

USBCAN-E-mini

小型智能 CAN 接口卡

UM01010101 V1.04 Date: 2019/03/18

产品用户手册

类别	内容
关键词	USBCAN-E-mini、高性能、便携
摘要	USBCAN-E-mini 符合 CAN2.0A/B 规范,支持 5Kbps~1Mbps 之间的任意波特率,提供多个操作系统的设备驱动,满足各种应用需求,为工业通讯 CAN 网络提供了可靠性、高效率的解决方案。

修订历史

版本	日期	原因
V1.00	2015/10/8	创建文档
V1.01	2017/08/09	更改公司名称、销售与服务网络
V1.02	2018/03/01	更改部分描述错误
V1.03	2018/06/15	更改驱动安装描述
V1.04	2019/03/18	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

目录

1. 功能简介.....	1
1.1 产品概述.....	1
1.2 参数指标.....	1
1.3 产品外观.....	2
1.4 典型应用.....	2
2. 设备安装.....	3
2.1 CAN-bus 连接器	3
2.2 信号指示灯.....	3
2.3 系统连接.....	4
2.3.1 CAN 总线连接	4
2.3.2 总线终端电阻.....	4
3. 驱动程序安装.....	6
3.1 安装驱动.....	6
4. 快速使用指南.....	10
4.1 CANTest 基本操作.....	10
4.1.1 设备类型选择.....	10
4.1.2 滤波设置.....	11
4.1.3 启动 CAN	12
4.1.4 获取设备信息.....	13
4.2 发送接收实验.....	14
4.2.1 搭建测试环境.....	14
4.2.2 打开设备.....	14
4.2.3 发送数据.....	14
4.2.4 实时保存与停止保存.....	16
4.2.5 总线利用率.....	16
4.2.6 错误信息显示.....	17
5. 接口库函数使用方法.....	18
5.1 在 windows 下调用动态库的方法	18
5.1.1 VC 调用动态库的方法	18
5.1.2 VB 调用动态库的方法	18
5.2 接口库函数使用流程.....	20
6. 检查和维护.....	21
7. 免责声明.....	23
附录 A DB9 转 OBD 接口.....	24
A.1 功能简介.....	24
A.2 技术参数.....	24
A.3 引脚信息.....	24
A.3.1 DB9 接头引脚排列	24
A.3.2 DB9 接头引脚描述	25
A.3.3 OBD 接头引脚排列	25
A.3.4 OBD 接头引脚描述	26



A.4 机械尺寸.....	26
附录 B ARM 标准波特率.....	27

1. 功能简介

1.1 产品概述

USBCAN-E-mini 智能 CAN 接口卡是系列 USBCAN 便携版本, 与 USBCAN—E-U 单路智能 CAN 接口卡完全兼容。USBCAN-E-mini 智能 CAN 接口卡与 USB1.1 总线兼容的, 集成 1 路 CAN 接口的智能型 CAN-bus 总线通讯接口卡。采用 USBCAN-E-mini 智能 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、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-E-mini 智能 CAN 接口卡是 CAN-bus 产品开发、CAN-bus 数据分析的强大工具; 同时, USBCAN-E-mini 接口卡具有体积小、即插即用等特点, 也是便携式系统用户的最佳选择。

USBCAN-E-mini 接口卡上自带电气隔离模块, 使 USBCAN-E-mini 接口卡避免由于地环流的损坏, 增强系统在恶劣环境中使用的可靠性。

USBCAN-E-mini 智能 CAN 接口卡支持 Win9x/Me、Win2000/XP/7/8/10 等操作系统。USBCAN-E-mini 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。

USBCAN-E-mini 接口卡还支持 OPC 接口, 能在支持 OPC 的组态软件中使用 USBCAN-E-mini 接口卡。另外, 还提供了 CANTest 通用测试软件以及 CANPro 协议分析平台软件, 可执行 CAN-bus 报文的收发和监测等功能。

1.2 参数指标

- PC 接口符合 USB2.0 协议规范, 兼容 USB3.0 和 USB1.1;
- 支持 CAN2.0A 和 CAN2.0B 协议, 符合 ISO/DIS 11898-1/2/3 标准;
- 集成 1 路 CAN-bus 接口;
- CAN-bus 通讯波特率在 5Kbps~1Mbps 之间任意可编程;
- 可以使用 USB 总线电源供电;
- CAN 通道采用电磁隔离、DC/DC 电源隔离, 隔离电压: 3000VDC;
- 单通道最高数据流量 (1Mbps): 14000 帧/秒(接收: 远程帧), 3000 帧/秒 (发送);
- 支持 Win9x/Me、Win2000、WinXP/7/8/10 等 Windows 操作系统;
- CAN 接口 EMC 等级: 接触放电 ±4KV, 群脉冲 ±1KV。

1.3 产品外观



图 1.1 USBCAN-E-mini 智能 CAN 接口卡 (1)



图 1.2 USBCAN-E-mini 智能 CAN 接口卡 (2)

1.4 典型应用

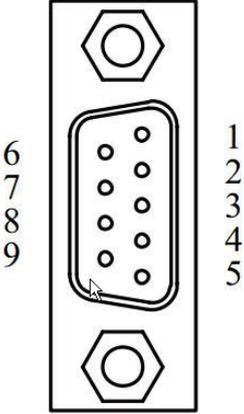
- CAN-bus 网络诊断与测试;
- 汽车电子应用;
- 电力通讯网络;
- 工业控制设备;
- 高速、大数据量通讯。

2. 设备安装

2.1 CAN-bus 连接器

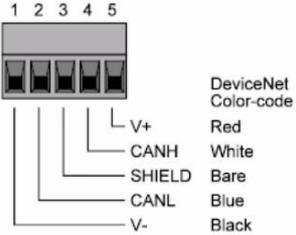
USBCAN-E-mini 接口卡集成 1 路 CAN-bus 通道，通过 DB9 针型插座或 DB9 孔型插座与实际的 CAN-bus 网络进行连接。DB9 插座的管脚信号定义如表 2.1 所示，管脚信号定义符合 DeviceNet 和 CANopen 标准。

表 2.1 CAN-bus 总线的信号连接(DB9 插座)

DB9 针型插座	引脚	信号	描述
	1	N.C.	未用
	2	CAN_L	CAN_L 信号线
	3	CAN_GND	参考地
	4	N.C.	未用
	5	CAN_SHIELD	屏蔽线
	6	CAN_GND	参考地
	7	CAN_H	CAN_H 信号线
	8	N.C.	未用
	9	N.C.	未用

用户可以通过选配的 DB9OPEN5 转换器，将 DB9 插座的 CAN-bus 信号转换至易于连接的 5 引脚 OPEN5 连接器，接口说明见表 2.2。

表 2.2 DB9OPEN5 转换器的信号分配(OPEN5 插座)

OPEN5 插座	引脚	信号	描述
	1	V-	网络电源负极
	2	CAN_L	CAN_L 信号线
	3	SHIELD	屏蔽线 (FG)
	4	CAN_H	CAN_H 信号线
	5	V+	网络电源正极

2.2 信号指示灯

USBCAN-E-mini 接口卡具有 1 个双色 SYS 指示灯、1 个双色 CAN 指示灯来指示 CAN 接口的运行状态。这 2 个指示灯的具体指示功能见表 2.3 和表 2.4

表 2. 3USBCAN-E-mini 接口卡的指示灯

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

- USBCAN-E-mini 接口卡上电后，指示灯 SYS 点亮并呈红色，表明设备已经供电，系统正在初始化；若指示灯 SYS 不亮，表示存在系统电源故障或系统发生有严重的错误。
- USB 接口连接正常后，指示灯 SYS 从红色转为绿色状态（驱动已装上）。当 USB 接口有数据在传输时，指示灯 SYS 呈绿色并闪烁。
- CAN 指示灯点亮表示 CAN 控制器已完成初始化，进入正常工作状态。

表 2. 4USBCAN-E-mini 接口卡的指示灯状态

CAN 指示灯状态	CAN 总线状态
CAN 灭	CAN 控制器与总线断开（该通道还没启动）
CAN 绿色指示灯常亮	CAN 总线运行正常
CAN 绿色指示灯常亮，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 系统连接

2.3.1 CAN 总线连接

USBCAN-E-mini 接口卡和 CAN-bus 总线连接的时候，仅需要将 CAN_L 连 CAN_L，CAN_H 连 CAN_H 信号。

CAN-bus 网络采用直线拓扑结构，总线的 2 个终端需要安装 120 Ω 的终端电阻；如果节点数目大于 2，中间节点不需要安装 120 Ω 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 2.1 所示。

