

# 车辆通用故障诊断协议的研究与开发

刘丽丽<sup>1,2</sup>, 徐皓冬<sup>1,2</sup>, 宋 岩<sup>1,2</sup>, 周 亚<sup>1,2</sup>

(1. 中国科学院研究生院, 北京 100039; 2. 中国科学院沈阳自动化研究所, 沈阳 110016)

**摘 要:** 介绍通用诊断协议 ISO14229 和基于 CAN 总线的车辆故障诊断协议 ISO15765, 分析两者之间的关系及 ISO15765 的体系结构, 在此基础上给出 ISO15765 协议栈总体设计, 并描述协议每一层的软件实现。协议栈设计利用分层接口方式, 下层向上层提供服务, 协议内部功能采用模块化设计。实验结果表明, 该协议栈能满足 ISO15765 规定的标准和一般的故障诊断功能需求。

**关键词:** ISO15765 协议; ISO14229 协议; 故障诊断; CAN 总线; ISO11898 协议

## Research and Development of Vehicle Unified Fault Diagnostic Protocol

LIU Li-li<sup>1,2</sup>, XU Ai-dong<sup>1,2</sup>, SONG Yan<sup>1,2</sup>, ZHOU Ya<sup>1,2</sup>

(1. Graduate University of Chinese Academy of Sciences, Beijing 100039, China;

2. Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang 110016, China)

**【Abstract】** ISO14299 specifying unified diagnostic services and CAN based vehicle fault diagnostic protocol ISO15765 are introduced, and the relationship between ISO14229 and ISO15765 and the architecture of ISO15765 are analyzed. The overall design of the ISO15765 protocol stack is provided in the paper. The design of the ISO15765 protocol stack uses layered and interface method. Lower layers provides services to higher layers. Modular design is adopted for developing internal operation. Experimental results show that the protocol stack satisfies ISO15765 standard and general fault diagnosis requirements.

**【Key words】** ISO15765 protocol; ISO14229 protocol; fault diagnosis; CAN bus; ISO11898 protocol

DOI: 10.3969/j.issn.1000-3428.2012.16.003

### 1 概述

为了满足客户对车辆的可靠性、安全性、舒适性和便利性等性能方面的要求, 车辆内部控制器、传感器和执行器等电子器件明显增多。这给车辆系统带来了 2 个重大问题: 车内线束增多和故障诊断难度增加。为了减少车身线束及提高各电控单元之间的通信速度, 采用串行总线来连接器件。

CAN 总线是当今自动化领域中最具应用前景的技术之一, 可提供高达 1 Mb/s 的数据传送速率, 在可靠性、实时性和灵活性方面优势明显, 适合测控单元之间的互连。现阶段, CAN 总线在汽车电子系统中得到广泛的应用, 代表汽车电子控制网络的主流发展趋势<sup>[1]</sup>, 基于 CAN 总线的故障诊断系统将越来越受关注。针对诊断设备和车辆内部电子控制单元(ECU)之间的数据交换, 各个汽车公司几乎都制定了相关的标准, 导致各个汽车公司之间诊断仪的不通用性。

ISO14229<sup>[2]</sup>是为诊断系统定义通用的需求而发布的,

与串行数据链路的种类无关。ISO14229 定义了通用诊断服务(UDS), 这些诊断服务允许诊断仪控制车辆内部 ECU 内的诊断功能。ISO14229 只定义了应用层的服务。

ISO15765<sup>[3]</sup>是一种基于 CAN 总线的诊断协议, 其实质就是在 CAN 总线上实现 UDS 即 ISO14229, 该协议满足了 E-OBD 的系统要求, 符合现代网络总线系统的发展趋势, 将成为未来车辆行业的通用诊断标准。我国的汽车诊断还是比较落后的<sup>[4]</sup>, 只有充分理解和掌握诊断协议 ISO14229 和 ISO15765, 才能有助于我国车辆诊断技术的发展。本文主要研究和开发了 ISO15765 协议栈, 并对本文所设计的协议栈进行了测试。

### 2 ISO15765 体系结构

ISO15765 是基于 CAN 总线、用于车辆故障诊断的高层协议<sup>[5]</sup>, 在 CAN 总线上实现了 UDS(ISO14229), 其体系结构如图 1 所示。ISO15765 协议相对于 OSI 参考模型而言, 定义了 4 层, 分别是应用层、网络层、数据链路层和物理层。

