芜湖职业技术学院

毕 业 设 计

题 目 基于CAN总线汽车台架的节点分析

姓 名 邵一林 学 号 17120401312

系 信息工程 年 制 三

专 业 汽车智能技术 班 级 1

2020 年 5 月 12 日

芜湖职业技术学院

毕 业 设 计 任 务 书

2017 — 2020 学年

信息工程 系 汽车智能技术 专业

编 号 批准日期

学 生：邵一林 系 主 任：钱峰

Ⅰ设计题目： 基于 CAN 总线汽车台架的节点分析

Ⅱ原始资料： 目前实训平台是模拟大众帕萨特车型，将众多汽车操控设施真实的模拟到平台上，使用者可以直观的去体验，但是使用无法者很好的获取报文信息，了解各个节点报文的发送与接收过程，对 CAN 总线的通信不能直观的了解。

芜湖职业技术学院

毕 业 设 计 报 告 书

题目　基于CAN总线汽车台架的节点分析

信息工程 系

汽车智能技术 专业

1 班　学生 邵一林

系主任 钱峰　指导教师 钱峰

2020 年 5 月 12 日

目录

基于 CAN 总线汽车台架的节点分析

摘 要：随着汽车制造技术中不断应用高新技术，CAN 总线这个名词逐渐被人们所熟知，CAN 总线是对汽车中标准的串行传输系统的一种简缩习惯的叫法。CAN 的英文全称是：Controller Area Network，既控制器局域网络，属于工业现场总线的范畴，是现场总线的新一代局域通讯网络，又称为控制器局域网现场总线（CAN）。作为专门应用于产业自动化领域的网络。采用 CAN 总线技术，通过与实训平台的接口 CAN-H 和 CAN-L，然后通过 USB 转 CAN 模块接收，经上位机解析来获取各个节点的信息。上位机页面简洁大方，使用者可以更加直观的获取实训平台中节点发送过来的报文信息，报文的结果在 PC 机上显示出来。

关键词 CAN 总线 ； CAN 报文 ； 上位机 ； 节点

Node analysis based on CAN bus vehicle platform

Abstract: With the continuous application of high and new technology in automobile manufacturing technology, the term CAN bus is gradually known by people. The English full name of CAN is: Controller Area Network, which is not only the Controller local Area Network, belongs to the category of industrial field bus, is a new generation of local Area communication Network of field bus, also known as Controller LAN field bus (CAN).As a network specially applied in the field of industrial automation. It is adopts CAN bus technology, through the interface can-h and can-l with the training platform, and then through the USB to the CAN module to receive, through the upper computer analysis to obtain the information of each node. The upper computer page is simple and generous, and users can more intuitively obtain the message information sent by the nodes in the training platform, and the message results are displayed on the PC.

Keywords: CAN bus; CAN message; upper computer; node

第一章 绪论

1.1 前言

目前全球各大汽车制造商在上世纪 90 年代后期研发的汽车都采用了 CAN 总线或者部分零件具有 CAN 总线通信功能。总线网络技术对于全球的汽车产业有着重要的意义，已经成为了汽车现代化的标志。

对于中国来说，在 CAN/LIN 总线技术的研发道路上还需克服瓶颈。一是人才方面，车用总线技术涉及到机械领域和电子信息领域的融合，但我国的实际情况在这两个领域的协调中存在一些问题。从电子信息行业到车用总线人员，缺乏对汽车行业的理解，其研发出现的产品很难满足汽车的要求，即使满足汽车的要求，但不清楚汽车研发、生产和服务的流程，也很难实现生产化。另外，总线研发人员必须具有深厚的电子信息基础。因此，从汽车行业进入研发领域，由于基础的限制，很难研发出高性价的产品，二是要突破国外的技术封锁。目前，我国的技术研发还处于初步阶段，大量的技术参考文献都来自国外，还需要向国外学习。本次的设计旨在将汽车电子与信息技术结合起来，使得汽车实训台架可以与微机进行通信。由微机可以查看到台架的状态与故障情况，获取汽车的 CAN 总线报文以方便具体分析，加深对汽车结构与故障状况的了解。

1.2 发展前景与趋势

实训平台的界面是基于 Windows 的应用软件。其主要优点有操作简单，便于使用者很好的获取报文信息，了解各个节点报文的发送与接收过程。缺点是操控界面较小，使用者对 CAN 总线的通信不能直观的了解。

本设计是使用 Visual Studio 2017 为开发平台，界面简洁大方，当收到实训平台节点发送的报文后，经过 USB 转 CAN 模块接受后，经过电平转换成 USB 可识别的电平信号，在传送到上位机中，上位机解析后即可显示节点的报文。这使得使用者可以清晰的读取报文信息，提高了实训平台的使用效率。

