

MiniPCleCAN-II

MiniPCle 接口 CAN 卡

UM01010101 V1.05 Date: 2020/11/04

产品用户手册

类别	内容
关键词	MiniPCle、CAN 报文监测
摘要	MiniPCle 接口 CAN 卡

修订历史

版本	日期	原因
V0.00	2015/07/01	创建文档
V1.01	2015/09/30	增加 Linux 驱动安装方法、快速使用说明和动态库使用方法
V1.02	2017/08/09	更改公司名称、销售与服务网络
V1.03	2018/09/03	修改尺寸图
V1.04	2019/03/12	更新文档页眉页脚、“销售与服务网络”内容和新增“免责声明”内容
V1.05	2020/11/04	增加强调 MiniPCle 插槽描述

目 录

1. 功能简介.....	1
1.1 功能特点.....	2
1.2 典型应用.....	2
2. 设备安装.....	3
2.1 供电.....	3
2.2 信号指示灯.....	3
2.3 MiniPCle 接口定义.....	4
2.4 CAN 第二功能引脚切换.....	5
3. 驱动安装.....	6
3.1 在 Windows 系统下第一次安装驱动程序.....	6
3.2 检查设备否安装成功.....	7
3.2.1 打开 WINDOWS 设备管理器.....	7
3.2.2 确认新的设备是否已经成功安装.....	8
3.3 在 Linux 下驱动安装.....	8
4. 快速使用指南.....	9
4.1 CANTest 基本操作.....	9
4.1.1 设备类型选择.....	9
4.1.2 滤波设置.....	11
4.1.3 启动 CAN.....	12
4.1.4 获取设备信息.....	12
4.2 发送接收实验.....	12
4.2.1 搭建测试环境.....	12
4.2.2 打开设备.....	14
4.2.3 发送数据.....	15
4.2.4 实时保存与停止保存.....	17
4.2.5 DBC 解码与按 ID 分类显示.....	18
4.2.6 总线利用率.....	18
4.2.7 错误信息显示.....	19
5. 接口库函数使用方法.....	20
5.1 在 windows 下调用动态库的方法.....	20
5.1.1 VC 调用动态库的方法.....	20
5.1.2 VB 调用动态库的方法.....	20
5.2 接口库函数使用流程.....	22
6. 电气特性.....	23
7. 结构尺寸.....	24
8. 检查和维护.....	26
9. 免责声明.....	28
附录 A SJA1000 标准波特率.....	29

1. 功能简介

MiniPCleCAN-II 是致远电子推出的高性能 MiniPCle 接口 CAN 卡，可以把 CAN 网络连接在带有 MiniPCle 卡槽的电脑上，MiniPCleCAN-II 采用标准 MiniPCle 板卡尺寸，可以非常方便地安装在带有 MiniPCle 接口的在笔记本电脑或工控电脑端，使之成为强大的 CAN 分析仪。

MiniPCleCAN-II 集成 2 路 CAN 接口，同时为了方便扩展，CAN 信号第二功能引脚可通过电阻 R24 选焊，切换到 MiniPCle 接口预留 IO 口上，方便用户可自行在底板上设计 CAN 收发电路。

采用 MiniPCleCAN-II 高性能 CAN 接口卡，PC 可以通过 USB 总线连接至 CAN-bus 网络，构成现场总线实验室、工业控制、高性能小区、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。

MiniPCleCAN-II 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。

MiniPCleCAN-II 高性能 CAN 接口卡支持 Win2000/XP/7/8/10 等操作系统，也支持 Linux 的操作系统。MiniPCleCAN-II 提供了统一的应用程序编程接口和完整的应用示范代码，含 VC、VB、Delphi 和 C++ 等开发例程示范，方便用户进行应用程序开发。

MiniPCleCAN-II 接口卡支持 CANTest 通用测试软件，可执行 CAN-bus 报文的收发和监测等功能。产品如图 1.1 所示。

MiniPCEeCAN 卡使用的是 MiniPCle 插槽里面的 USB D+和 USB D-信号线，请在购买和使用产品前，确认主板 MiniPCle 插槽是否有提供 USB D+和 USB D-信号引脚，引脚序号详见表 2.3，如果主板插槽缺失 USB 信号引脚，则无法使用此 CAN 卡。

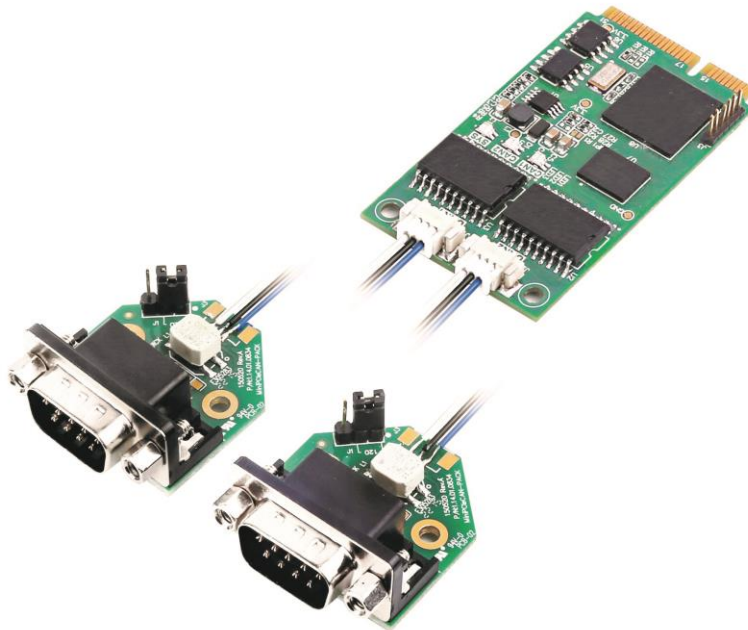


图 1.1 产品外观

功能特点

- 采用标准 MiniPCle 接口；
- 支持 CAN2.0A 和 CAN2.0B 协议，符合 ISO/DIS11898 规范；
- 集成 2 路 CAN-bus 接口；
- CAN TTL 信号可切换为第二功能 IO，方便自行设计 CAN 收发电路；
- CAN-bus 通讯波特率在 5Kbps~1Mbps 之间任意可编程；
- 采用 MiniPCle 接口供电；
- CAN-bus 接口采用电气隔离，隔离模块绝缘电压：DC 2500V；
- 支持 Win2000、WinXP、Win7、Win8、Win10 操作系统及 Linux 操作系统；
- 支持 CANtest 测试软件；
- 体积小巧，即插即用；
- 工作温度-40~85℃；
- 标准 MiniPCle 板卡尺寸 30mm（宽）* 50.95mm（长）。

1.1 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

2.1 供电

MiniPCleCAN-II 采用 MiniPCle 接口 3.3V 供电，指示灯 SYS 点亮，先显示红色，表示设备有电源供给，稍后将闪烁数次，并稳定显示绿色，表示与 PC 实现通讯连接。

2.2 信号指示灯

MiniPCleCAN-II 接口卡具有 1 个双色 SYS 指示灯、1 个双色 CAN1 指示灯、1 个双色 CAN2 指示灯来指示设备的运行状态。这 3 个指示灯的具体指示功能见表 2.1，当这 3 个指示灯处于各种状态下时，CAN 总线的状态如表 2.2 所示。

表 2.1 指示灯指示功能

指示灯	状态	指示状态
SYS	红色	设备初始化状态指示
	绿色	MiniPCle 接口信号指示
CAN1	绿色	CAN 接口运行正确
	红色	CAN 接口出现错误
CAN2	绿色	CAN 接口运行正确
	红色	CAN 接口出现错误

MiniPCleCAN-II 接口卡上电后，系统初始化状态指示灯 SYS 点亮为红色，表明设备已经供电，系统正在初始化；如果系统初始化状态指示灯 SYS 不亮，表示存在系统电源故障或系统发生有严重的错误。

与 PC 连接正常后（已安装驱动），指示灯 SYS 亮为绿色。当 USB 接口有数据在传输时，USB 信号指示灯 SYS 会闪烁绿色。

CAN1、CAN2 指示灯点亮为绿色时表示 CAN 控制器已完成初始化，进入正常工作状态。

当 CAN 控制器出现错误时，CAN1、CAN2 指示灯将点亮为红色；当清除 CAN 控制器的错误后，CAN1、CAN2 指示灯将点亮为绿色。

表 2.2 CAN 总线状态

CAN 指示灯状态	CAN 总线状态
CAN1、CAN2 全灭	CAN 控制器与总线断开
CAN1、CAN2 红绿交替闪烁	CAN 控制器未启动，提示用户启动 CAN 控制器
CAN1、CAN2 绿色指示灯常亮	CAN 总线运行正常
CAN1、CAN2 红色指示灯闪烁	CAN-bus 总线有错误或数据溢出，有可能丢失帧