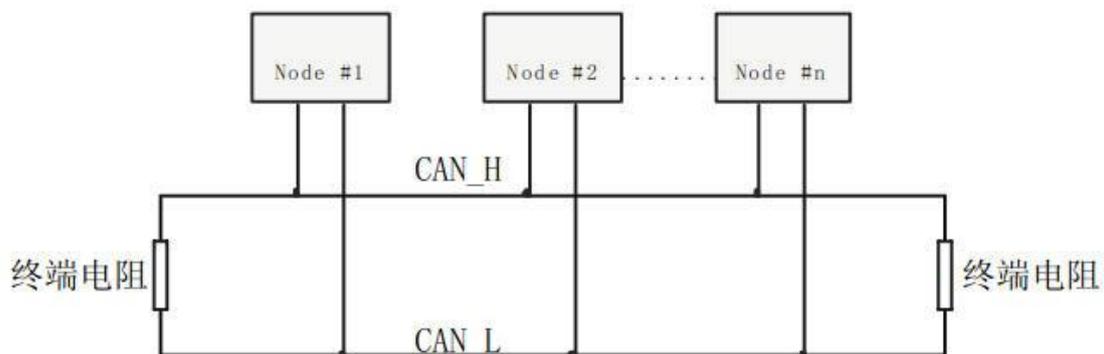


图 2. 1CAN-bus 网络的拓扑结构

注意：CAN-bus 电缆可以使用普通双绞线、屏蔽双绞线。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。

2.3.2 总线终端电阻

为了增强 CAN 通讯的可靠性，CAN 总线网络的两个端点通常要加入终端匹配电阻，如

图 2.1 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 $120\ \Omega$ 终端电阻。另外，USBCAN-E-mini 接口卡采用 FreeScale 高性能 CAN 收发器 MC33901WEF，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。