第二章 设计任务书

设计一个 can 接口和一个上位机程序，能解决现在教学系统中无法直观监测数据报文，和实时发送控制命令的问题。

第三章 总体设计

总体系统由软件系统构成的上位机和硬件系统构成的 can 接口组成。

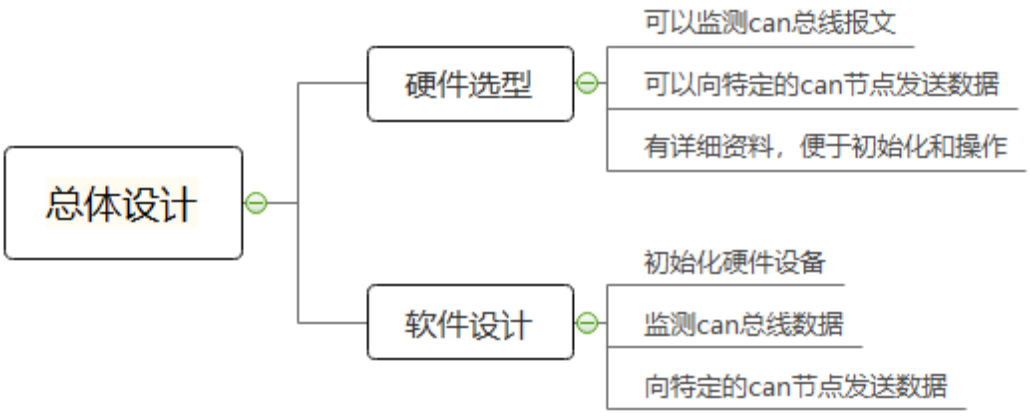

图 2 总体设计

硬件方面，需要实现全面监控整体 CAN 总线的数据帧。同时具有 CAN 协议转换功能，可以对数据进行双向的数模/模数转换，并且在速度和识别率等方面，要达到要求。

软件主要实现三方面功能。其一，要对硬件初始化设置通信的波特率。其二，规定数据帧长度，对接收到的数据进行分析，区分 ID 号和 DLC（数据帧长度）并根据数据帧长度读取数据。对于处理后的数据排列梳理，显示到窗口中。其三，要有向台架发送数据帧的功能。将已知功能的数据帧存入数组中，定义不同控件，以可视化方式向下位机发送数据。最好定义一个函数，作为发送使用，每个控件调用相同函数，减少代码冗余。

第四章 硬件系统

3.1 CAN 总线概述

CAN 即控制器局域网络。由于其高性能、高可靠性、及独特的设计，CAN 越来越受到人们的重视。国外已有许多大公司的产品采用了这一技术。

传统的电气系统大多采用点对点的单一通信方式，相互之间少有联系，这样必然造成庞大的布线系统。据统计一辆采用传统布线方法的高档汽车中，导线长度可达 2000 米，电气节点达 1500 个，而且该数字大约每十年增长 1 倍。这种传统布线方法不能适应汽车的发展。如图 1-1 所示

CAN 总线可有效减少线束长度，节省空间。例如某车针对车门、后视镜、摇窗机、门锁控制等功能的传统布线需要 20～30 根，应用总线 CAN 则只需要 2 根。无论从材料成本还是工作效率看，传统布线方法都将不能适应汽车的发展。下图分别为相同节点的传统点对点通讯方式和使用总线的通讯方式，从图 1-2 可以直观地比较线束的变化

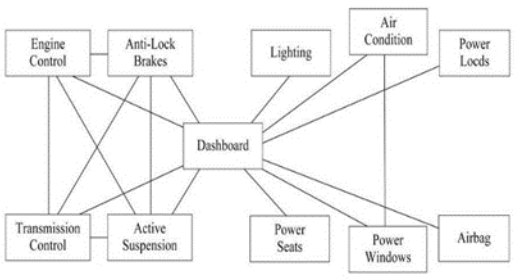

图 3-1 传统的节点通讯方式

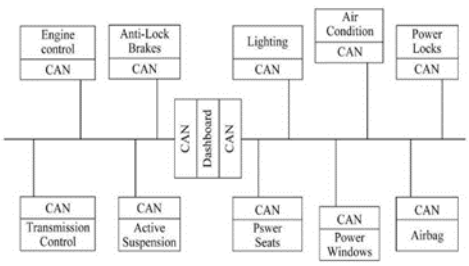

图 3-2 CAN 总线通讯方式

3.2 器件的选型 (Leaf Light V2)

3.2.1 模块概述

（1）USB 转 CAN 模块

此次实验首先是选用深圳维特智能科技有限公司的usb转can模块。模块是将 TTL 信号转换为 CAN 信号的模块。采用串口作为嵌入式系统的接口，数据 传输简单。

模块搭载一个 32 位的 STM32 处理芯片和一个 CAN 电平转换芯片。14 组屏蔽滤波器， 每组滤波器具有五种帧过滤方式。参数设置为 AT 指令设置方式，指令简洁精练，只有 6 条指令。更宽的波特率范围，网络适应性强。

完全支持 CAN 总线的 2.0A 及 2.0B 规范。采用 CortexM3 内核的处理器，数据处理能力更强，功耗更低，处理器集成了 CAN 控制器，让传输更省时。采样点自动调整为或接近 CIA 值。14 组 32 位屏蔽滤波器，随意选择，任意设定。 设置命令采用 AT 命令，设置更简单。 串口接口，操作更方便，缩短您的开发周期。 具有浪涌保护电路，抑制瞬态干扰，保护内部电路。 兼容 5V、3.3V 电源，可具有 TTL 接口的嵌入式系统对接。