2.3 MiniPCle 接口定义



图 2.1 MiniPCle 引脚顺序示意图

MiniPCle 接口定义如表 2.3 所示.第 17,19,37,39 四个引脚为 CAN 的第二功能引脚;36,37 为 USB 引脚。

表 2.3 MiniPCle 接口定义

信号名称	引脚编号	说明
nWAKE	1	nWAKE 信号/默认上拉
nRESET	22	nRESET 信号/未使用
TD0_REV	17	CAN0 TXD 第二功能 IO
RD0_REV	19	CAN0 RXD 第二功能 IO
TD1_REV	37	CAN1 TXD 第二功能 IO
RD1_REV	39	CAN1 RXD 第二功能 IO
USB_D-	36	USB_D-
USB_D+	38	USB_D+
3.3V	2、24、52	电源
GND	9、15、18、21、26、27、29、34、 35、40、50	地

2.4 CAN 第二功能引脚切换

当需要自行设计 CAN 收发电路或者提高 CAN 信号线防护等级时，可以考虑将 CAN 信号第二功能引脚通过 MiniPCle 保留引脚接入底板，自行设计 CAN 收发电路，提高了系统的灵活性。

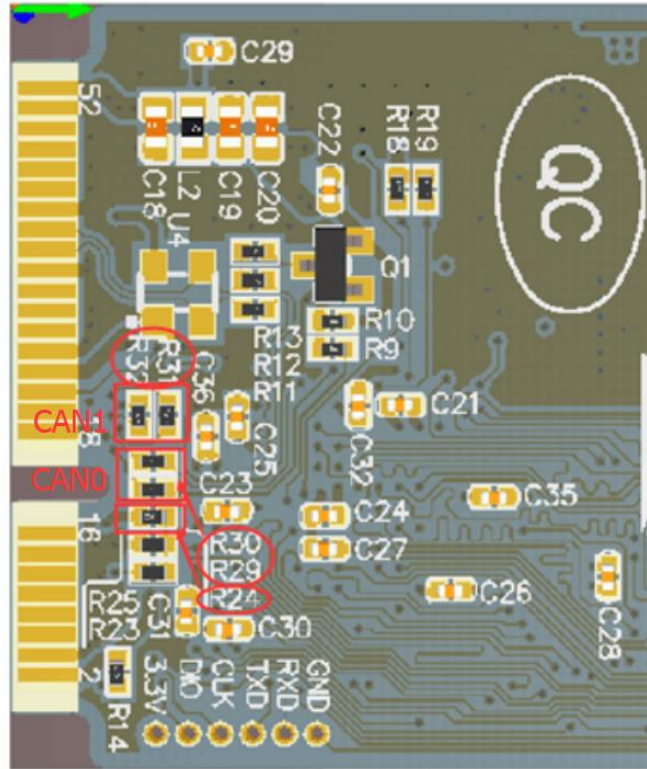


图 2.2 CAN 信号第二功能选择电阻

CAN 信号第二功能引脚切换通过 R24 切换（在板卡背面可找到该电阻，如图 2.2 所示）。R24 默认焊接，系统采用默认的板载 CAN 收发器 ADM3053，当 R24 去掉时，系统上电时自动启用 CAN 信号第二功能引脚，引脚顺序参考[错误!未找到引用源。](#)。这时 R29、R30 需要分别焊接 0 欧姆电阻使能 CAN0，R31、R32 分别焊接 0 欧姆电阻使能 CAN1。

3. 驱动安装

3.1 在 Windows 系统下第一次安装驱动程序

- A. “\USBCAN\Driver” 目录下，找到 usbcan.inf 文件，将它拷贝到系统的 windows\inf 目录下，找到 usbcan.sys 将它拷贝到 windows\system32\driver 下。

注：驱动下载地址 http://www.zlg.cn/canbus/product_detail.php?id=4。

- B. 文件复制完成后，此时将 MiniPCleCAN-II 智能型 MiniPCle 接口 CAN 卡与 PC 机正确连接；Window 将检测到新硬件，自动启动“发现新硬件”向导程序，点击“下一步”继续。如图 3.1 所示。



图 3.1 新硬件驱动安装向导

- C. 向导开始搜索新硬件驱动。如图 3.2 所示。



图 3.2 搜索驱动

- D. 稍候片刻，如果是在 Windows XP / Windows2000 操作系统下可能会出现与操作系统兼容性问题的警告，不理睬它，直接点击“仍然继续”按钮。如图 3.3 所示。

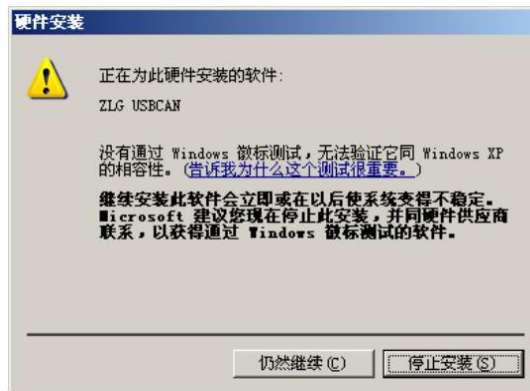


图 3.3 驱动安装

- E. 继续安装后，会出现找到新硬件，并安装完成。如图 3.4 所示。



图 3.4 驱动安装完成

- F. 点击“完成”后，此时 MiniPCleCAN-II 智能型 MiniPCle 接口 CAN 卡初始化指示灯 SYS 指示灯红色灭，SYS 指示灯亮绿色，表明硬件驱动安装成功并可以应用了。

3.2 检查设备否安装成功

3.2.1 打开 WINDOWS 设备管理器

- 鼠标右击桌面上我的电脑图标；
- 从下拉菜单中选取“属性”选项；
- 选择“硬件”标签；
- 鼠标单击“设备管理器”按钮打开当前硬件设备列表。

3.2.2 确认新的设备是否已经成功安装

检查“通用串行总线设备”设备类中，“USBCAN”设备是否已经在当前硬件列表中。成功安装后在“设备管理器”界面中可以看到“通用串行总线设备”设备类下的“USBCAN”设备。下图所示为计算机上“ZLG USBCAN 系列智能 CAN 接口卡”设备正常安装的情况。如图 3.5 所示。



