN 供电接口
2	USB	USB	USBCAN 通讯接口，与电脑连接
3	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 CAN 总线接口位置

引脚 (由左至右)	端口 (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 16 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”中找到 ECANTools 软件对其进行卸载。

3. 设备使用

3.1 与 PC 连接

USBCAN-Modul 16 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 16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。

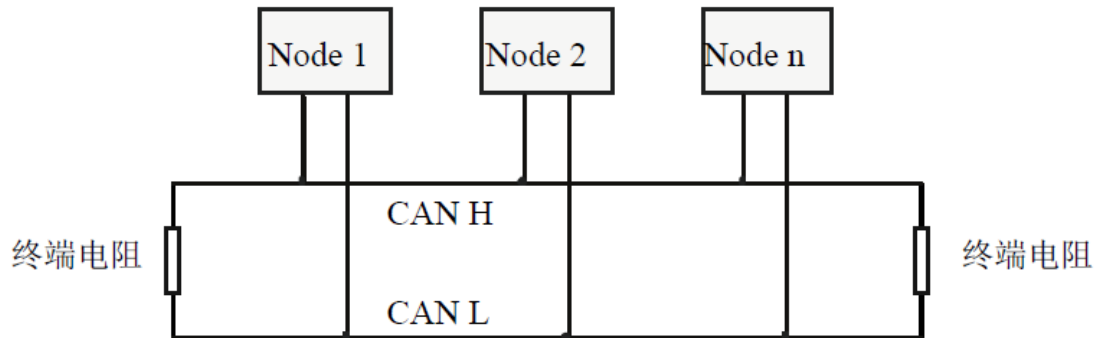


图 3.1 CAN-Bus 网络的拓扑结构

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 $\Phi 1.0\text{mm}^2$ ，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	40m
500 kbit/s	110m
250 kbit/s	240m
125 kbit/s	500m
50 kbit/s	1.3km
20 kbit/s	3.3km
10 kbit/s	6.6km
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。USBCAN-Modul 16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。

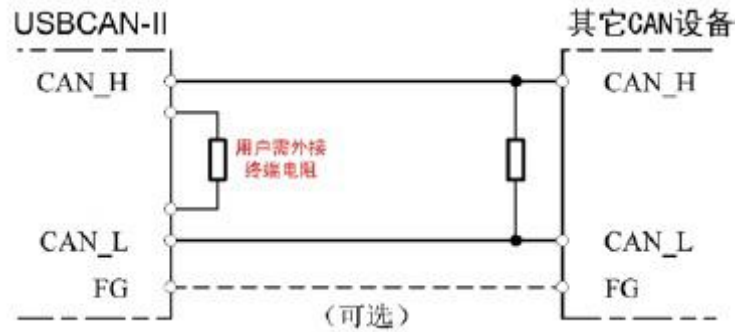


图3.2 USBCAN-Modul 16与其他CAN节点设备连接

注意：USBCAN-Modul 16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~R4分别控制对应的四条通道。

3.4 系统状态指示灯

USBCAN-Modul 16接口卡具有4个PWR指示灯、16个ERR指示灯、16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	红+绿	电源指示
RUN	绿	系统指示
ERR	红	错误指示灯

表3.2 USBCAN-Modul 16接口卡指示灯

USB接口及电源接口连接且供电正常后，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	红灯亮	USB供电正常
	绿灯亮	系统供电正常
	不亮	对应的供电故障
RUN	闪烁	CAN数据传输
	不亮	无数据