（2）Kvaser Leaf Light V2

在后来又选用了Kvaser公司的产品Leaf Light v2，它是便捷的CAN通讯产品。让电脑通过USB和D-SUB9接口，与CAN总线网络连接在一起。新的Leaf Light v2为电脑与CAN总线网络之间的桥接，

提供卓越的可靠性与相宜的成本。 适合广泛的应用诸如，汽车，矿业，船舶，军工，石油和天然气的勘探，军事，工业和重型机械。

图 3-3 Leaf Light V2

Kvaser Leaf Light V2 提供时间戳精准度达到 100 微秒，无损的 CAN 报文传送与接收，并支持标准帧与扩展帧。

CAN 总线分析仪 Kvaser Leaf Light V2 主要特性：

- 全速 USB 界面，提供无损的 CAN 报文传送和接收
- 支持标准帧和扩展帧
- 8000 报文/秒的速率，时间戳精准度达到 100 微秒
- 电绝缘隔离（GalvanicIsolation），纳入为标配之一
- 有效防止电源浪涌或电击对产品造成的损坏
- 功耗只有 70 毫安（mA），适合手提电脑应用，减低对电池使用时间的影响

◆ 内置缓存，且通过高性能内核处理，减少电脑对处理时间关键任务的影响

◆ 提供免费的软件，免费的驱动固件更新和免费的技术支持

◆ 提供免费的 Kvaser CANlib 软件开发包，提供相应的编程例子

并且 Leaf Light v2 提供免费文档、软件和驱动程序。Kvaser CANLIB SDK 软件开发工具包免费提供，它包含针对 Kvaser CAN 接口开发软件所需要的所有资源。内含完整文档和用 C、C ++、C#、Delphi、Java、Python 和 VB 编写的大量程序示例。 所有 Kvaser CAN 接口卡使用同一软件 API 接口。针对某一款接口开发的程序无需变动适用于其他 Kvaser CAN 接口类型。

3.2.2 器件选择

在实验开始时，选择的是成本低廉的 USB 转 CAN 模块，经过查看资料后，利用串口读取并进行模块配置。设计了以下软件。并通过模拟实现了 CAN 通信（一对一）。

图 3-4 原实验设计的软件

但是在进行台架总线报文检测时，发现无法读取数据。与该公司技术人员咨询后，发现该模块只能作为 can 节点的模拟使用，无法监测该台架总线上的所有报文信息。

后来经过寻找发现了 Leaf Light v2 设备，可以让电脑通过 USB 接口直接和 D-SUB9 接口相连，与 CAN 总线网络连接在一起。通过实验，发现它可以实现数据的监视，并且向设备发送数据，完全符合需求。所以用其作为硬件设备

3.3　汽车 CAN 总线系统综合实训考核装置

本实训平台由控制装置、实训桌、以及各个模块组成件构成。它是大众帕萨特车型，将众多汽车操控设施真实的模拟到平台上，使用者可以直观的去体验。而且实训平台各个节点的接口都已经清晰明了的显示出来。通过连接 CAN-H 和 CAN-L 两个接口，经过解析即可获取节点的报文信息，如图 3-5 所示。

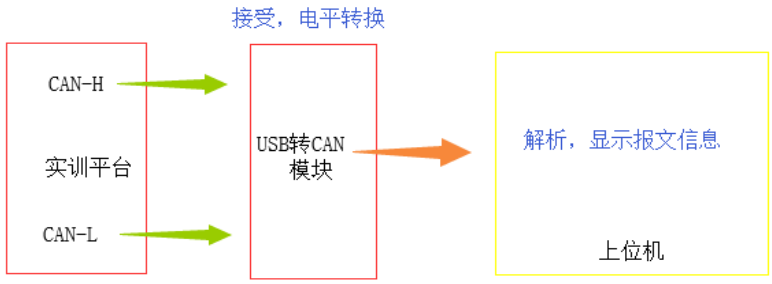

图 3-5 节点的解析流程图

第五章 系统软件设计

系统硬件构成简单，结合硬件工作特性，主要围绕软件设计方面进行主要的开发工作。

4.1 报文节点的概念以及获取

在总线中传送的报文，每帧由 7 部分组成。CAN 协议支持两种报文格式，其唯一的不同是标识符（ID）长度不同，标准格式为 11 位，扩展格式为 29 位。

当使用者触发实训平台的某个节点后，该节点就像 CAN 总线上发送报文信息，USB 转 CAN 模块与实训平台的 CAN-H 和 CAN-L 连接后接收报文信息，经过电平转换发送给上位机，上位机读取并解析报文内容，即可识别是哪个几点发送过来的。