**基金项目:** 国家科技支撑计划基金资助项目(2009BADB5B01)

**作者简介:** 刘丽丽(19987-), 女, 硕士研究生, 主研方向: 车辆故障诊断, 工业通信; 徐皓冬, 研究员; 宋 岩, 助理研究员; 周 亚, 硕士研究生

**收稿日期:** 2011-09-30 **修回日期:** 2011-12-07 **E-mail:** liulili@sia.cn

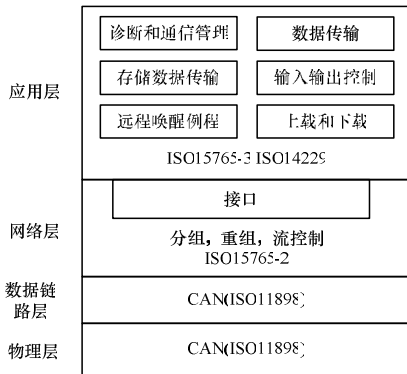


图1 ISO15765 体系结构

数据链路层和物理层是由 CAN 国际标准 ISO11898<sup>[6]</sup> 所描述、具有非破坏性的网络仲裁机制，其协议数据单元是 CAN 数据帧，包含的最大数据长度为 8 Byte。物理层的物理介质是 CAN 总线，具有较高的通信速率、传输距离较远、抗电磁干扰能力强等优点。

网络层主要是为应用层提供接口，并为长报文(数据长度大于 8 Byte)的传输提供了分组和重组的功能，由 ISO15765-2<sup>[7]</sup> 文档所描述。

应用层是由 ISO14229 和 ISO15765-3<sup>[8]</sup> 所描述的，主要是在 CAN 总线上实现 UDS，ISO14229 主要定义以下 6 个方面有关诊断的通用服务: 诊断和通信管理功能单元，数据传输服务，储存数据传输功能，输入输出控制功能单元，远程唤醒例程功能单元和上载与下载功能单元。诊断和通信管理功能单元主要是定义有关诊断会话控制、通信参数设定的服务，保证诊断通信过程中的正常工作；数据传输服务单元主要是对整车的参数进行实时监控，有助于故障的诊断；存储数据的传输功能是读取或清除 ECU 内部故障代码，主要用于故障的诊断；输入输出控制功能单元主要是对车辆 ECU 的输入输出进行控制，也有助于故障的诊断；远程唤醒例程功能单元主要是可唤醒 ECU 内部有关诊断的例程，通过对这些例程的启动可有助于诊断的判断；上载与下载功能单元主要是在诊断仪和车辆内部 ECU 之间进行大量数据通信，上下载数据。