ERROR	亮	错误
	不亮	正常

表3.3 USBCAN-Modul 16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。

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。



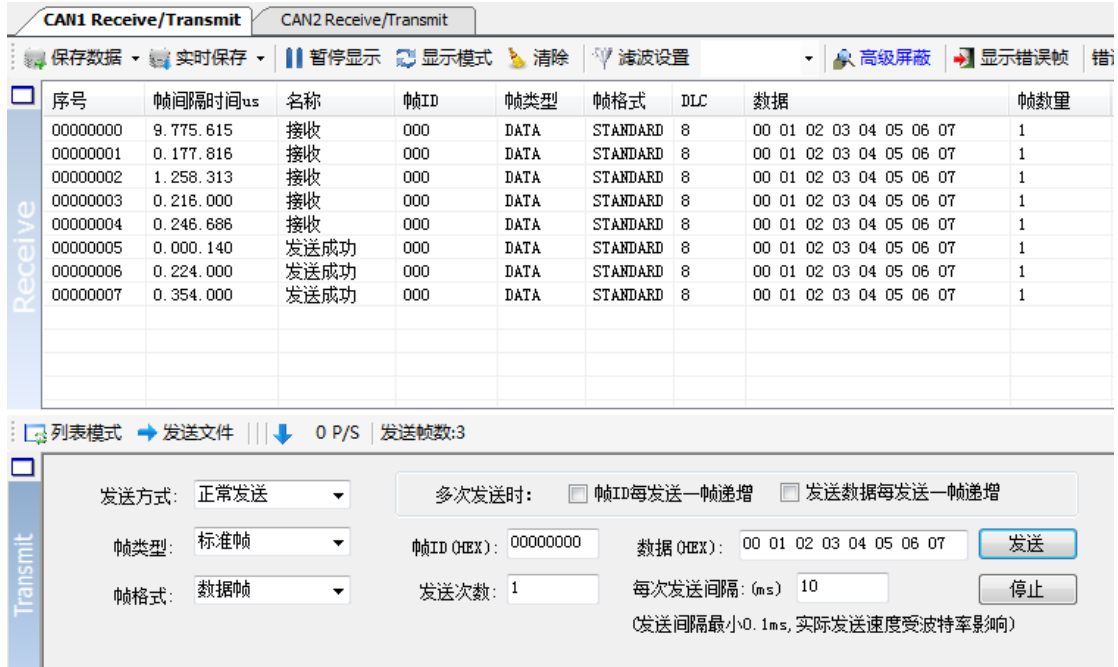
2.选择对应的设备类型后，点击“打开设备”即可在设备列表中出现已经插入电脑USB接口的CAN设备。

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：**

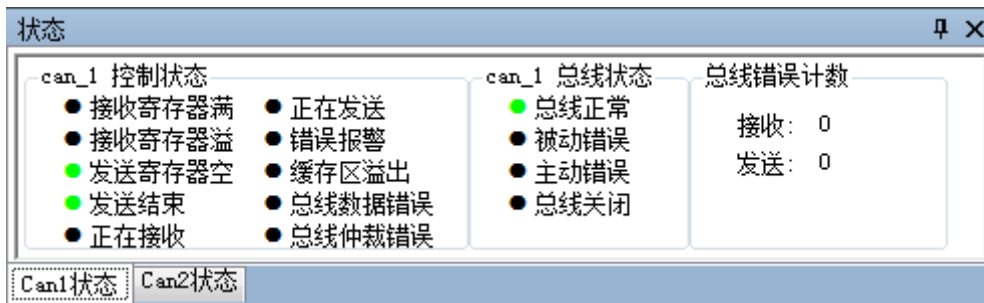
```

example.can - 记事本
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)
50, 1, 1, 0D223344, 01 02 03 04 05 06 07 08
60, 0, 1, 1, 01 02 03 04 05 06 07 08
70, 0, 0, 1, 01 02 03 04 05 06 07
600, 0, 1, 1, 01 02 03 04 05 06 07 08
700, 0, 0, 1, 01 02 03 04 05 06 07 08
800, 0, 1, 1, 01 02 03 04 05 06 07 08
900, 0, 1, 1, 01 02 03 04 05 06 07 08
//第一位：帧间隔时间，单位毫秒
//第二位：标准帧0，扩展帧1
//第三位：数据帧0，远程帧1
//第四位：帧ID
//第五位：帧数据
    
```