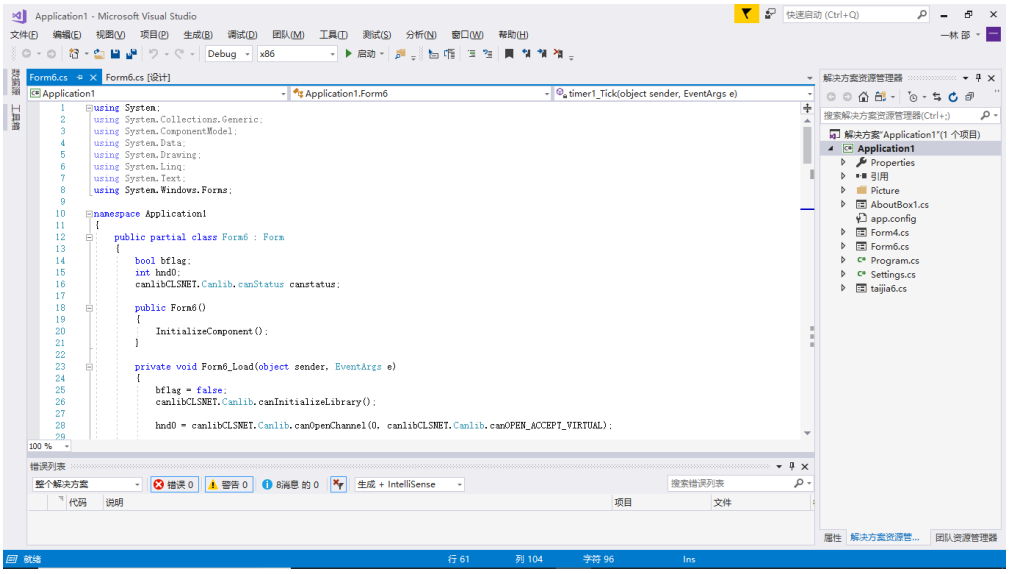

图 4-1 开发环境的配置

本设计的上位机是在 Microsoft Visual Studio 2017 上开发的，报文接收以后经过C#程序语言编写并提供的一个托管的运行时环境，使程序比以往更加稳定、安全，能够快速地解析并显示报文信息。如图所示（图 4-1 为开发环境的配置，图 4-2 为主程序流程图。）

4.2 程序流程图

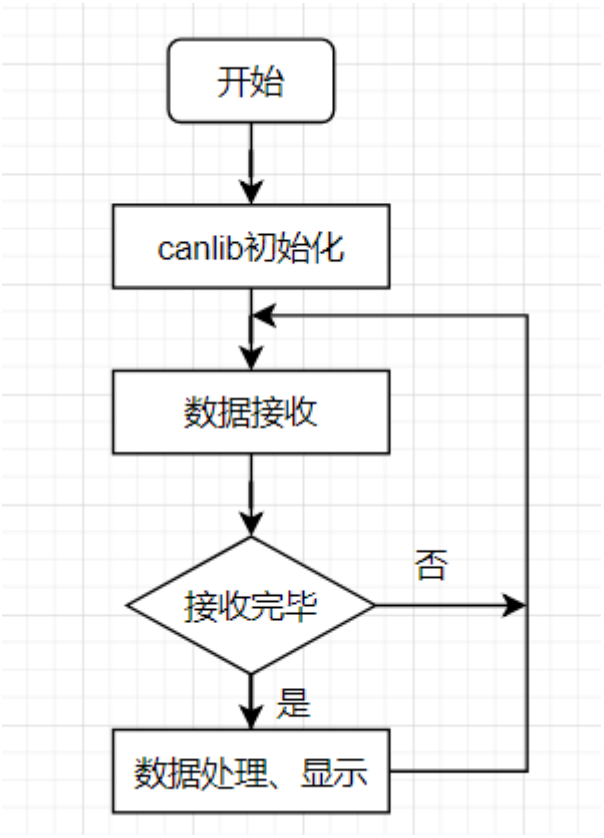

图 4-2-1 数据接收程序流程图

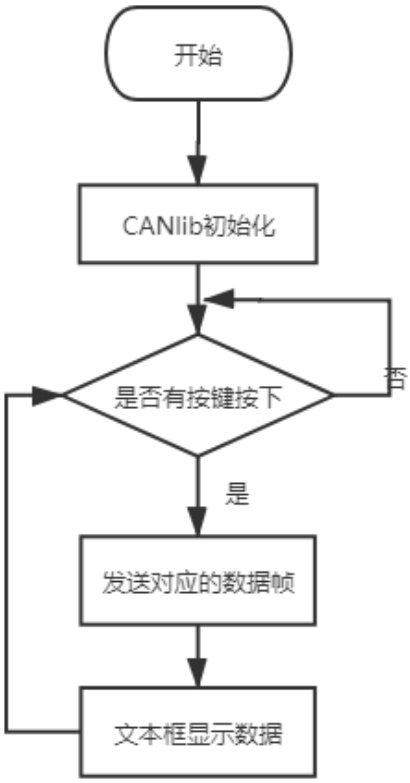

图 4-2-2 数据发送程序流程图

4.3 仿真软件与算法介绍

本设计仿真软件主要包括以下三项功能：

1、 发送模拟汽车 CAN 报文至 CAN 总线当中。

2、 从 CAN 总线当中接收汽车 CAN 报文。

3、 提供人机界面。

根据上述需求，该仿真软件系统包括 2 个功能程序：

2、数据接收子程序——利用来接收模拟的汽车 CAN 报文数据以列表的方式显示，其中列表当中有序号，时间，帧类型，帧长度，帧 ID，帧数据。并且可以实时开始、暂停。