USBCAN-E-mini 智能 CAN 接口卡的内部集成有 $120\ \Omega$ 终端电阻，无需外加终端电阻。

USBCAN-E-mini 接口卡的 USB 端口符合 USB1.1 协议规范，可以与具有 USB1.1 标准、或 USB2.0 标准的 PC 机连接通讯。

3. 驱动程序安装

下面以 Win7 操作系统的 PC 为实例，说明如何正确地安装一个 USBCAN-I-mini 驱动。

3.1 安装驱动

首先把 USBCAN-E-mini 接到电脑，同时确保设备供电正常。

右击“计算机”，点击【属性】(如图 3.1 所示)，打开设备管理器，未安装驱动显示【未知设备】(如图 3.2 所示)。

如果此时设备管理器没有显示该信息，请检查：

- USB 线连接是否有问题
- 电脑的 USB 口是否被禁用
- 电脑的 USB 口供电是否正常

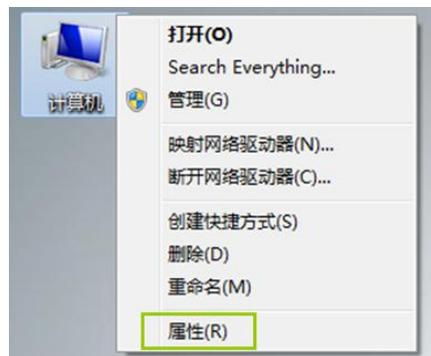


图 3.1 打开电脑设备管理器

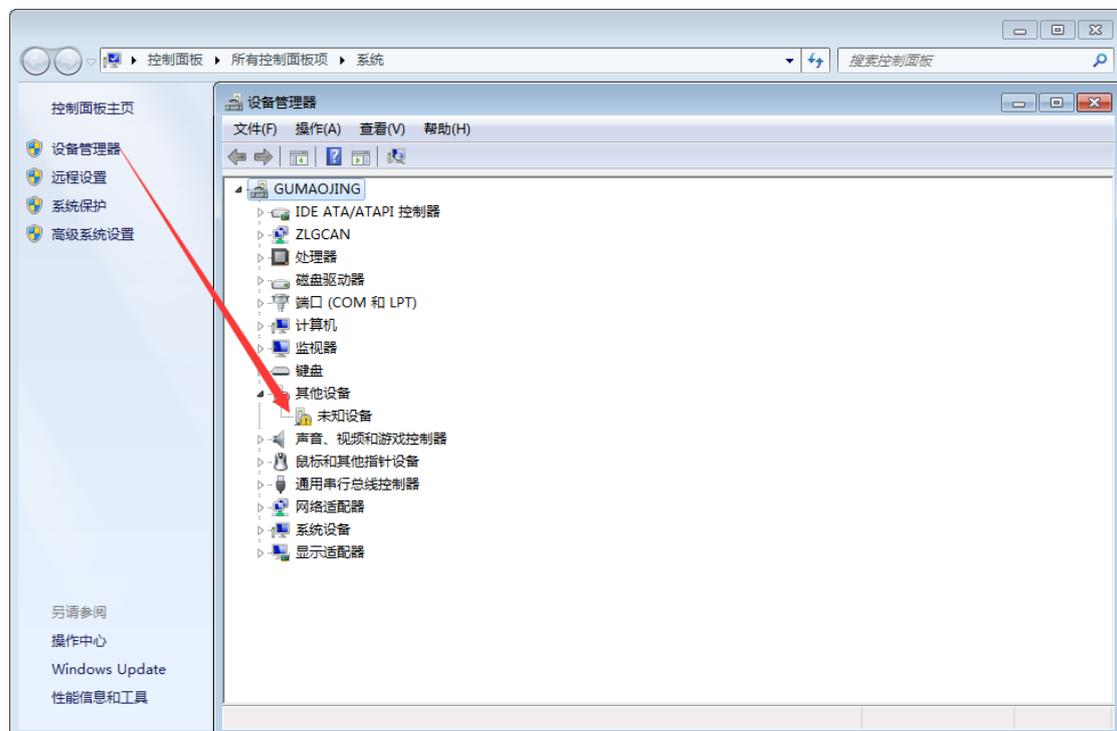


图 3.2 在设备管理器中显示未安装状态

右击【未知设备】-选择“更新驱动程序软件”。

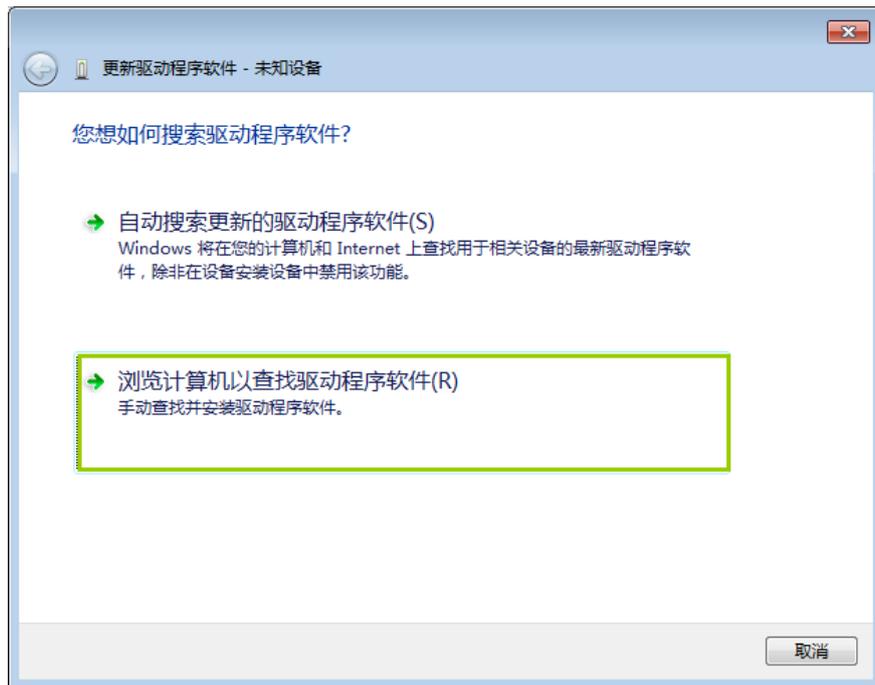


图 3.3 浏览计算机已查找驱动程序软件

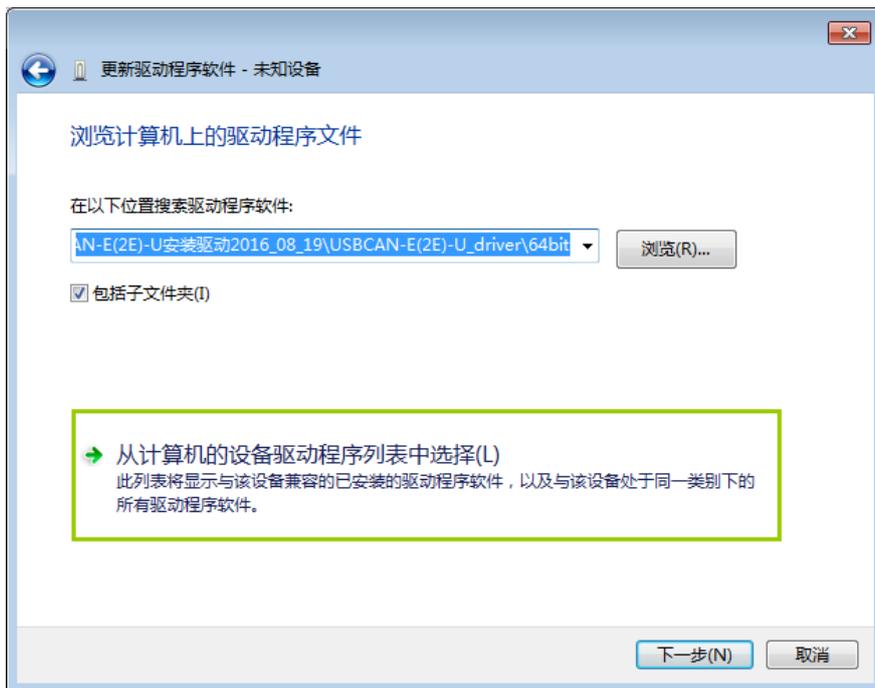


图 3.4 从计算机的设备驱动列表中选择

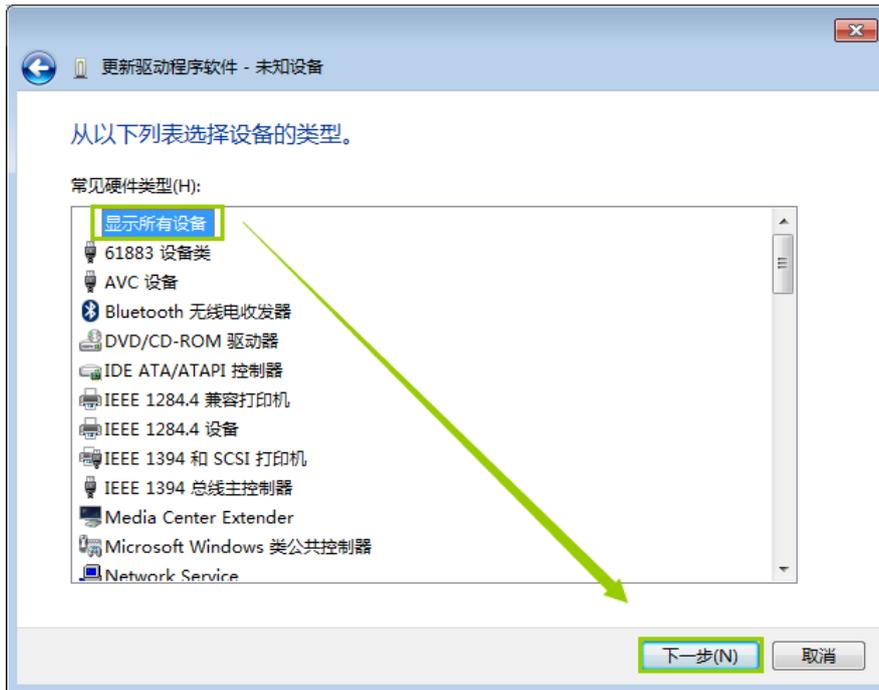


图 3.5 直接点击下一步

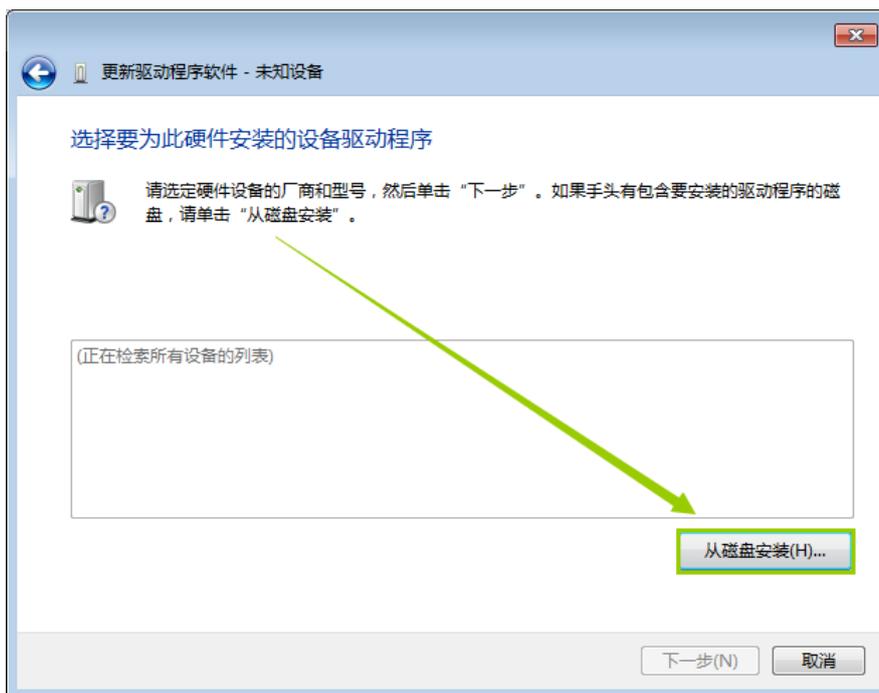