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。

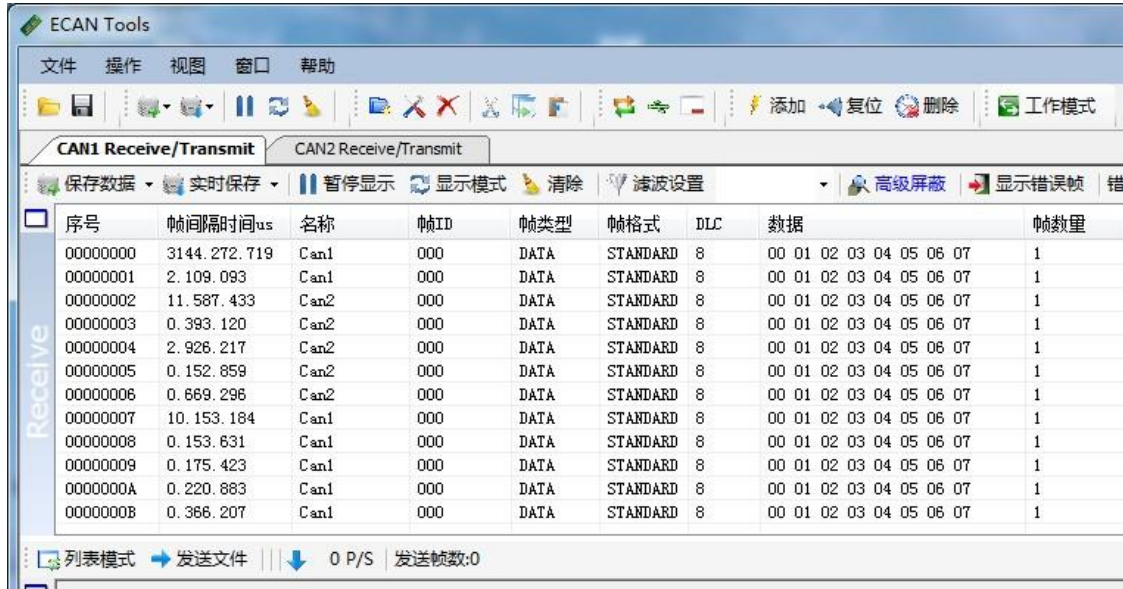
CAN控制器仲裁丢失：本节点与其他设备在竞争总线时失败，待发报文将在下一次总线空闲时尝试发送，能捕捉到相应的错误位信息并显示出来。

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

CAN中继功能可将2条波特率相同或不同的CAN总线数据相互转发，用法类似于CAN中继器。还可设置中继监控模式，通过电脑可以看到数据流向，比如CAN1收CAN2发，此功能可以用来测试两个节点的数据流向，清晰的看出哪些数据是哪个通道接收到的，便于进行通信协议破解等用途。



配置脱机模式时，用户可以用记事本 (.txt) 编辑指令，下载到CAN设备中，下载好的CAN设备通电后就可以根据记事本中的指令进行脱机工作，可进行脱机的指令有：发送、延时、触发发送、接收修改后发送等常用功能，用户即使不会二次开发软件，也可轻松实现脱机自动控制。上电就可以自动发送数据，此功能可用于生产线测试，破解协议后模拟协议数据等，用途非常广泛。

脱机模式应用举例：

10 (开始等待时间),1(模式 1=CAN1收发 2=CAN2收发 3=CAN1收CAN2发 4=CAN2收CAN1发),40000000 (触发ID, 如果是40000000表示不触发),0000 (替换掩码0000 00xx,xxxx xxxx 每位代表一个替换 1变换,0不换。按ID,长度,数据顺序,二进制),0D223344 (发送ID),0 (0=标准帧,1=标准远程帧,2=扩展帧,3=扩展远程帧),8 (长度),01 02 03 04 05 06 07 08,100 (发送次数,0表示无限次),500 (间隔时间ms)

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。

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。

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”

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...
    