3、数据发送子程序——将要台架做出的反应的报文数据写进各个按钮控件中，并按照报文格式显示在上方的显示栏中。按动按钮即可发送 can 报文数据给对应的汽车节点。

以上 2 个程序主要是在顺序结构，条件结构以及 While 结构程序中运行。主控程序随软件启动运行，它可以自动设置波特率以及帧类型，同时可以实时调度其余 2 个程序的运行，并将各子程序的运行结果传递给相关功能面板的显示控件以反馈给用户，子程序功能就是实现汽车 CAN 报文数据的发送和接收以及列表显示、存储。当主控程序正常启动接口卡并使其工作后，数据接收子程序才能运行，数据列表与保存子程序，必须在前两者正常工作后，才能根据用户要求执行相关功能。

CAN 报文数据的发送和接收是系统监测功能的基础，而软件本身需要依靠接口卡从 CAN 总线接收数据，因此，要实现所有的功能，

必须首先正确启动 Leaf Light v2 通讯转换模块，初始化其相关运行参数，使其可以正常运行。Leaf Light v2 转换模块智能 CAN 接口卡提供了大量的 VCI 函数库(应用程序接口)，这些函数可以从接口卡的驱动文件 canlibCLSNET. dll 中导出，并可以在 C#中调用，实现对接口卡的启动、设置等相关操作。

4.4 报文窗口的显示

面板设计主要有两大页面：

1，数据流查看。2，CAN-BUS 实训台模拟控制。

图 4-3 最终程序运行窗口（模拟控制）

图 4-3 最终程序运行窗口（报文接收）

1、数据流查看

用来将接收到的报文在窗口中显示出来，有开始、暂停、清空三个按钮操作。

2、CAN-BUS 实训台模拟控制

实训模拟台控制，通过上位机软件向 can 总线发送数据帧。控制台架上的相关组件运行

第六章 系统调试运行

将 CAN-BUS 实训台架与 Kvaser Leaf Light V2 CAN 接口连接，Kvaser Leaf Light V2 模块与 PC 上位机正常通讯后，点击开始按钮，即可看到台架向上位机输出的数据（每个节点的 CAN 报文）。由于台架对于上位机发送的数据是循环发送，每个节点都发送数据。所以各个 CAN 节点具体数据需要分析。

通过菜单栏打开 CAN-BUS 实训台模拟控制界面。点击各个模块按钮，由 PC 端向台架发送数据，可以看到台架相应模块做出指定的动作。同时上方显示发送的报文具体格式。

由于上位主机所接插的 Leaf Light V2 一般需要驱动，所以在上位机软件的编制过程中，实现与 CAN 总线的通讯部分可以直接调用相应的函数，如上位主机与 CAN 通讯的主要任务：对 Leaf Light V2 的初始化、CAN 信息包的发送、CAN 信息包的接收等，都有现成的函数可以使用，为用户使用 CAN 进行通讯提供了方便。实现 CAN 信息包的发送，首先要确定信息包的 11 位信息标识符，填进帧头，并在数据域中填进需要发送的数据信息，通过发送函数发送给所有 CAN 节点或特定的 CAN 节点上。而对于使用接收函数所接收的 CAN 信息包，亦通过其 11 位信息标识符，判定其来源，对数据域的数据进行处理，取得有效的信息，进行显示或存储。作为使用者只需要定义变量作为数据的出入口即可。

第七章 存在的问题

在实现本设计中，存在着几个问题，首先原来选择的硬件装置无法完成需要的操作，没有相应的功能。经过长时间的调整，查找许多的资料，向原厂家技术人员资讯后才发现无法达到想要的功能，需要更换。又花费大量时间才找到适合的硬件设备。

在调试完软件以后发现读写的数据散乱，无法及时找到需要的数据，也无法将所有的数据与各个单元对应起来。想了很多办法，发现台架的资料太少，无法将数据与单元对应起来，单元之间并不是依次循环发送，有些单元好几个周期才发送一次数据，统一操作的发送与接收的数据页不同，无法监控到。可能是厂家的加秘密手段，需要更详细的数据资料才能将软件完善下去。

致谢

随着毕业日子的到来，毕业设计也接近了尾声。经过几周的奋战我的毕业设计终于完成了。在没有做毕业设计以前觉得毕业设计只是对这几年来所学知识的单纯总结，但是通过这次做毕业设计发现自己的看法有点太片面。毕业设计不仅是对前面所学知识的一种检验，而且也是对自己能力的一种提高。通过这次毕业设计使我明白了自己原来知识还比较欠缺。自己要学习的东西还太多，以前老是觉得自己什么东西都会，什么东西都懂，有点眼高手低。通过这次毕业设计，我才明白学习是一个长期积累的过程，在以后的工作、生活中都应该不断的学习，努力提高自己知识和综合素质。