图 3.6 从磁盘安装

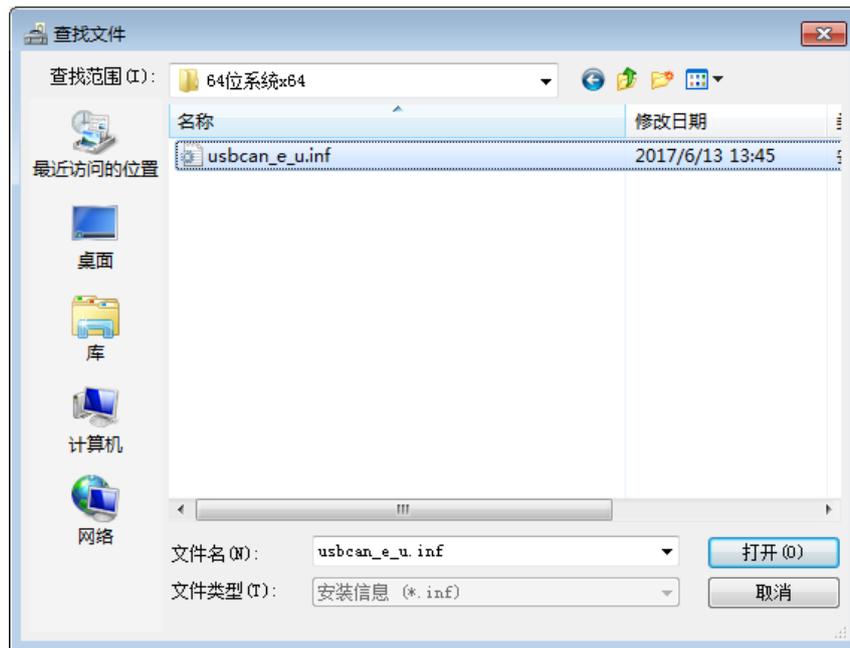


图 3.7 选择驱动文件

选中“USBCAN_e_u.inf”文件，打开并确定，电脑即开始加载安装驱动文件，接下来的窗口显示了已经安装好的设备。单击【关闭】按钮结束安装。

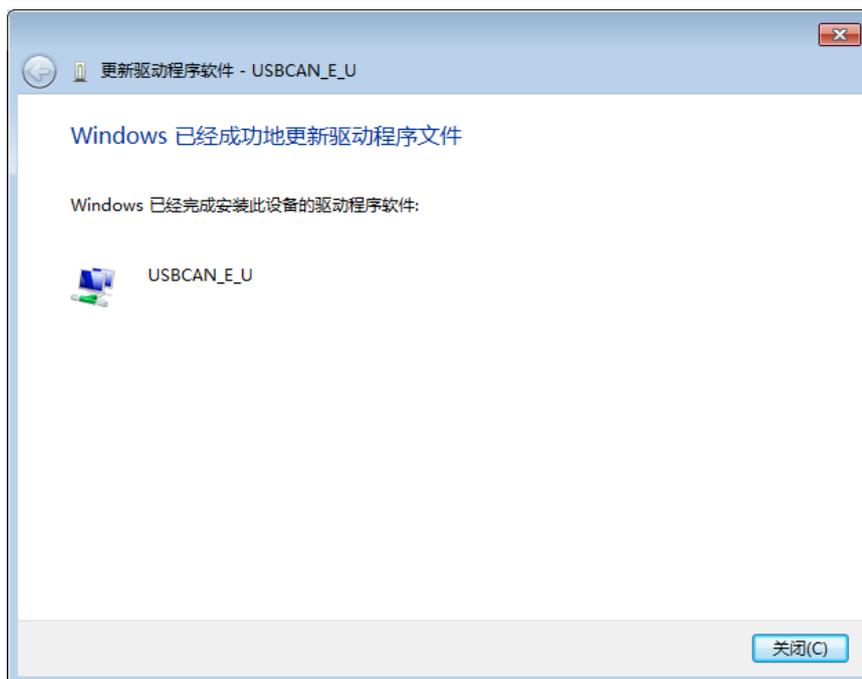
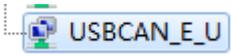


图 3.8 驱动程序安装完成

在设备管理器中，显示  ，驱动则正确安装，硬件的“SYS”灯绿色。

4. 快速使用指南

4.1 CANTest 基本操作

CANtest 测试软件的安装文件 CANTest_Setup_Vx.xx.exe 可以在配套光盘中找到。安装好后如图 4.1 所示。



图 4.1 CANTest 软件图标

注：CANTest 软件可到广州致远电子有限公司官网进行下载 <http://www.zlg.cn>。

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 USBCAN-E-U, 如图 4.2 所示。

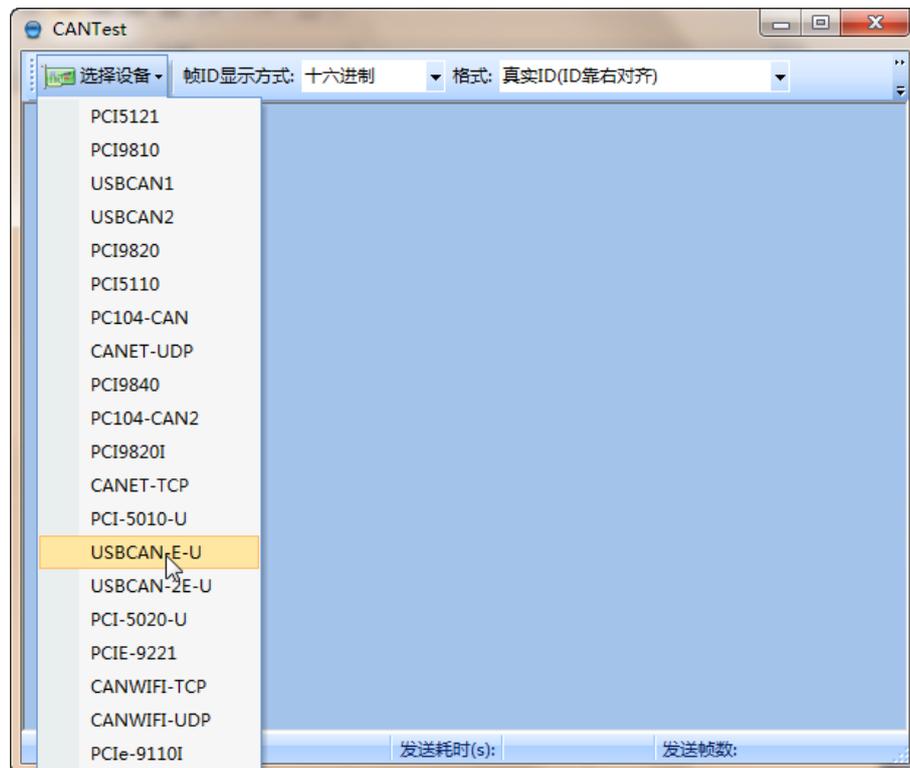


图 4.2 设备类型选择

选择确定后会弹出【打开设备】对话框, 如图 4.3 所示。

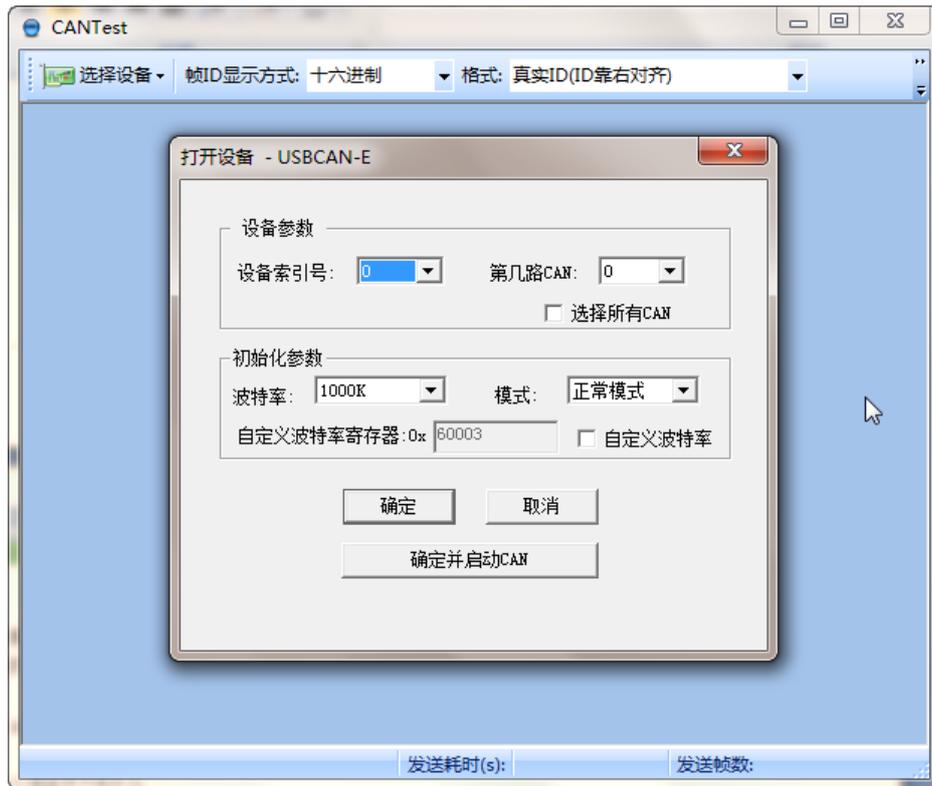


图 4.3 打开设备对话框

在这个对话框中您可以选择您要打开的设备索引号和通道号，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）。