### 3 协议栈开发

按照 ISO15765 的系统架构，协议栈的总体软件结构如图 2 所示。整个协议栈分为 3 个部分：应用层，网络层和 CAN 协议部分。各层为上层提供服务。

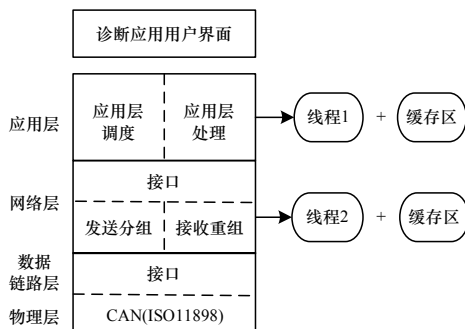


图2 ISO1576 协议栈的软件总体设计

为了能实时地接收到底层数据，分别为应用层和网络层提供了接收线程和存储数据的缓存区。

### 3.1 应用层实现

应用层为诊断服务提供了 6 个服务原语，如图 3 所示，每个原语的具体含义及本文所提供的接口如表 1 所示。

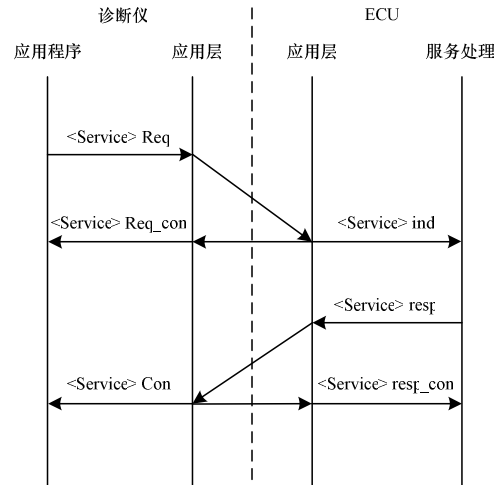


图3 应用层服务原语

表1 应用层服务原语接口

服务原语	含义	接口函数
请求原语(Req)	请求某个诊断服务	xxx_Req()
请求确认原语(Req_con)	表示请求报文是否传送到 ECU 端	SendReqCon()
确认原语(Con)	接收响应数据	SendConfirm()
指示原语(ind)	接收请求数据	SendInd()
响应原语(resp)	发送响应报文	xxx_Resp()
响应确认原语(resp_con)	表示响应报文是否发送到诊断仪端	SendRespCon()

本文将应用层分为 2 个模块，分别是应用层调度和应用层处理。

(1)应用层调度: 主要是为应用层处理模块服务的，作为诊断仪时，根据具体的用户诊断应用调用相关的处理函数，当接收到响应数据时根据报文服务标识符 A\_PCI 判断属于哪种服务的响应，并调用相关的应用处理函数；作为车辆内部 ECU 时，主要功能是解析请求服务标识符并进行预处理，当接收到来自诊断仪的诊断服务请求时，根据报文服务标识符 A\_PCI 判断是属于哪种服务请求，然后据此来调用相应的应用处理函数，当 ECU 不支持该服务请求时直接发送负响应到诊断仪端。

(2)应用层处理: 应用层处理主要是负责处理一些公用诊断服务，如读取和清除故障诊断信息、开始会话、保持通信连接、安全访问等。作为诊断仪端时，经过应用层调度后，进入诊断请求服务处理，该处理主要是将请求报文组装成相应的请求 A\_PDU(应用层协议数据单元)，即赋予 A\_PCI 相应的服务标识，并根据协议规定的请求报文格式填充各参数，然后发送到网络层，当接收到响应报文时，经过应用层调度后，调用相应的应用层处理函数，该

处理函数主要是根据协议规定的相应响应服务格式解析各个参数,并将各个参数数据反馈给用户,同时也做一些错误处理,如响应报文格式错误或子功能不支持等错误;作为 ECU,应用层处理主要是根据具体的来自诊断仪的服务请求进行处理,并发送相应的响应报文,如接收到的是读取故障信息的请求,则 ECU 取出存在内部的故障代码,并将这些故障代码信息封装到读取故障信息响应报文中,然后发送到诊断仪端。

应用层的软件是基于状态转换来实现的,应用层的状态主要有准备、接收网络层数据、应用层调度和应用层处理,图 4 为应用层的状态切换。

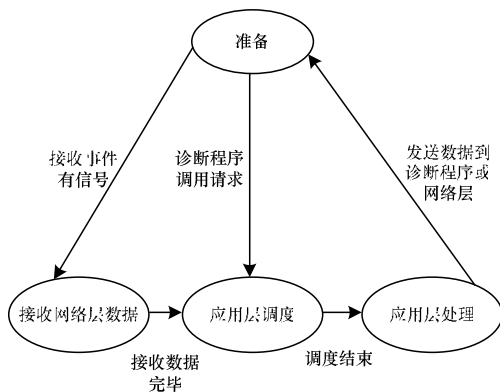


图 4 应用层状态切换图

应用层经初始化后先进入准备状态。

作为接收端,当应用层接收线程的事件有信号时,接收来自网络层的数据;接收完数据进入到应用层调度的状态,在此状态中根据接收数据首字节即 A\_PCI 来调度相应的服务处理,进入到应用层处理状态,处理完后,将数据反馈到用户诊断程序,然后又进入到准备状态。

作为发送端,诊断程序调用应用层的服务请求,应用层进入应用层调度状态;调度结束后应用层进入应用层处理状态,在此状态中将数据封装成相应的应用层协议数据单元,最后将数据发送到网络层然后又进入到准备状态。