在这次毕业设计中也使我们的同学关系更进一步了，同学之间互相帮助，有什么不懂的大家在一起商量，听听不同的看法对我们更好的理解知识，所以在这里非常感谢帮助我的同学。

总之，不管学会的还是学不会的，的确觉得困难比较多。真是万事开头难，不知道如何入手。最后终于做完了有种如释重负的感觉。此外，还得出一个结论：知识必须通过应用才能实现其价值！有些东西以为学会了，但真正到用的时候才发现是两回事，所以我认为只有到真正会用的时候才是真的学会了。

在此要感谢我的指导老师钱峰老师对我悉心的指导，感谢老师给我的帮助。在设计过程中，我通过查阅大量有关资料，与同学交流经验和自学，并向老师请教等方式，使自己学到了不少知识，也经历了

不少艰辛，但收获同样巨大。在整个设计中我懂得了许多东西，也培养了我独立工作的能力，树立了对自己工作能力的信心，相信会对今后的学习工作生活有非常重要的影响。而且大大提高了动手的能力，使我充分体会到了在创造过程中探索的艰难和成功时的喜悦。虽然这个设计做的也不太好，但是在设计过程中所学到的东西是这次毕业设计的最大收获和财富，使我终身受益。

参考文献

[1] Kvaser 公司. Kvaser CanKing 实操说明文档. 2017

[2] Kvaser 公司.kvaser_leaf_light_v2_usersguide. 2017

[3] 郭力子. 《Visual C#》[M]. 北京：机械工业出版社，2009

[4] 朱双华.汽车 CAN 系统故障诊断与检测技术.长沙:国防科技大学出版社，2008.

[5] 邬宽明.CAN 总线原理和应用系统设计.北京：北京航空航天大学出版社，2002.

[6] 杜善封.CAN 总线测控技术及应用[M].电子工业出版社.2007

[7] 李勇，张怡.CAN 总线的设计与实现[M].西北工业大学电子系.2000

[8] 饶运涛. 《现场总线 CAN 原理与应用技术》[M]. 北京:北京航空航天大学出版社, 2003

附录

源程序代码：

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.Text.RegularExpressions;

namespace Application1
{
    public partial class Form4 : Form
    {
        string strdata;
        string output;

        canlibCLSNET.Canlib.canStatus canstatus;
        int hnd0;

        int flag, DLC;
        int ID;

        public bool CanErrorFlag = false;

        byte[] msg = new byte[8];

        //第一位ID, 第二位 DLC, 第三位--第11位   DATA[0]-DATA[7]
        string[] strarr ={
                                "1281,2,16,0,0,0,0,0,0,0,0",
                                "1281,2,32,0,0,0,0,0,0,0,0",
                                "1281,2,64,0,0,0,0,0,0,0,0",
                                "1281,2,128,0,0,0,0,0,0,0,0",
                                "385,3,0,1,0,0,0,0,0,0,0",
                                "385,3,0,4,0,0,0,0,0,0,0",
                                "385,3,0,64,0,0,0,0,0,0,0",
                                "385,3,0,16,0,0,0,0,0,0,0",
                                "641,2,0,2,0,0,0,0,0,0,0",
                                "641,2,0,1,0,0,0,0,0,0,0",
```

```
                                "385,3,64,0,0,0,0,0,0,0,0",
                                "385,3,16,0,0,0,0,0,0,0,0"
                               };
        public Form4()
        {
            InitializeComponent();
        }

        //初始化canbus
        private void can_Init()
        {
            canlibCLSNET.Canlib.canInitializeLibrary();

            hnd0 = canlibCLSNET.Canlib.canOpenChannel(0,
canlibCLSNET.Canlib.canOPEN_ACCEPT_VIRTUAL);

            canlibCLSNET.Canlib.canSetBusParams(hnd0,
canlibCLSNET.Canlib.canBITRATE_100K, 0, 0, 0, 0, 0);

            canlibCLSNET.Canlib.canBusOn(hnd0);

        }
        private void Form4_Load(object sender, EventArgs e)
        {
            can_Init();

        }

        //发送数据到控制台
        public void CanDataSend(int td, Int32 ID, byte[] data, Int32 DLC)
        {
            int flag = 0;
            canstatus = canlibCLSNET.Canlib.canWrite(td, ID, data, DLC, flag);
            //在发送区，显示要发送的数据信息
            ID2.Text = ID.ToString();
            DLC2.Text = DLC.ToString();
            //发送D0显示.Text = (DLC >= 1 ？ 数据[0].ToString() : "");
            D02.Text = (DLC >= 1 ? data[0].ToString() : "");
            D12.Text = (DLC >= 2 ? data[1].ToString() : "");
            D22.Text = (DLC >= 3 ? data[2].ToString() : "");
            D32.Text = (DLC >= 4 ? data[3].ToString() : "");
            D42.Text = (DLC >= 5 ? data[4].ToString() : "");
            D52.Text = (DLC >= 6 ? data[5].ToString() : "");
            D62.Text = (DLC >= 7 ? data[6].ToString() : "");
```

```
            D72.Text = (DLC >= 8 ? data[7].ToString() : "");