注：设备索引号指识别设备的代码，同一设备的不同 CAN 接口要选择相同的设备索引号，不同设备则选择不同的设备索引号，设备索引号初始值为 0。

通道号是用于区分同一设备下的不同 CAN 通道，初始值为 0。由于本设备只具备单路 CAN 通道，所以通道号只能为 0。

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。

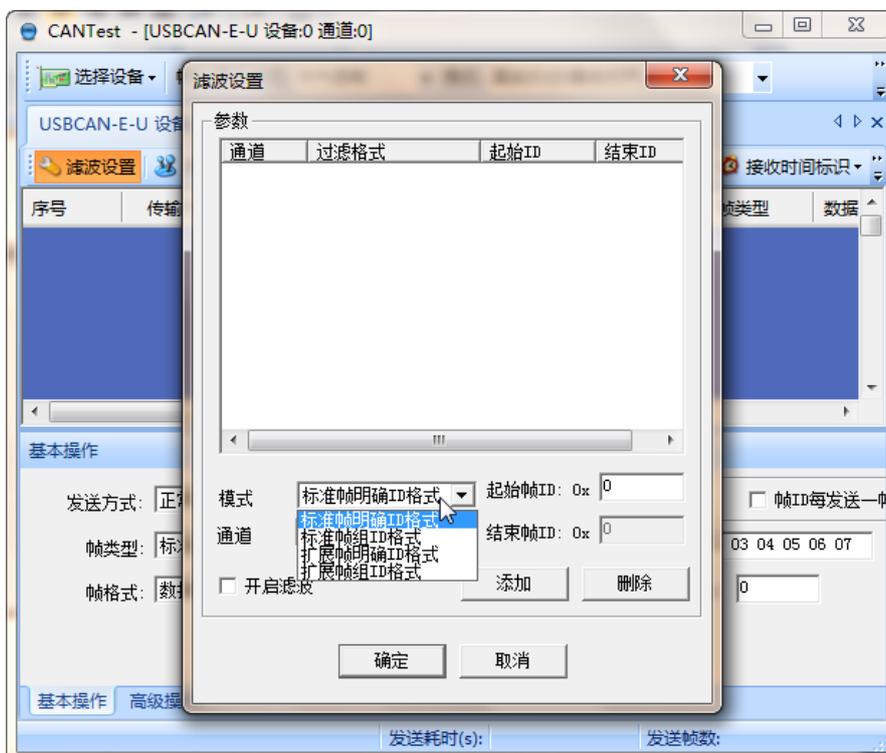


图 4.5 滤波设置 2

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。

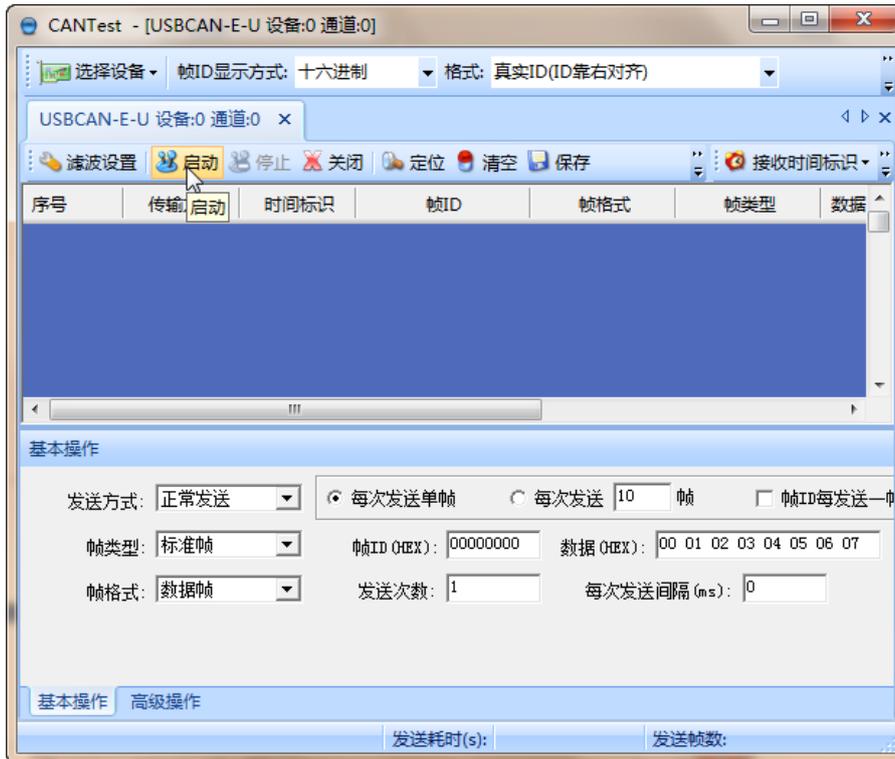


图 4.6 启动

4.1.4 获取设备信息

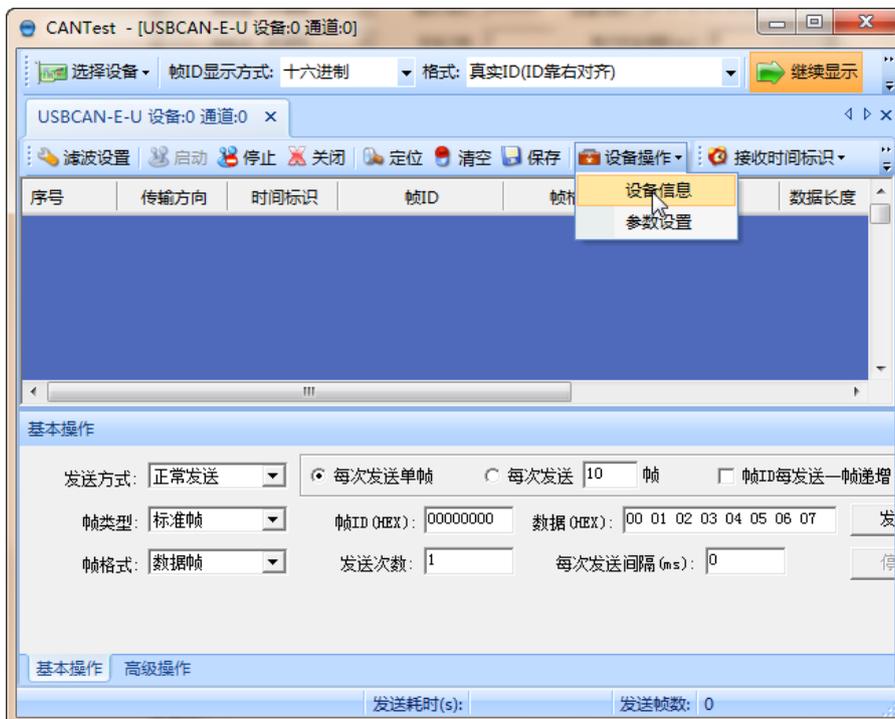


图 4.7 设备操作

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 USBCAN-E-mini 的简单发送接收测试，总线利用率的演示。

4.2.1 搭建测试环境

将设备接入电脑和被测设备，确保连线正确，接口定义如附录 B 所示。

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，按图 4.8 所示配置好设备并启动。

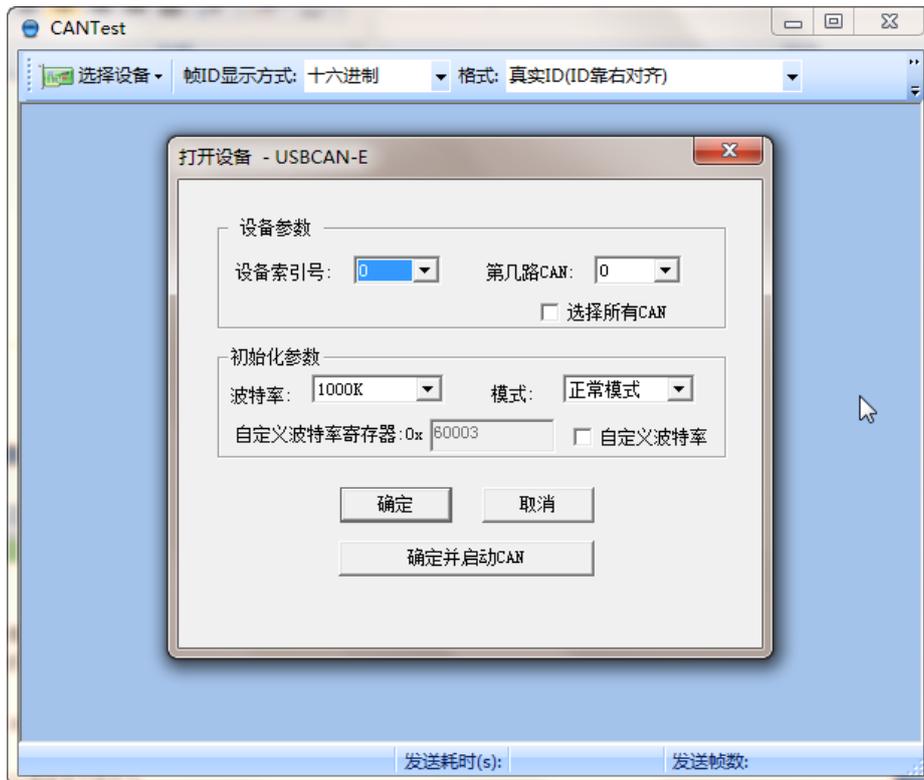


图 4.8 设备初始参数设置

注：设备索引号指识别设备的代码，同一设备的不同 CAN 接口要选择相同的设备索引号，不同设备则选择不同的设备索引号，设备索引号初始值为 0。

通道号是用于区分同一设备下的不同 CAN 通道，初始值为 0。由于本设备只具备单路 CAN 通道，所以通道号只能为 0。

4.2.3 发送数据

当您启动 CAN 成功后，在如图 9 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 4.9 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 4.10 所示。