图 3.5 安装完成

当 USBCAN mini 智能 CAN 接口卡与 PC 机进行数据传输时,接口卡上的 SYS 指示灯 会 闪烁绿色。

3.3 在 Linux 下驱动安装

A. usbcan 驱动基于 libusb 实现, 请先通过以下命令安装依赖库:

```
#apt-get install libusb-1.0-0
```

B. 把"usbcan.so、libusbcan.so.1"拷到"/lib"目录下, 在 test 目录运行 make 即可编译。

注: usbcan.ko、libusbcan.so.1 文件请向研发索要。

C. 在 test 目录到"./test", 查看参数调用示例并参考进行测试。。

D. test 会对每个通道以自发自收方式进行测试, 如果板卡正常, 会在结束时显示收发帧数和发送速度。

注: 现在采用的 Linux 自带的 USB 驱动。以前旧的发布方式, 不开源, 所以每次客户更换内核需要向研发部申请定制驱动。

4. 快速使用指南

4.1 CANTest 基本操作

CANtest 测试软件可以在配套光盘中找到（需要安装）。如图 4.1 所示。

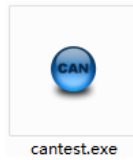


图 4.1 CANTest 软件图标

注：CANTest 软件下载地址 http://www.zlg.cn/canbus/product_detail.php?id=4。

4.1.1 设备类型选择

在进行操作之前,首先得从“选择设备”菜单中选择 USBCAN2, 如图 4.2 所示。

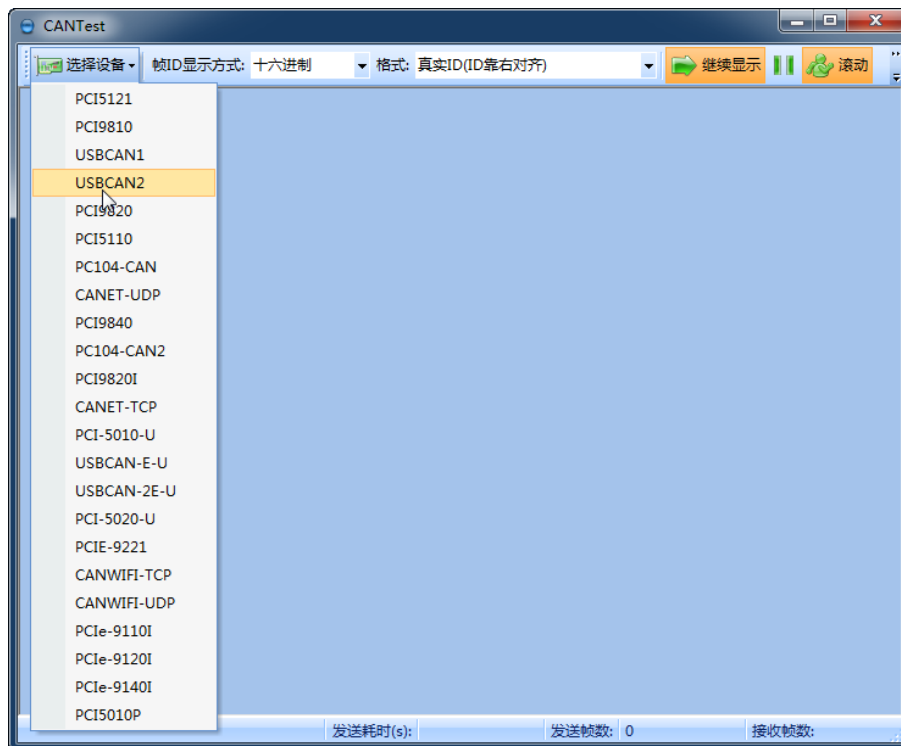


图 4.2 选择设备

选择确定后会弹出“打开设备”对话框，如图 4.3 所示。



图 4.3 打开设备对话框

在这个对话框中您可以选择您要打开的设备索引号和 CAN 通道，以及设置 CAN 的初始化参数，然后点“确定”按钮来打开设备操作窗口（或者也可以点击“确定并启动 CAN”按钮打开设备操作窗口并自动打开设备和启动 CAN 通道）。