```

The screenshot shows the ECAN Tools software interface. The main window displays a table of received CAN frames. The table has columns for sequence number, timestamp, name, ID, type, format, DLC, data, and count. The data shows a series of received frames with various IDs and data lengths.

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	1E5CB79	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	02DC588B	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：ECANVCL.h、ECANVCL.lib、ECANVCL.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。

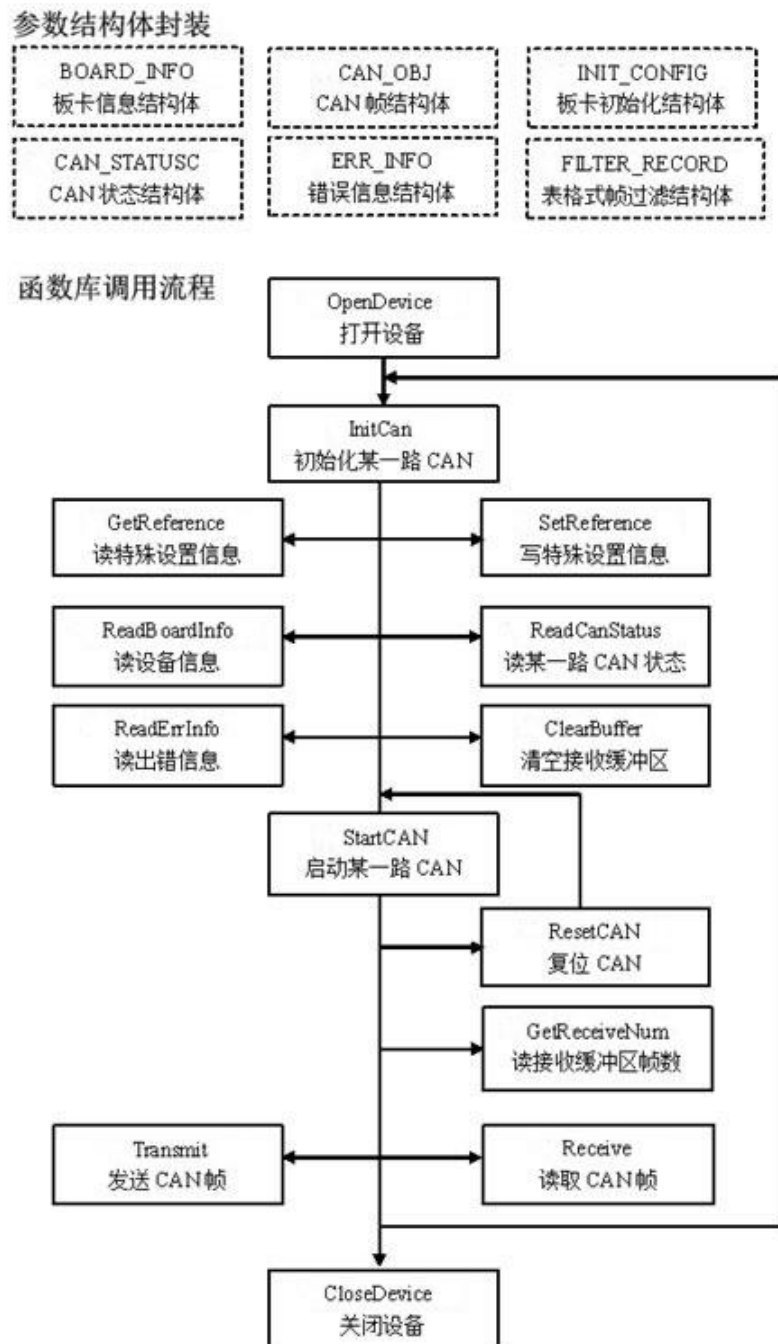


图6.1 二次开发函数调用

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800mA
环境试验	
工作温度	-40°C~+85°C
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	(长)482mm * (宽)119mm * (高)44mm
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 16 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-Modul 16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 16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。

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：辽宁省沈阳市皇姑区崇山中路 42 号工业设计中心

邮编：110000

电话：024-31230060

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：18309815706

售后服务电话与微信号：13840170070



全国服务电话：400-6655-220