图 4.10 发送数据高级设置

发送接收效果如图 4.11、图 4.12 所示。

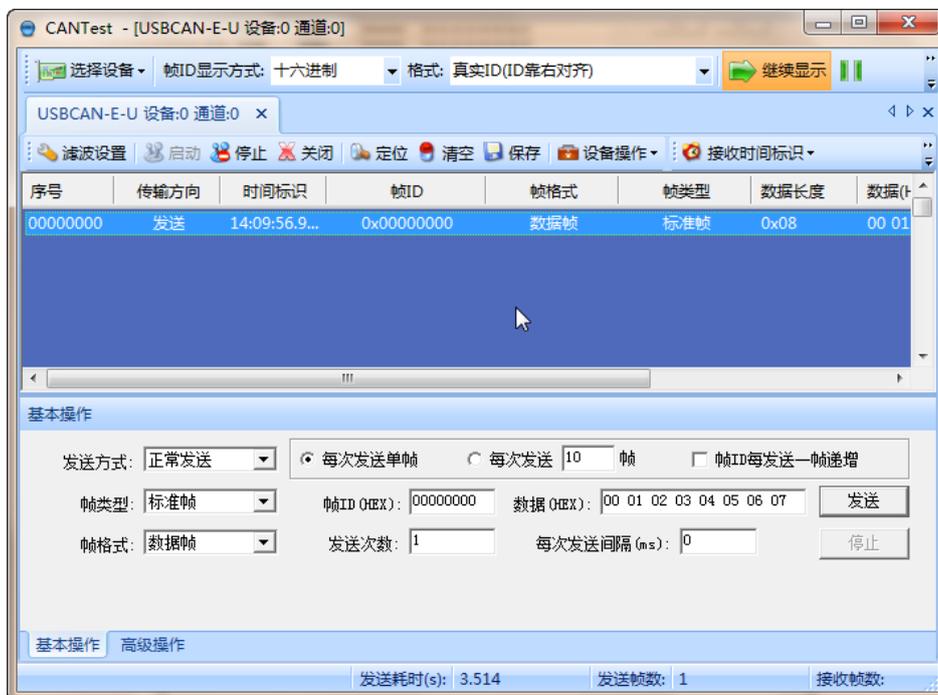


图 4.11 发送

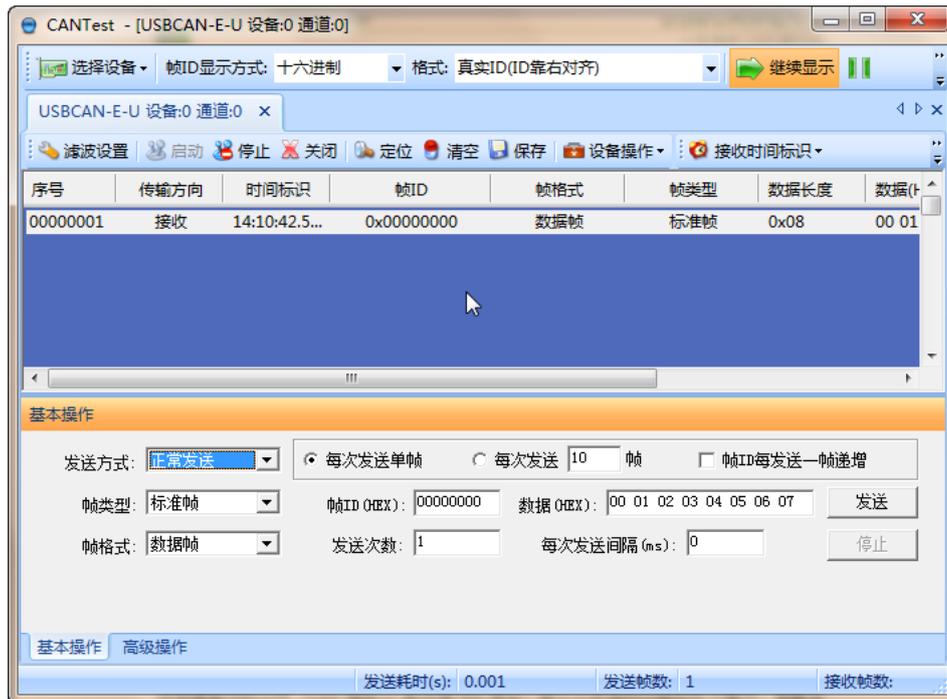


图 4.12 接收

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 4.13 所示。

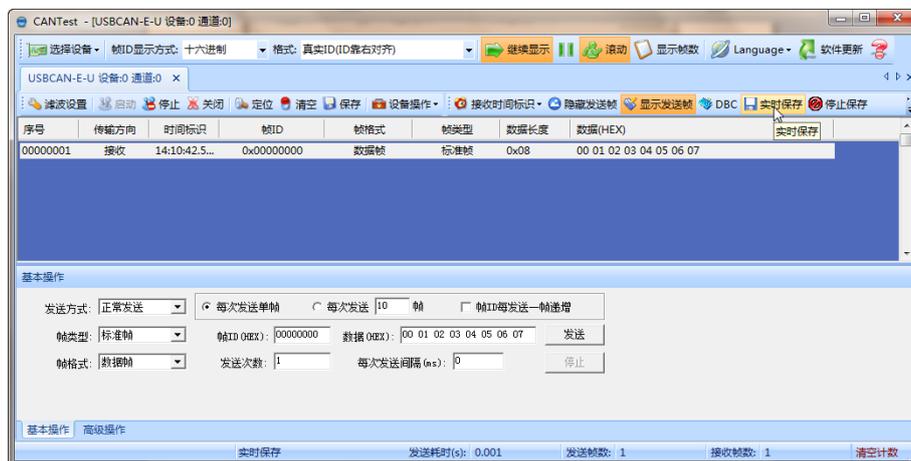


图 4.13 保存

4.2.5 总线利用率

点击【BusFlow】，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 4.14 所示。

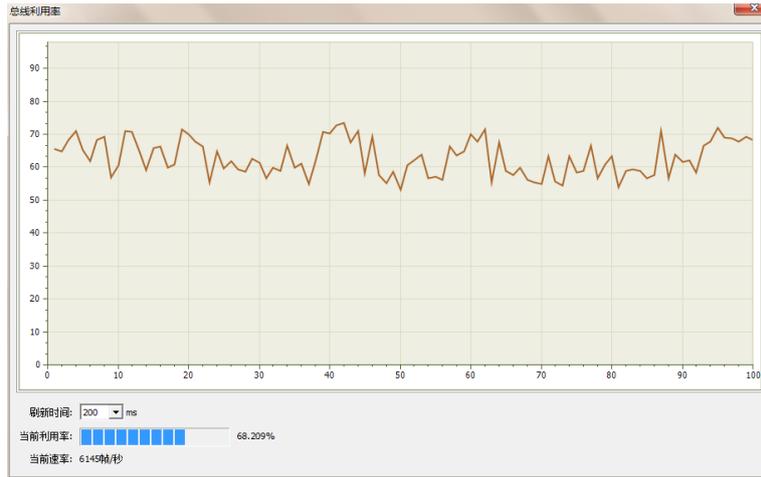


图 4.14 总线利用率

4.2.6 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 4.15 所示。

该窗口显示了错误信息的列表。窗口顶部有设置选项，包括最大显示行数（100000）、滚动开关、清空列表和导出报表按钮。列表包含以下列：序号、时间、错误码、消极错误代码类型、消极错误属性、消极错误表示、REC、TEC、仲裁错误表示。