4.1.2 滤波设置

接着，设备操作窗口中可以点击“滤波设置”按钮进行滤波设置（如果不需要设置滤波，可以略过此步骤）如图 4.4 所示。

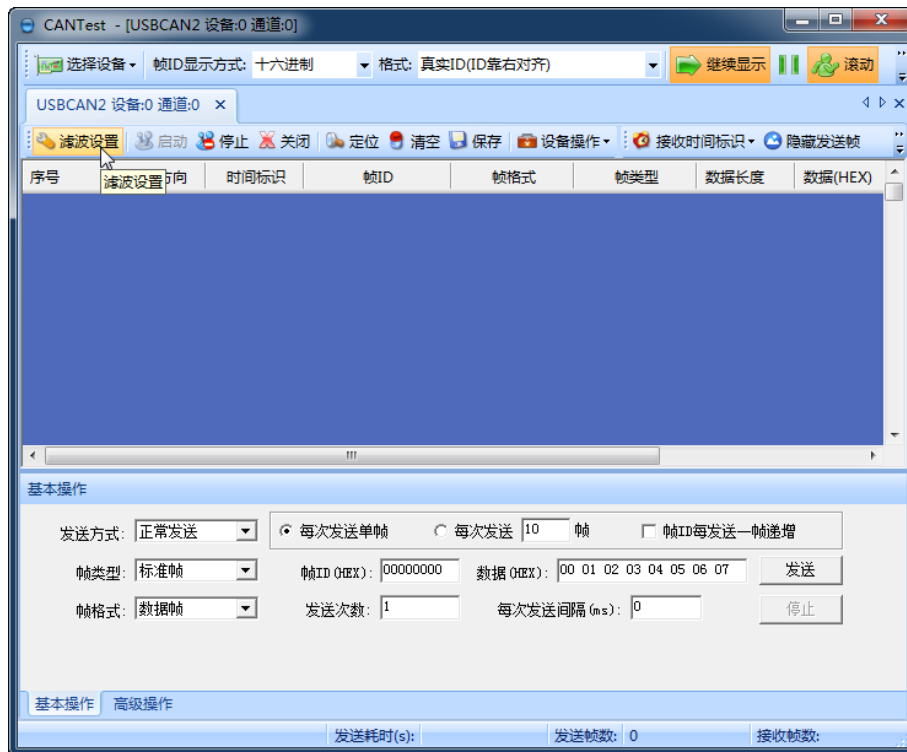


图 4.4 滤波设置 1

此时会弹出“滤波设置”对话框，如图 4.5 所示。

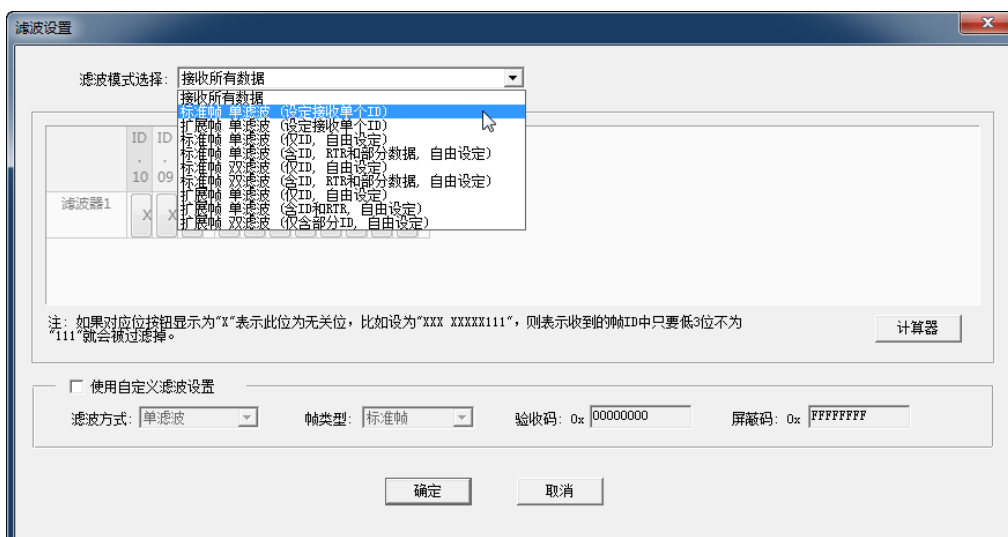


图 4.5 滤波设置 2

在其中先选择滤波模式，然后通过设定滤波器来设置需要过滤的 CAN 帧。

4.1.3 启动 CAN

点击“启动”按钮启动 CAN 通道，此时接收到的 CAN 数据将会自动在数据列表中显示如图 4.6 所示。



图 4.6 启动

4.1.4 获取设备信息

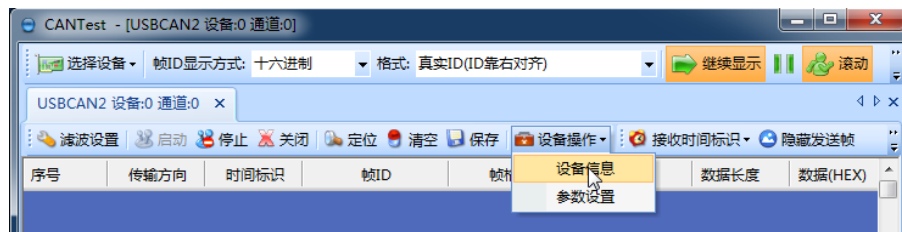


图 4.7 设备信息

在启动 CAN 通道后，您可以选择“设备操作”菜单中的“设备信息”选项来获得当前设备的详细信息，如图 4.7 所示。

4.2 发送接收实验

本节讲解 MiniPCleCAN-II 的简单发送接收测试，DBC 解码，与总线利用率的演示。

4.2.1 搭建测试环境

确保连线正确，接口定义如图 4.8 所示，连接效果图如图 4.9 所示。

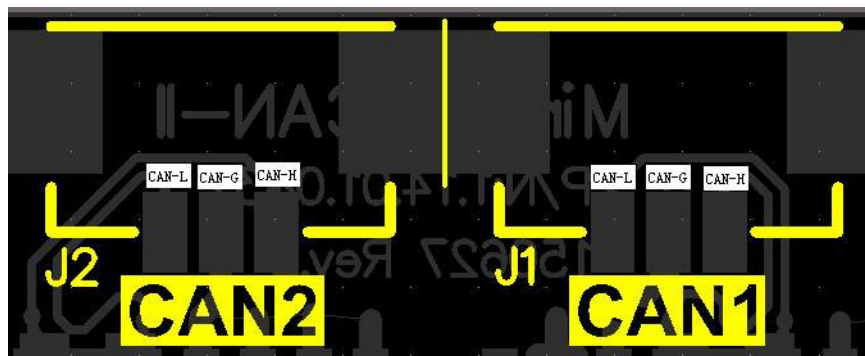


图 4.8 接口定义

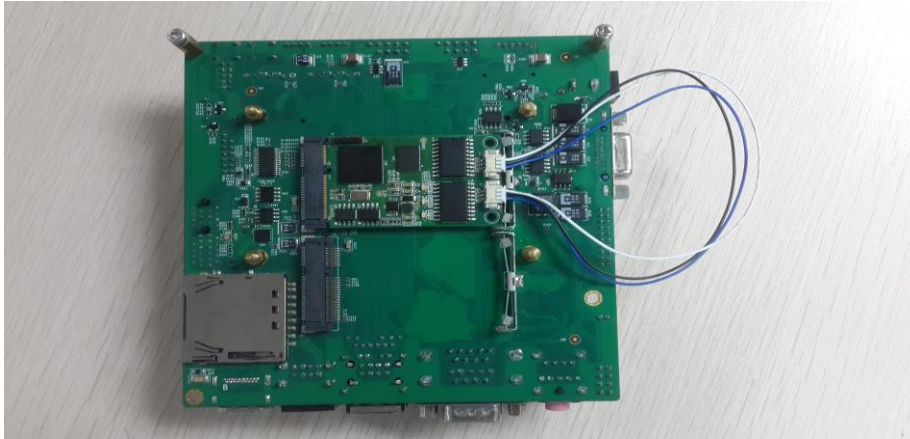


图 4.9 连线示意图 1

我是基于我们公司的核心板，和底板搭建的，其实只要接入 MiniPCle 接口的设备，安装好驱动就可以了。整体如图 4.10、图 4.11 所示。



图 4.10 连线示意图 2



图 4.11 连线示意图 3

4.2.2 打开设备

首先打开 CANTest 软件，选择好设备类型，参照 4.1.1 小结。然后按图 4.12、图 4.13 所示配置好设备，注意 CAN 的路数选择。设备索引号为识别设备的代码，同一设备的不同 CAN 接口要选择相同的设备索引号，不同设备选择不同的设备索引号，一般来讲设备索引号从 0 开始。第几路 CAN 是在同一设备索引号下用来区分不同的 CAN 路数，本设备有两路 CAN，所以在第几路 CAN 选项有 0, 1 两个选项。波特率选择 100K，由于本次为测试没有安装终端电阻，所以波特率选择不要超过 100K，真正使用的话一定要加相应阻值的终端电阻。



图 4.12 第 1 路 CAN

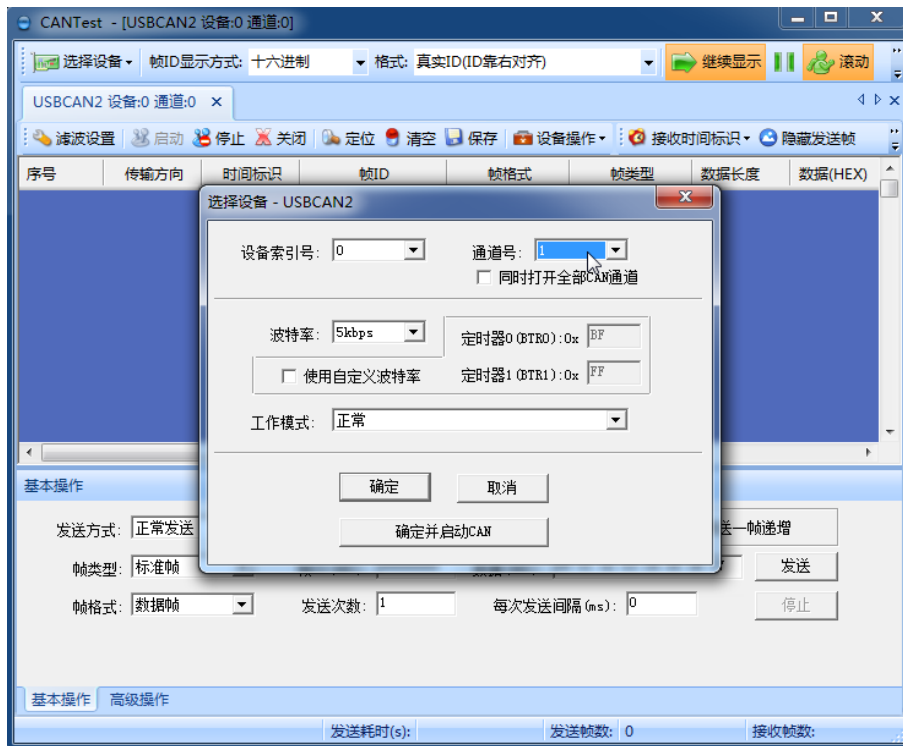


图 4.13 第 2 路 CAN

4.2.3 发送数据

当您启动 CAN 成功后，在如图 4.14 中设置好您要发送的 CAN 帧的各项参数，然后点击“发送”按钮就可以发送数据了（其中发送格式下拉框中的自发自收选项表示发送出去的 CAN 帧自己也能收到，这个选项在测试的时候才需用到，在实际的应用中请选用正常发送）。

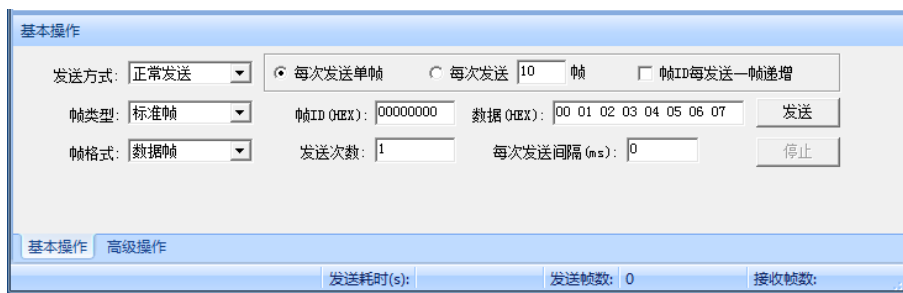


图 4.14 发送数据基本设置

您还可以点击“高级操作”标签进入高级操作页面，在此页面您可以设置每次发送多个不同的 CAN 帧（最多可设置 100 帧），和每帧之间间隔、每批之间间隔，如图 4.15 所示。