        }
        //
        public string PrintMessage(int ID, byte[] msg, int DLC, int flag, long time11)
        {
            string result = "";
            result = ID.ToString() + " : " + DLC.ToString();
            for (int i = 0; i < DLC; i++)
            {
                result += " : " + msg[i];
            }
            return result;
        }

        //Can数据接收
        public void CanReadData(int td, out int ID, byte[] msg, out int DLC, out int flag, out
long time11)
        {
            string GetData = "";

            canstatus = canlibCLSNET.Canlib.canRead(td, out ID, msg, out DLC, out flag, out
time11);
            if (canstatus == canlibCLSNET.Canlib.canStatus.canOK)
            {
                if ((flag & canlibCLSNET.Canlib.canMSG_ERROR_FRAME) != 0)
                {

                }
            }

        }

        private void LockClose_Click(object sender, EventArgs e)
        {
            ID = 641;
            DLC = 2;
            byte[] msg = { 0, 1, 0, 0, 0, 0, 0, 0 };

            CanDataSend(hnd0, ID, msg, DLC);

        }
```

```
private void Up2_Click(object sender, EventArgs e)
{
    ID = 1281;
    DLC = 2;
    byte[] msg = { 16, 0, 0, 0, 0, 0, 0, 0 };

    CanDataSend(hnd0, ID, msg, DLC);
}

private void Down2_Click(object sender, EventArgs e)
{
    ID = 1281;
    DLC = 2;
    byte[] msg = { 32, 0, 0, 0, 0, 0, 0, 0 };

    CanDataSend(hnd0, ID, msg, DLC);
}

private void Left2_Click(object sender, EventArgs e)
{
    ID = 1281;
    DLC = 2;
    byte[] msg = { 64, 0, 0, 0, 0, 0, 0, 0 };

    CanDataSend(hnd0, ID, msg, DLC);
}

private void Right2_Click(object sender, EventArgs e)
{
    ID = 1281;     //id=641    dlc=2    d0=128,门锁动作
    DLC = 2;
    byte[] msg = { 128, 0, 0, 0, 0, 0, 0, 0 };

    CanDataSend(hnd0, ID, msg, DLC);
}

private void timer1_Tick(object sender, EventArgs e)
{
    flag = 0;
    long time11;
    CanReadData(hnd0, out ID, msg, out DLC, out flag, out time11);
}

private void LockOpen_Click(object sender, EventArgs e)
```

```
{
    ID = 641;
    DLC = 2;
    byte[] msg = { 0, 2, 0, 0, 0, 0, 0, 0 };

    CanDataSend(hnd0, ID, msg, DLC);
}

private void Shunsz_Click(object sender, EventArgs e)
{
    ID = 385;
    DLC = 2;
    byte[] msg = { 64, 0, 0, 0, 0, 0, 0, 0 };

    CanDataSend(hnd0, ID, msg, DLC);
}

private void Nisz_Click(object sender, EventArgs e)
{
    ID = 385;
    DLC = 2;
    byte[] msg = { 16, 0, 0, 0, 0, 0, 0, 0 };

    CanDataSend(hnd0, ID, msg, DLC);
}

private void Shunsz2_Click(object sender, EventArgs e)
{
    ID = 385;
    DLC = 2;
    byte[] msg = { 0, 4, 0, 0, 0, 0, 0, 0 };

    CanDataSend(hnd0, ID, msg, DLC);
    CanDataSend(hnd0, ID, msg, DLC);
}
private void Nisz2_Click(object sender, EventArgs e)
{
    ID = 385;
    DLC = 2;
    byte[] msg = { 0, 1, 0, 0, 0, 0, 0, 0 };

    CanDataSend(hnd0, ID, msg, DLC);
    CanDataSend(hnd0, ID, msg, DLC);
}
```

```
        private void Shunsz3_Click(object sender, EventArgs e)
        {
            ID = 385;
            DLC = 2;
            byte[] msg = { 0, 64, 0, 0, 0, 0, 0, 0 };

            CanDataSend(hnd0, ID, msg, DLC);
        }

        private void button1_Click(object sender, EventArgs e)
        {
            this.Close();
        }

        private void Nisz3_Click(object sender, EventArgs e)
        {
            ID = 385;
            DLC = 2;
            byte[] msg = { 0, 16, 0, 0, 0, 0, 0, 0 };
            CanDataSend(hnd0, ID, msg, DLC);
        }
        private void groupBox3_Enter(object sender, EventArgs e)
        {
        }
    }
}
//报文显示窗口代码
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;

namespace Application1
{
    public partial class Form6 : Form
    {
        bool bflag;
        int hnd0;
        canlibCLSNET.Canlib.canStatus canstatus;
```