序号	时间	错误码	消极错误代码类型	消极错误属性	消极错误表示	REC	TEC	仲裁错误表示
1991	23:43:51.000469	0x0000004:CAN控制器消极错误	0:位错误	0:发送错误	1 0 0 0 1:活动错误标志	0	136	
1992	23:43:51.000469	0x0000010:CAN控制器总线错误	0:位错误	0:发送错误	1 0 0 0 1:活动错误标志	0	136	
1993	23:43:51.000469	0x0000002:CAN控制器错误报警	0:位错误	0:发送错误	0 0 0 0 0	0	0	
1994	23:43:51.000469	0x0000004:CAN控制器消极错误	0:位错误	0:发送错误	0 0 0 1 1:帧开始	0	136	
1995	23:43:51.000469	0x0000010:CAN控制器总线错误	0:位错误	0:发送错误	0 0 0 1 1:帧开始	0	112	
1996	23:43:51.000469	0x0000020:总线关闭错误	0:位错误	0:发送错误	0 0 0 0 0	0	0	
1997	23:43:51.000469	0x0000010:CAN控制器总线错误	0:位错误	0:发送错误	0 0 0 1 1:帧开始	0	127	
1998	23:43:51.000469	0x0000010:CAN控制器总线错误	0:位错误	0:发送错误	0 0 0 1 1:帧开始	0	240	
1999	23:43:51.000469	0x0000010:CAN控制器总线错误	0:位错误	0:发送错误	0 0 0 1 1:帧开始	0	232	
2000	23:43:51.000469	0x0000010:CAN控制器总线错误	0:位错误	0:发送错误	0 0 0 1 1:帧开始	0	224	
2001	23:43:51.000469	0x0000010:CAN控制器总线错误	0:位错误	0:发送错误	0 0 0 1 1:帧开始	0	216	
2002	23:43:51.000469	0x0000010:CAN控制器总线错误	0:位错误	0:发送错误	0 0 0 1 1:帧开始	0	208	
2003	23:43:51.000469	0x0000010:CAN控制器总线错误	0:位错误	0:发送错误	0 0 0 1 1:帧开始	0	200	
2004	23:43:51.000469	0x0000010:CAN控制器总线错误	0:位错误	0:发送错误	0 0 0 1 1:帧开始	0	184	
2005	23:43:51.000469	0x0000010:CAN控制器总线错误	0:位错误	0:发送错误	0 0 0 1 1:帧开始	0	176	
2006	23:43:51.000469	0x0000010:CAN控制器总线错误	0:位错误	0:发送错误	0 0 0 1 1:帧开始	0	168	
2007	23:43:51.000469	0x0000010:CAN控制器总线错误	0:位错误	0:发送错误	0 0 0 1 1:帧开始	0	160	
2008	23:43:51.000469	0x0000020:CAN控制器错误报警	0:位错误	0:发送错误	0 0 0 0 0	0	0	
2009	23:43:51.000469	0x0000004:CAN控制器消极错误	0:位错误	0:发送错误	1 0 0 0 1:活动错误标志	0	136	
2010	23:43:51.000469	0x0000010:CAN控制器总线错误	0:位错误	0:发送错误	1 0 0 0 1:活动错误标志	0	136	
2011	23:43:51.000469	0x0000002:CAN控制器错误报警	0:位错误	0:发送错误	0 0 0 0 0	0	0	
2012	23:43:51.000469	0x0000004:CAN控制器消极错误	0:位错误	0:发送错误	0 0 0 1 1:帧开始	0	120	
2013	23:43:51.000469	0x0000010:CAN控制器总线错误	0:位错误	0:发送错误	0 0 0 1 1:帧开始	0	96	
2014	23:43:51.000469	0x0000020:总线关闭错误	0:位错误	0:发送错误	0 0 0 0 0	0	0	
2015	23:43:51.000469	0x0000010:CAN控制器总线错误	0:位错误	0:发送错误	0 0 0 1 1:帧开始	0	127	
2016	23:43:51.000469	0x0000010:CAN控制器总线错误	0:位错误	0:发送错误	0 0 0 1 1:帧开始	0	240	
2017	23:43:51.000469	0x0000010:CAN控制器总线错误	0:位错误	0:发送错误	0 0 0 1 1:帧开始	0	232	
2018	23:43:51.000469	0x0000010:CAN控制器总线错误	0:位错误	0:发送错误	0 0 0 1 1:帧开始	0	216	
2019	23:43:51.000469	0x0000010:CAN控制器总线错误	0:位错误	0:发送错误	0 0 0 1 1:帧开始	0	208	
2020	23:43:51.000469	0x0000010:CAN控制器总线错误	0:位错误	0:发送错误	0 0 0 1 1:帧开始	0	192	
2021	23:43:51.000469	0x0000010:CAN控制器总线错误	0:位错误	0:发送错误	0 0 0 1 1:帧开始	0	184	
2022	23:43:51.000469	0x0000010:CAN控制器总线错误	0:位错误	0:发送错误	0 0 0 1 1:帧开始	0	176	
2023	23:43:51.000469	0x0000010:CAN控制器总线错误	0:位错误	0:发送错误	0 0 0 1 1:帧开始	0	168	
2024	23:43:51.000576	0x0000004:CAN控制器消极错误	0:位错误	0:发送错误	0 0 1 1 0:ID20-ID18	0	127	

图 4.15 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 等，如图 5.1 所示。

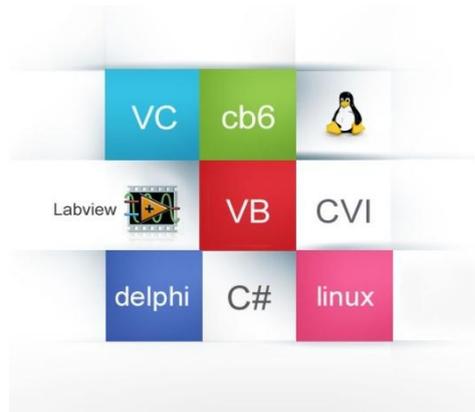


图 5.1 支持开发环境

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。

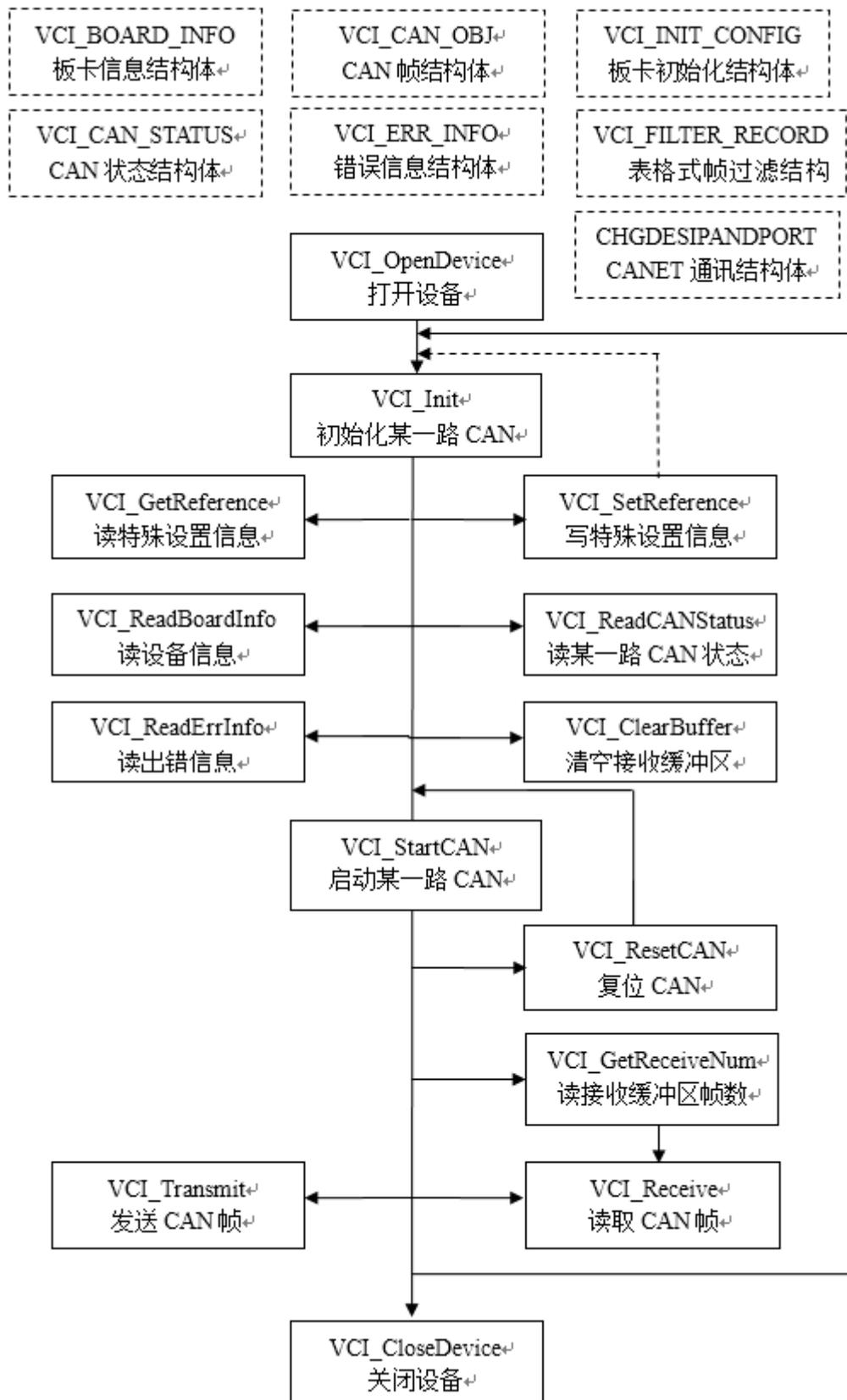
ByRef (可选)