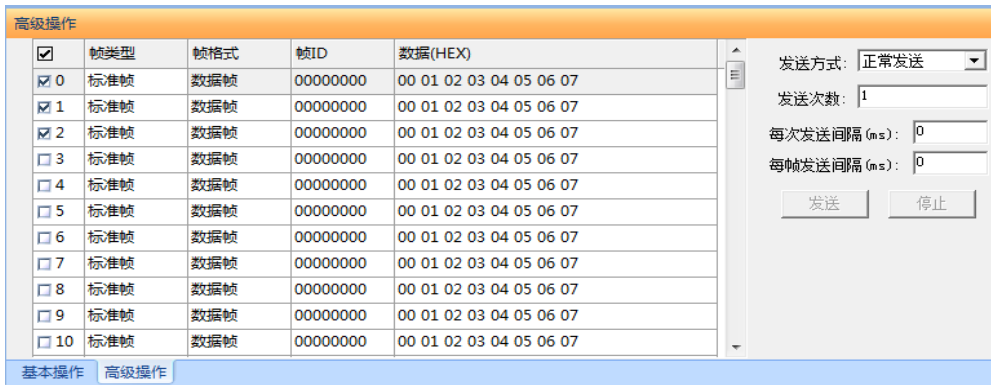


图 4.15 发送数据高级设置

发送接收效果如图 4.16、图 4.17 所示。

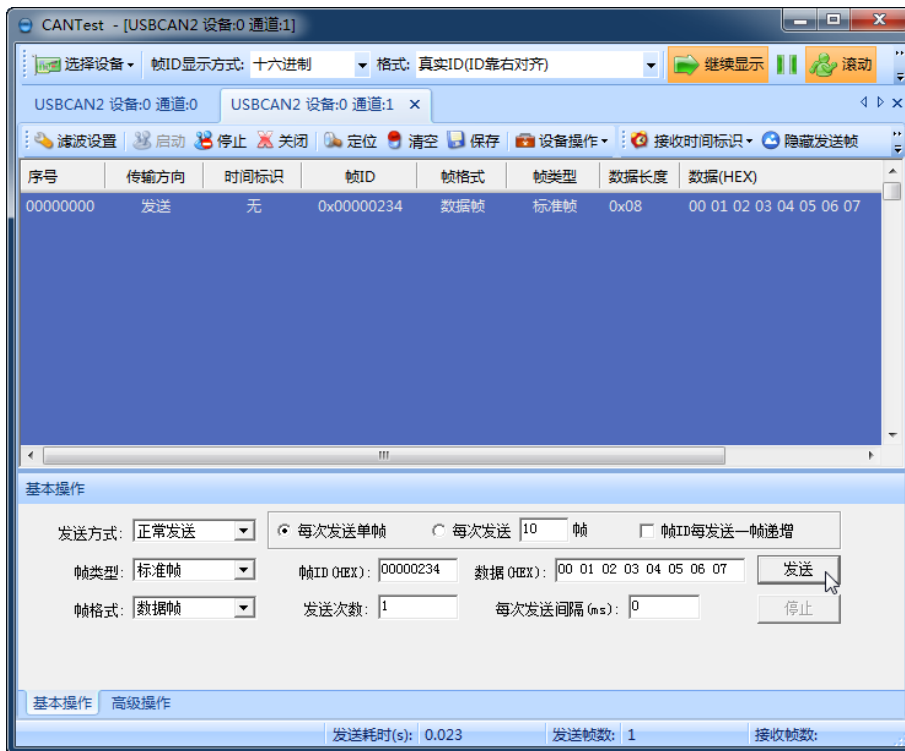


图 4.16 发送

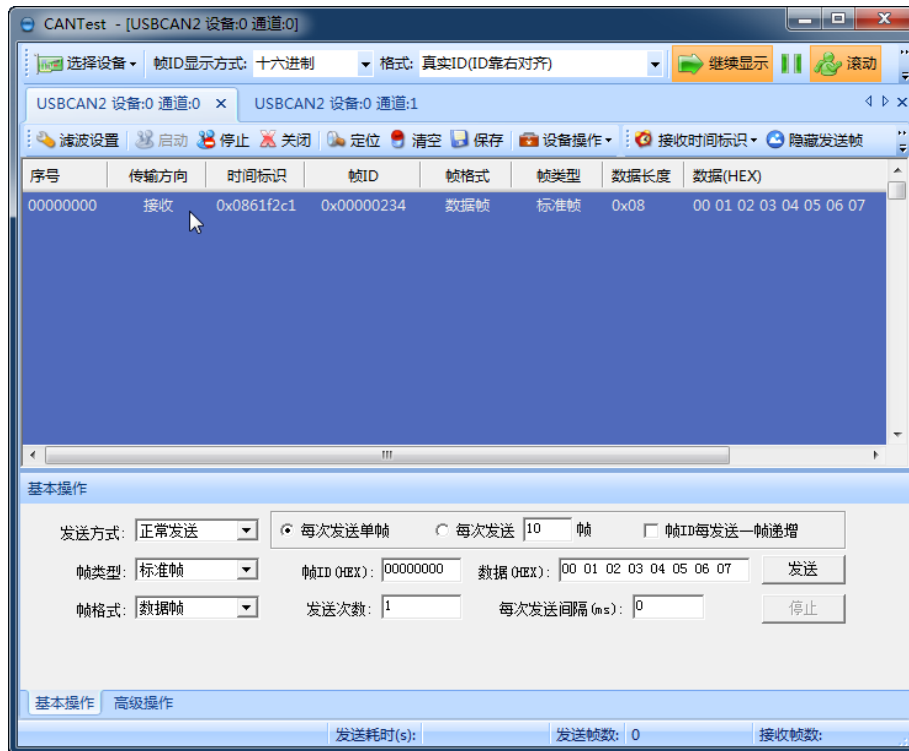


图 4.17 接收

4.2.4 实时保存与停止保存

当用户需要长时间记录报文时，需要使用实时保存功能，当软件缓冲区记录满之后，转存到硬盘中的文件（CSV 格式），软件缓冲区清空。报文文件名可以自动依次编号。需要在启动之前使能此功能，注意保存位置不能指定在 C 盘，可能无法保存。点击停止保存时，则不进行转存，如图 4.18 所示。

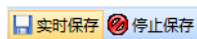



图 4.18 保存

4.2.5 DBC 解码与按 ID 分类显示

用户点击  后，打开 DBC 界面，用户可以导入需要的 DBC 文件进行帧解码（解码显示在界面下方，默认自带 J1939 解码），或者用户使用此界面对 CAN 帧进行按 ID 分类的显示，即“ID 固定，数据变化”。有变化的数据段会标红。如图 4.19 所示。

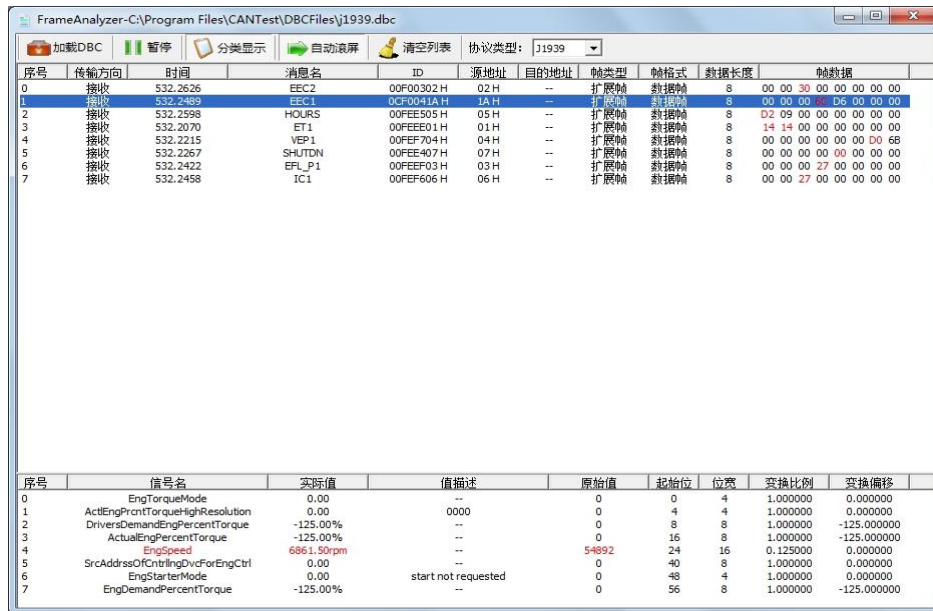
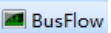