```
        public Form6()
        {
            InitializeComponent();
        }

        private void Form6_Load(object sender, EventArgs e)
        {
            bflag = false;
            canlibCLSNET.Canlib.canInitializeLibrary();

            hnd0 = canlibCLSNET.Canlib.canOpenChannel(0,
canlibCLSNET.Canlib.canOPEN_ACCEPT_VIRTUAL);

            canlibCLSNET.Canlib.canSetBusParams(hnd0,
canlibCLSNET.Canlib.canBITRATE_100K, 0, 0, 0, 0, 0);

            canlibCLSNET.Canlib.canBusOn(hnd0);
        }

        private void timer1_Tick(object sender, EventArgs e)
        {
            int    ID, DLC, flag;
            byte[] msg=new byte[30];
            long time11;

           // while (bflag)
             if(bflag)
             {
                 string str2 = textBox1.Text;
                 string getData = "";
                 string str1 = "";
                 string str11, str12;
                 canstatus = canlibCLSNET.Canlib.canRead(hnd0, out ID, msg, out DLC, out
flag, out time11);
                 if (canstatus == canlibCLSNET.Canlib.canStatus.canOK)
                 {
                     switch (ID )
                     {
                         case 1281 :
                             {
                                 toolStripStatusLabel1.Text = "";
                                 switch (msg[1].ToString())
                                 {
```

```
                                    case "16": toolStripStatusLabel1.Text = "主驾驶室
后视镜上"; break;
                                    case "32": toolStripStatusLabel1.Text = "主驾驶室
后视镜下"; break;
                                    case "64": toolStripStatusLabel1.Text = "主驾驶室
后视镜左"; break;
                                    case "128": toolStripStatusLabel1.Text = "主驾驶室
后视镜右"; break;
                                    default: break;
                                }
                                switch (msg[2].ToString())
                                {
                                    case "16": toolStripStatusLabel1.Text = "副驾驶室
后视镜上"; break;
                                    case "32": toolStripStatusLabel1.Text = "副驾驶室
后视镜下"; break;
                                    case "64": toolStripStatusLabel1.Text = "副驾驶室
后视镜左"; break;
                                    case "128": toolStripStatusLabel1.Text = "副驾驶室
后视镜右"; break;
                                    default : break;
                                }
                                break;
                            }
                        case 385:
                            {
                                toolStripStatusLabel1.Text = "";
                                switch (msg[2].ToString())
                                {
                                    case "1": toolStripStatusLabel1.Text = "左后侧车窗
逆时针"; break;
                                    case "4": toolStripStatusLabel1.Text = "左后侧车窗
顺时针"; break;
                                    case "16": toolStripStatusLabel1.Text = "右后侧车
窗逆时针"; break;
                                    case "64": toolStripStatusLabel1.Text = "右后侧车
窗顺时针"; break;
                                    case "0": if(msg[1] == 16)
                                                toolStripStatusLabel1.Text = "副驾驶
侧车窗逆时针";
                                            else if(msg[1] == 64)
                                                toolStripStatusLabel1.Text = "副驾驶
侧车窗顺时针";
                                            break;
```

```
                    default: break;
                }
                break;
            }
        case 641:
            {
                toolStripStatusLabel1.Text = "";
                switch (msg[2].ToString())
                {
                    case "1": toolStripStatusLabel1.Text = "车门锁关";
break;
                    case "2": toolStripStatusLabel1.Text = "车门锁开";
break;
                    default: break;

                }
                break;
            }
        default: break;
    }
    str11 = ID.ToString();
    if (str11.Length == 1)
    {
        str11 += "            ";
    }
    else if (str11.Length == 2)
    {
        str11 += "           ";
    }
    else if (str11.Length == 3)
    {
        str11 += "           ";
    }
    else if (str11.Length == 4)
    {
        str11 += "         ";
    }

    getData = str11+ DLC.ToString() + "           ";
    for (int i = 0; i < 8; i++)
    {
        str12 = msg[i].ToString();
        if (str12.Length == 1)
        {
```

```
                    str12 +=  "      ";
                }
                else if (str12.Length == 2)
                {
                    str12 +=  "    ";
                }
                else if (str12.Length == 3)
                {
                    str12 +=  " ";
                }
                str1+=str12;
            }
            getData += str1+"\r\n";


            //listBox1.Items.Add(getData);
        }
        textBox1.Text = getData + str2;
    }

}


//start
private void button1_Click(object sender, EventArgs e)
{
    bflag = true;
   // timer1.Start();
}

private void button2_Click(object sender, EventArgs e)
{
   // timer1.Stop();
    bflag = false;
}

private void button4_Click(object sender, EventArgs e)
{
    textBox1.Text = "";
}

private void  打开控制台ToolStripMenuItem_Click(object sender, EventArgs e)
{
    Form4 frm = new Form4();
```

```
            frm.ShowDialog();

        }

        private void 清空文本ToolStripMenuItem_Click(object sender, EventArgs e)
        {
            textBox1.Text = "";
        }

        private void 关闭QToolStripMenuItem_Click(object sender, EventArgs e)
        {
            this.Close();
        }

        private void 关于ToolStripMenuItem_Click(object sender, EventArgs e)
        {
            AboutBox1 about = new AboutBox1();
            about.ShowDialog();
        }

        private void 作者ToolStripMenuItem_Click(object sender, EventArgs e)
        {
            System.Diagnostics.Process.Start("https://space.bilibili.com/315241796");
        }
    }
}
```