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. 检查和维护

USBCAN-E-mini 接口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，请阅读下面的内容，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。

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	USB 端口电源+5V DC	使用电压表在电源输入端检查 USB 端口电压。
2	周围环境	检查周围环境温度（包括封闭环境的内部温度）	-25 ℃ ~ +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



7. 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向用户呈现详实、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 DB9 转 OBD 接口

A.1 功能简介

DB9-OBD 是 USBCAN-E-mini 的配件，一端连接 USBCAN-E-mini 的 Port 头（DB9 接口）上，另一端可以直接插到汽车的 OBD 接口处，方便用户使用 USBCAN-E-mini 对汽车进行诊断测试。



图 B.1.1DB9-OBD 外观示意图

A.2 技术参数

- ◆ 输入电压范围：60V；
- ◆ 输入电流范围：4A@40℃；
- ◆ 线间隔离电阻： $\geq 100\text{M}\Omega$ ；
- ◆ 温度范围：-20~+80℃；
- ◆ 插拔次数： ≥ 100 。

A.3 引脚信息

A.3.1 DB9 接头引脚排列

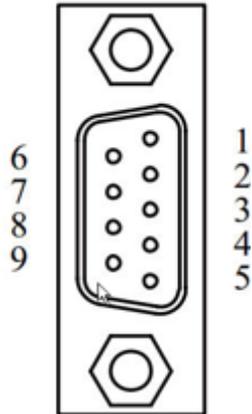


图 B.3.1 DB9 接头引脚排列

A.3.2 DB9 接头引脚描述

引脚序号	引脚名称	说明
1	N.C	未用
2	CAN_L	CAN_L 信号线
3	CAN_GND	参考地
4	N.C	未用
5	CAN_SHIELD	屏蔽线
6	CAN_GND	参考地
7	CAN_H	CAN_H 信号线
8	N.C	未用
9	N.C	未用

图 B.3.2 DB9 接头引脚描述

A.3.3 OBD 接头引脚排列

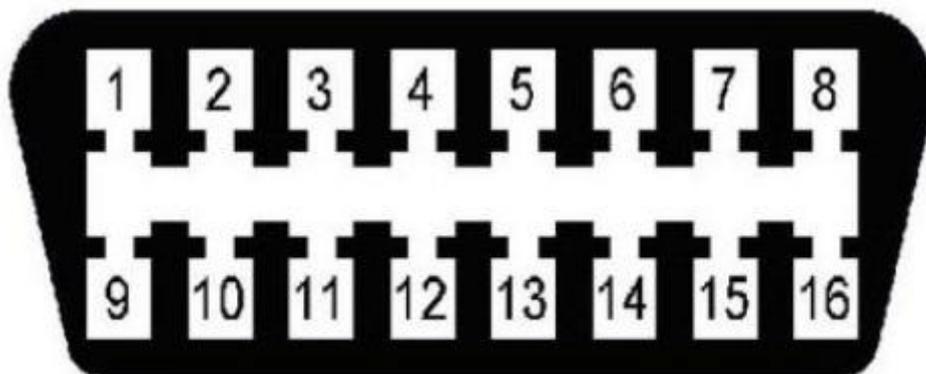


图 B.3.3 OBD 接头引脚排列

A.3.4 OBD 接头引脚描述

引脚序号	引脚名称	说明
5	GND	直连 DB9 的 6 管脚
12	CANH	--
13	CANL	--

注：其他管脚未用到，悬空。

图 B.3.3 OBD 接头引脚描述

A.4 机械尺寸

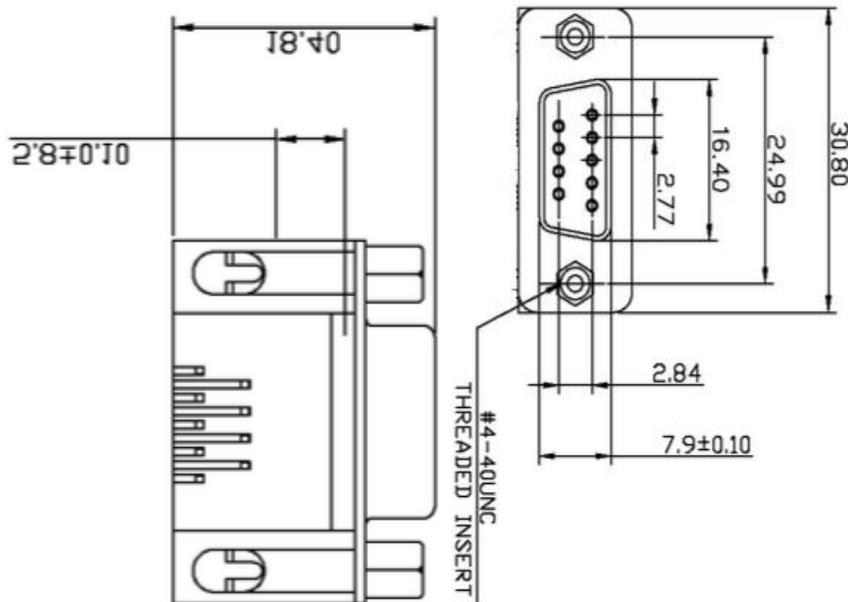


图 B.4.1 DB9 接头机械尺寸

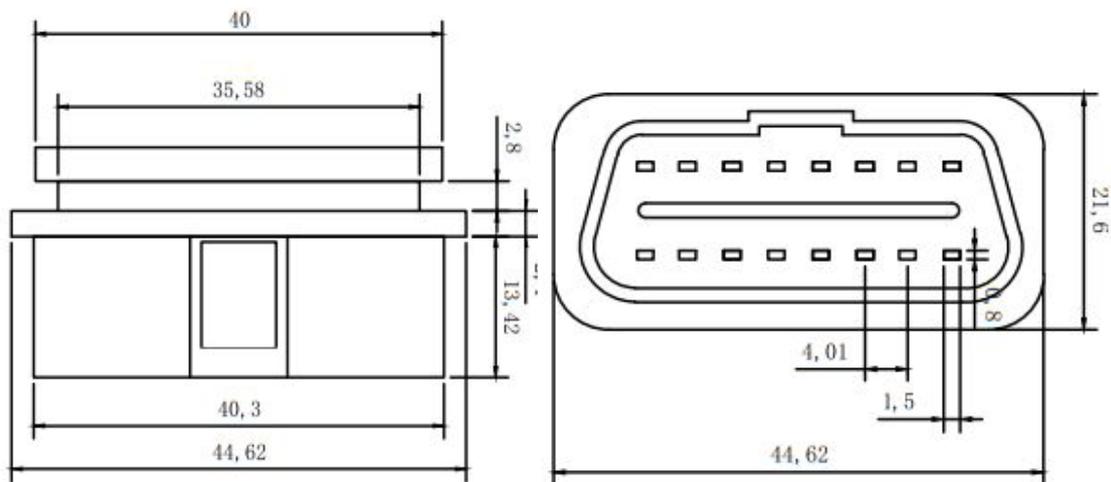


图 B.4.2 OBD 接头机械尺寸

附录B ARM 标准波特率

序号	Baudrate(Kbps)	晶振频率=36MHz
		自定义波特率寄存器（位定时）
1	5	0x003AC1C1
2	10*	0x003AC0E0
3	20*	0x004DC059
4	40	0x004DC02C
5	50*	0x003AC02C
6	80	0x003AC01B
7	100*	0x004DC011
8	125*	0x003AC011
9	200	0x004DC008
10	250*	0x003AC008
11	400	0x0039C005
12	500*	0x003CC003
13	666	0x003CC002
14	800*	0x0039C002
15	1000*	0x003CC001

注：带*号的是 CIA 协会推荐的波特率。