图 4.19 DBC 协议解析

4.2.6 总线利用率

点击 ，可以打开总线利用率的界面。可以实时监测目前总线的利用率与帧流量。可以调整刷新时间来调整显示速度。如图 4.20 所示。

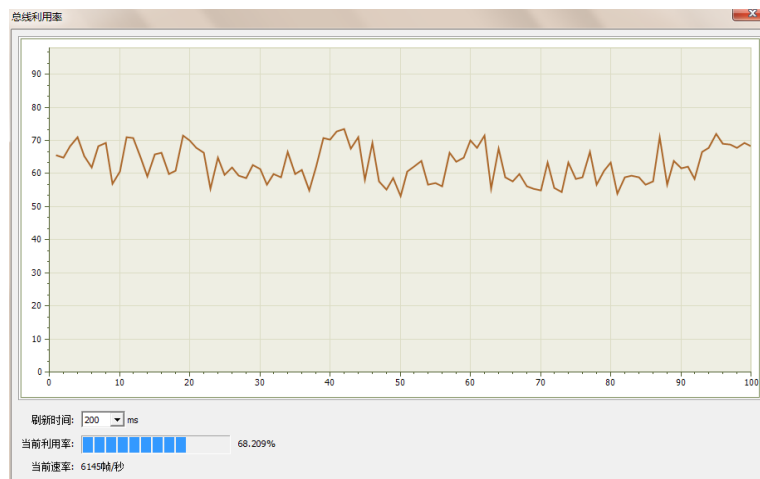

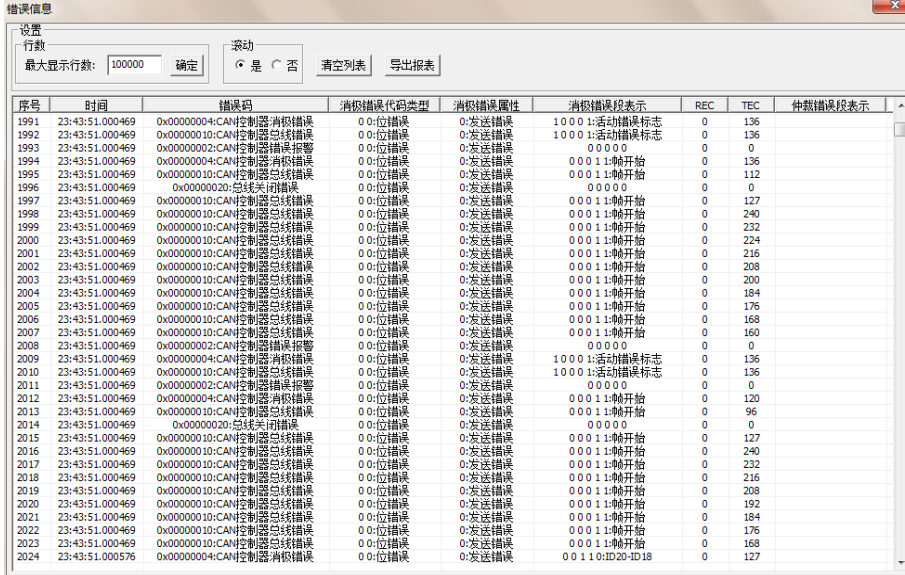


图 4.20 总线利用率

4.2.7 错误信息显示

点击  错误信息，打开错误信息显示界面，当对应 CAN 路发生错误时，则会打印出错误信息（发送错误计数器与接收错误计数器值）、错误发生的时间。如图 4.21 所示。



错误信息窗口包含以下元素：

- 设置：行、最大显示行数: 100000、确定、滚动、是、否、清空列表、导出报表
- 表格列：序号、时间、错误码、消极错误代码类型、消极错误属性、消极错误段表示、REC、TEC、仲裁错误段表示

序号	时间	错误码	消极错误代码类型	消极错误属性	消极错误段表示	REC	TEC	仲裁错误段表示
1991	23:43:51.000469	0x00000004:CAN控制器消极错误	0:0:位错误	0:发送错误	10 0 0 1:活动错误标志	0	136	
1992	23:43:51.000469	0x00000010:CAN控制器总线错误	0:0:位错误	0:发送错误	10 0 0 1:活动错误标志	0	136	
1993	23:43:51.000469	0x00000002:CAN控制器错误报警	0:0:位错误	0:发送错误	0 0 0 0	0	0	
1994	23:43:51.000469	0x00000004:CAN控制器消极错误	0:0:位错误	0:发送错误	0 0 0 1:帧开始	0	136	
1995	23:43:51.000469	0x00000010:CAN控制器总线错误	0:0:位错误	0:发送错误	0 0 0 1:帧开始	0	112	
1996	23:43:51.000469	0x00000020:总线关闭错误	0:0:位错误	0:发送错误	0 0 0 0	0	0	
1997	23:43:51.000469	0x00000010:CAN控制器总线错误	0:0:位错误	0:发送错误	0 0 0 1:帧开始	0	127	
1998	23:43:51.000469	0x00000010:CAN控制器总线错误	0:0:位错误	0:发送错误	0 0 0 1:帧开始	0	240	
1999	23:43:51.000469	0x00000010:CAN控制器总线错误	0:0:位错误	0:发送错误	0 0 0 1:帧开始	0	232	
2000	23:43:51.000469	0x00000010:CAN控制器总线错误	0:0:位错误	0:发送错误	0 0 0 1:帧开始	0	224	
2001	23:43:51.000469	0x00000010:CAN控制器总线错误	0:0:位错误	0:发送错误	0 0 0 1:帧开始	0	216	
2002	23:43:51.000469	0x00000010:CAN控制器总线错误	0:0:位错误	0:发送错误	0 0 0 1:帧开始	0	208	
2003	23:43:51.000469	0x00000010:CAN控制器总线错误	0:0:位错误	0:发送错误	0 0 0 1:帧开始	0	200	
2004	23:43:51.000469	0x00000010:CAN控制器总线错误	0:0:位错误	0:发送错误	0 0 0 1:帧开始	0	184	
2005	23:43:51.000469	0x00000010:CAN控制器总线错误	0:0:位错误	0:发送错误	0 0 0 1:帧开始	0	176	
2006	23:43:51.000469	0x00000010:CAN控制器总线错误	0:0:位错误	0:发送错误	0 0 0 1:帧开始	0	168	
2007	23:43:51.000469	0x00000010:CAN控制器总线错误	0:0:位错误	0:发送错误	0 0 0 1:帧开始	0	160	
2008	23:43:51.000469	0x00000002:CAN控制器错误报警	0:0:位错误	0:发送错误	0 0 0 0	0	0	
2009	23:43:51.000469	0x00000004:CAN控制器消极错误	0:0:位错误	0:发送错误	10 0 0 1:活动错误标志	0	136	
2010	23:43:51.000469	0x00000010:CAN控制器总线错误	0:0:位错误	0:发送错误	10 0 0 1:活动错误标志	0	136	
2011	23:43:51.000469	0x00000002:CAN控制器错误报警	0:0:位错误	0:发送错误	0 0 0 0	0	0	
2012	23:43:51.000469	0x00000004:CAN控制器消极错误	0:0:位错误	0:发送错误	0 0 0 1:帧开始	0	120	
2013	23:43:51.000469	0x00000010:CAN控制器总线错误	0:0:位错误	0:发送错误	0 0 0 1:帧开始	0	96	
2014	23:43:51.000469	0x00000020:总线关闭错误	0:0:位错误	0:发送错误	0 0 0 0	0	0	
2015	23:43:51.000469	0x00000010:CAN控制器总线错误	0:0:位错误	0:发送错误	0 0 0 1:帧开始	0	127	
2016	23:43:51.000469	0x00000010:CAN控制器总线错误	0:0:位错误	0:发送错误	0 0 0 1:帧开始	0	240	
2017	23:43:51.000469	0x00000010:CAN控制器总线错误	0:0:位错误	0:发送错误	0 0 0 1:帧开始	0	232	
2018	23:43:51.000469	0x00000010:CAN控制器总线错误	0:0:位错误	0:发送错误	0 0 0 1:帧开始	0	216	
2019	23:43:51.000469	0x00000010:CAN控制器总线错误	0:0:位错误	0:发送错误	0 0 0 1:帧开始	0	208	
2020	23:43:51.000469	0x00000010:CAN控制器总线错误	0:0:位错误	0:发送错误	0 0 0 1:帧开始	0	192	
2021	23:43:51.000469	0x00000010:CAN控制器总线错误	0:0:位错误	0:发送错误	0 0 0 1:帧开始	0	184	
2022	23:43:51.000469	0x00000010:CAN控制器总线错误	0:0:位错误	0:发送错误	0 0 0 1:帧开始	0	176	
2023	23:43:51.000469	0x00000010:CAN控制器总线错误	0:0:位错误	0:发送错误	0 0 0 1:帧开始	0	168	
2024	23:43:51.000576	0x00000004:CAN控制器消极错误	0:0:位错误	0:发送错误	0 0 1 1 0:ID20-ID18	0	127	

图 4.21 错误信息

5. 接口库函数使用方法

高效易用的二次开发函数，可支持各类开发环境，如 VC，C#，Labview，Linux 等，如图 5.1 所示。

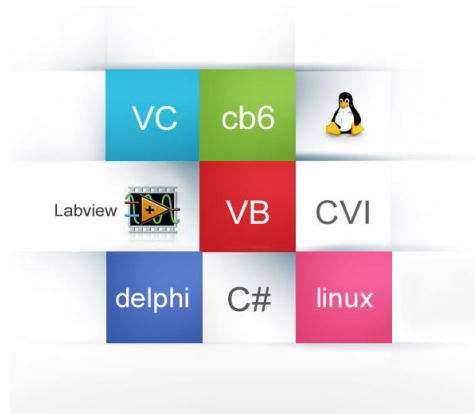


图 5.1 支持开发环境

5.1 在 windows 下调用动态库的方法

首先，把库函数文件都放在工作目录下。库函数文件总共有三个文件：ControlCAN.h、ControlCAN.lib、ControlCAN.dll 和一个文件夹 kerneldlls。

5.1.1 VC 调用动态库的方法

- (1) 在扩展名为.CPP 的文件中包含 ControlCAN.h 头文件。
如：`#include "ControlCAN.h"`
- (2) 在工程的连接器设置中连接到 ControlCAN.lib 文件。
如：在 VC7 环境下，在项目属性页里的配置属性→连接器→输入→附加依赖项中添加 ControlCAN.lib

5.1.2 VB 调用动态库的方法

通过以下方法进行声明后就可以调用了。

语法：

```
[Public | Private] Declare Function name Lib "libname" [Alias "aliasname"]
[[([arglist])] [As type]
```

Declare 语句的语法包含下面部分：

Public (可选)

用于声明在所有模块中的所有过程都可以使用的函数。

Private (可选)

用于声明只能在包含该声明的模块中使用的函数。

Name (必选)

任何合法的函数名。动态链接库的入口处 (entry points) 区分大小写。

Libname (必选)

包含所声明的函数动态链接库名或代码资源名。

Alias (可选)

表示将被调用的函数在动态链接库 (DLL) 中还有另外的名称。当外部函数名与某个函数重名时, 就可以使用这个参数。当动态链接库的函数与同一范围内的公用变量、常数或任何其它过程的名称相同时, 也可以使用 Alias。如果该动态链接库函数中的某个字符不符合动态链接库的命名约定时, 也可以使用 Alias。

Aliasname (可选)

动态链接库。如果首字符不是数字符号 (#), 则 `aliasname` 是动态链接库中该函数入口处的名称。如果首字符是 (#), 则随后的字符必须指定该函数入口处的顺序号。

Arglist (可选)

代表调用该函数时需要传递参数的变量表。

Type (可选)

Function 返回值的数据类型; 可以是 Byte、Boolean、Integer、Long、Currency、Single、Double、Decimal (目前尚不支持)、Date、String (只支持变长) 或 Variant, 用户定义类型, 或对象类型。

`arglist` 参数的语法如下:

```
[Optional] [ByVal | ByRef] [ParamArray] varname[( )][As type]
```

部分描述:

Optional (可选)

表示参数不是必需的。如果使用该选项, 则 `arglist` 中的后续参数都必需是可选的, 而且必须都使用 `Optional` 关键字声明。如果使用了 `ParamArray`, 则任何参数都不能使用 `Optional`。

ByVal (可选)

表示该参数按值传递。

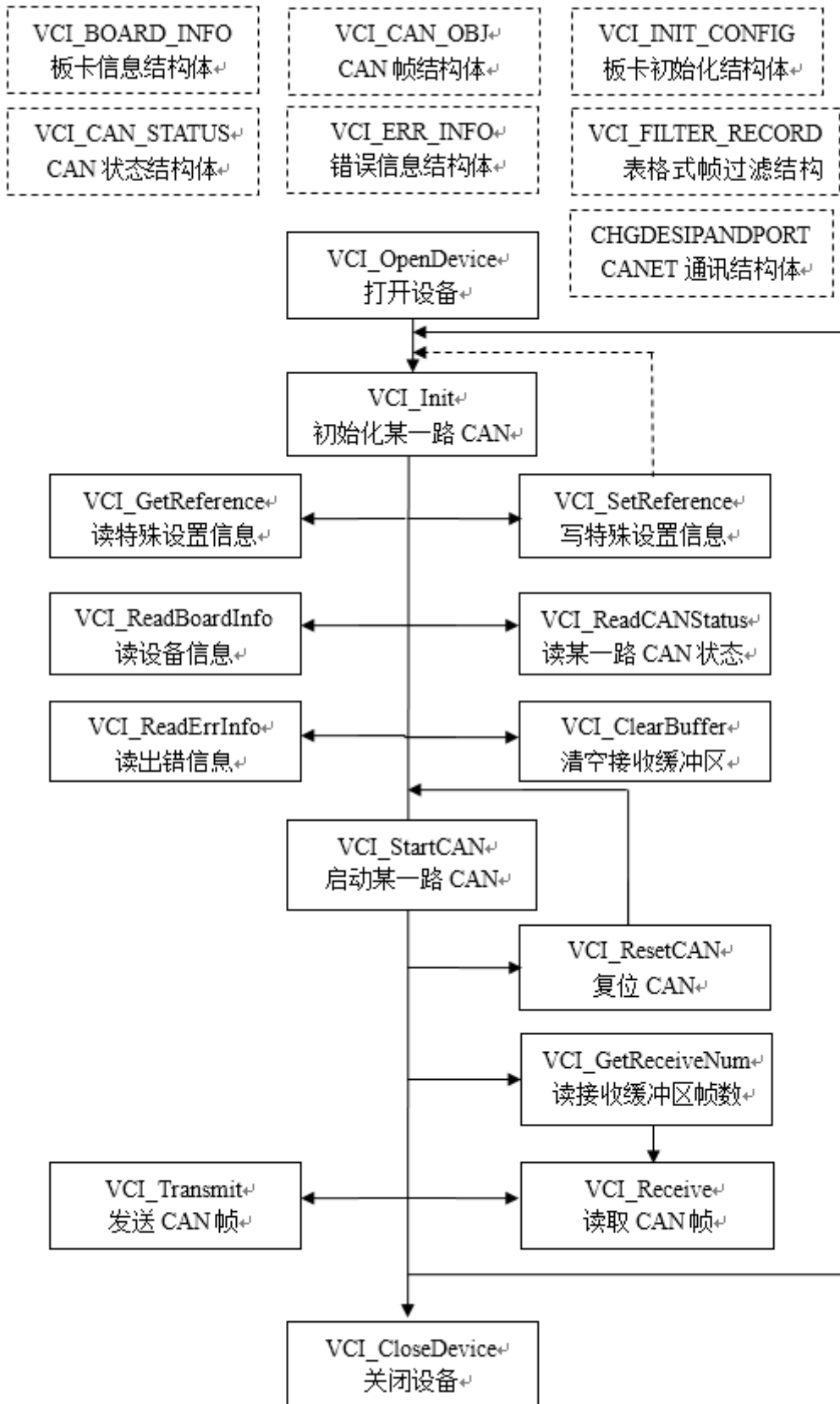
ByRef (可选)

表示该参数按地址传递。

例如:

```
Public Declare Function VCI_OpenDevice Lib "ControlCAN" (ByVal devicetype As Long,
ByVal deviceind As Long, ByVal reserved As Long) As Long
```


5.2 接口库函数使用流程



6. 电气特性

除非特别说明，如表 6.1 所列参数是指 $T_{amb}=25^{\circ}\text{C}$ 时的值。

表 6.1 电气特性

参数名称	典型值	备注
电源电压	3.3V	
工作电流	200 mA(+3.3V 供电电压)	
静电等级	4KV(接触)/8KV(空气)	
温度范围	-40~+85°C	

7. 结构尺寸

板卡尺寸：30mm（宽）× 50.95mm（长）× 7.35mm（高）。详细尺寸如图 7.1 以及图 7.2 所示。单位：mm（毫米）。

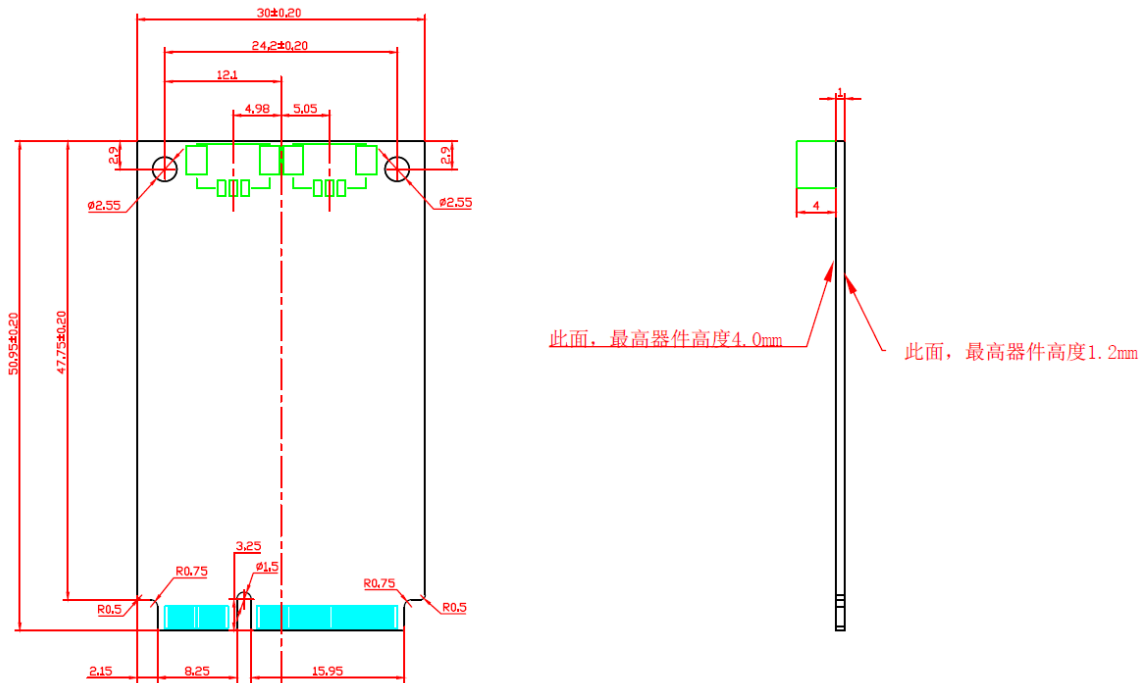


图 7.1 主板结构尺寸

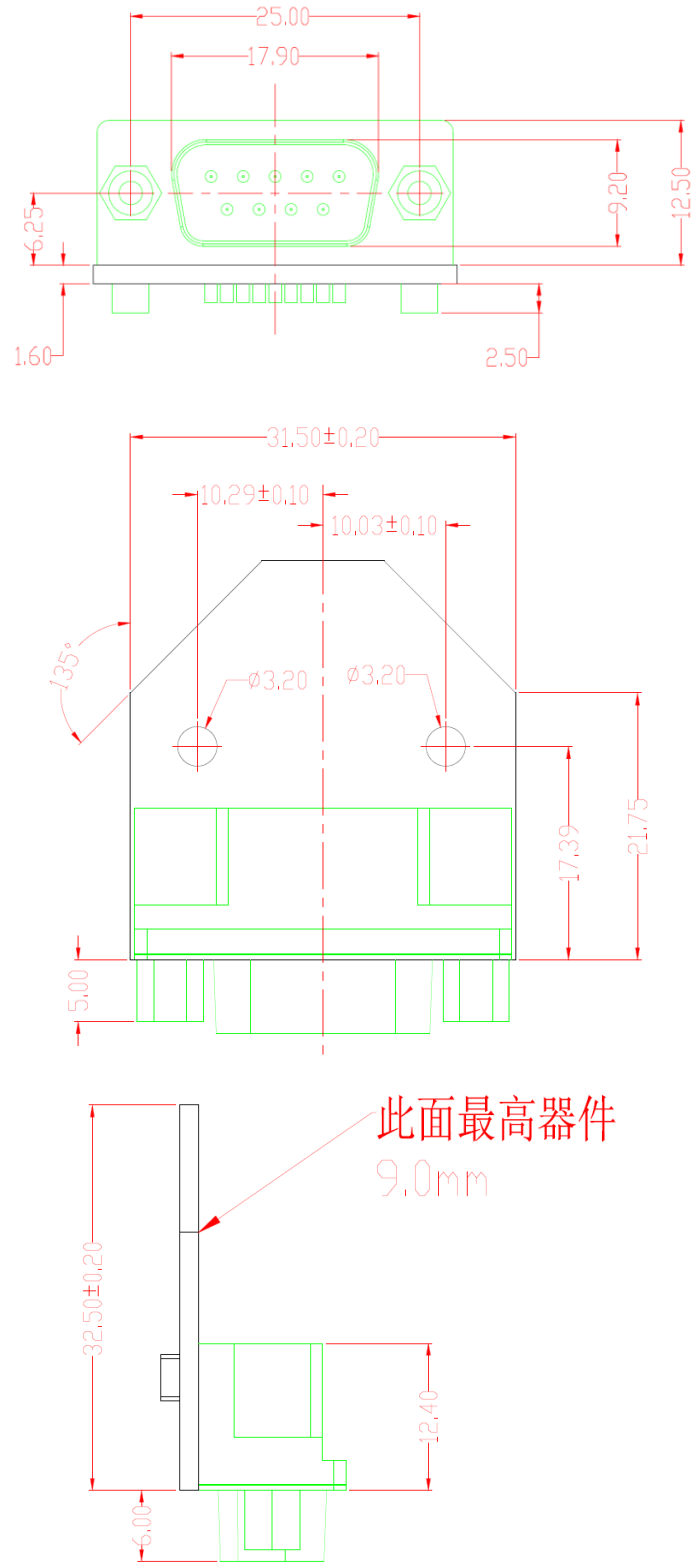


图 7.2 PACK 板尺寸

8. 检查和维护

MiniPCle 接口 CAN 卡的主要电气部件都是半导体元件，尽管它有很长的寿命，但在不正确环境下也可能加速老化。应该进行定期检查，以保证保持所要求的条件。推荐每 6 月到一年，至少检查一次。在不利的环境条件下，应该进行更频繁的检查。

如果在维护过程中遇到问题，请阅读表 8.1 的内容，以便找到问题的可能的原因。如果仍无法解决问题，请联系广州致远电子有限公司。

表 8.1 检查与维护

序号	项目	检查	标准	行动
1	电源供应	在电源供应端 检查电压波动	MiniPCle 电源+3.3V DC	使用电压表检查 供电电压。
2	周围环境	检查周围环境 温度（包括封 闭环境的内部 温度）	-40 ℃ ~ +85 ℃	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）	没有空调时相对湿度必须在 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
		检查灰尘、粉 末、盐、金属 屑的积累	没有积累	清洁并保护设备
		检查水、油或 化学喷雾碰撞 到设备	没有喷雾碰到设备	如果需要清洁保 护设备
		检查在设备区 域中易腐蚀或 易燃气体	没有易腐蚀或易燃气体	通过闻或使用一 个传感器检查
		检查震动和冲 击水平	震动和冲击在规定范围内	如果需要安装衬 垫或其它减震装 置
		检查设备附近 的噪声源	没有重要噪声信号源	隔离设备和噪声 源或保护设备
3	安装和接线	检查每个单元 的连接并已经 与下一个单元 安全锁定	没有松动	把连接器完全压 到一起和用滑块 把它们锁住

序号	项目	检查	标准	行动
3	安装和接线	检查电缆连接器完全插入和锁住	没有松动	纠正任何不正确安装连接器
		检查外部接线中是否有松动螺丝钉	没有松动	用螺丝起子拧紧螺丝钉
		检查外部接线中的压接连接器	在连接器间有足够的空间	肉眼检查如果有必要则调节
		检查外部线电缆的损坏	没有损坏	肉眼检查和如果有必须则替换电缆



9. 免责声明

广州致远电子有限公司隶属于广州立功科技股份有限公司。本着为用户提供更好服务的原则，广州致远电子有限公司（下称“致远电子”）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，致远电子不能完全保证该文档在任何时段的时效性与适用性。致远电子有权在没有通知的情况下对本手册上的内容进行更新，恕不另行通知。为了得到最新版本的信息，请尊敬的用户定时访问致远电子官方网站或者与致远电子工作人员联系。感谢您的包容与支持！

附录A SJA1000 标准波特率

序号	Baudrate (Kbps)	晶振频率=16MHz	
		BTR0 (Hex)	BTR1 (Hex)
1	5	BF	FF
2	10*	31	1C
3	20*	18	1C
4	40	87	FF
5	50*	09	1C
6	80	83	FF
7	100*	04	1C
8	125*	03	1C
9	200	81	FA
10	250*	01	1C
11	400	80	FA
12	500*	00	1C
13	666	80	B6
14	800*	00	16
15	1000*	00	14

注：带*号的是 CIA 协会推荐的波特率。