### 3.2 网络层实现

网络层提供了 2 个功能:(1)为上层提供服务,向应用层提供服务接口;(2)内部操作。

网络层提供给上层的服务是由网络层接口完成的,其定义的 4 个通信服务原语及本文所提供的接口函数见表 2。

表 2 网络层服务原语

服务原语	含义	接口函数
请求原语 N_USData.req	请求发送数据	N_USData_Req()
确认原语 N_USData.con	表示请求数据是否发送成功	SendNConPrim()
首帧指示原语 N_USData_FF.ind	接收数据首帧	SendNFFInd()
指示原语 N_USData.ind	接收到报文所有数据	SendNInd()

内部操作主要是对报文数据进行分组和重组。由于数据链路层用到的是 CAN 协议,CAN 数据单元只含有 8 Byte,则当报文数据大于 8 Byte 时,要对此进行分组或重组操作。对此网络层定义了 4 种 N\_PDU(网络层的协议数据单元)的格式:单帧(SF),首帧(FF),连续帧(CF)和流控制帧(FC),这 4 种帧是根据网络层协议控制标识(N\_PCI)来标识的。当请求原语中所要传输的数据长度小于 8(本地诊断)或 7(远程诊断)时,就被直接封装成 SF 单帧的格式发送到数据链路层。反之,就要对报文数据进行分组,其最多可传输 4 095 Byte 的数据。图 5 为单帧情况下的通信。

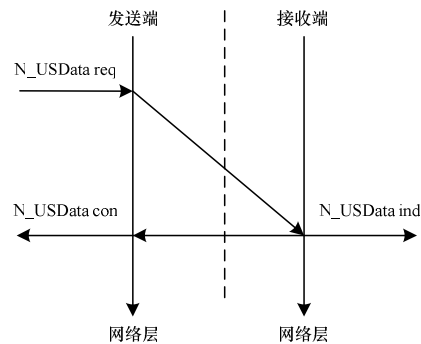


图 5 单帧通信

图 6 为多帧情况下对等实体之间的通信。当网络层请求数据大于 7 或 6 时,发送端要进行分组,其分组过程是:首先将网络层请求报文分组成 FF 和多个 CF,每个 CF 帧都含有帧序号 SN,由 1 开始,先发送 FF 帧,然后等待接收端发来 FC 帧,当接收到 FC 帧时,根据 FC 的参数 BS(Block Size)来发送连续 BS 个 CF 帧,然后再等待 FC 帧,这样持续下去直到把所有的 CF 帧发送完,当发送完所有 CF 帧后,发送 N\_USData.con 到应用层表示成功地将整个请求报文发送到接收端。当分组失败,如在规定的时间内接收不到 FC 帧,则发送带有表示错误信息 N\_Result 的 N\_USData.con 到应用层,表示发送失败。

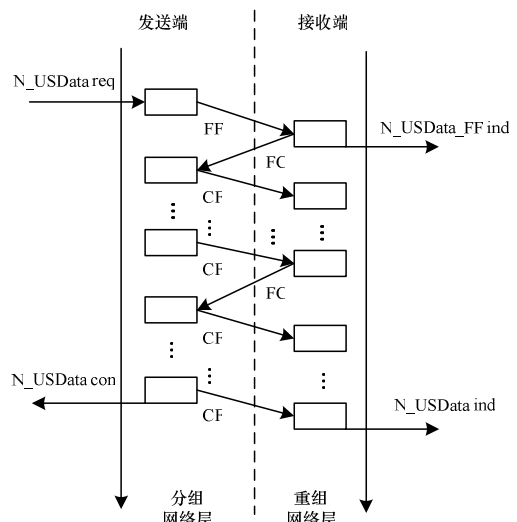


图 6 多帧通信

对应的接收端也要对数据进行重组,重组模块的过程如下: FF 帧中包含报文总长度的参数,当接收到来自发送端的 FF 帧后,向应用层发送首帧指示原语,并根据自身接收能力设定 BS 参数(达到了流控制的功能),然后发送带有 BS 参数的 FC 帧到发送端,告知发送端可发送的 CF 个数,然后等待接收 CF 帧,接收完 BS 个 CF 帧后,再发送 FC 帧,直到接收完所有的 CF 帧,然后根据接收到 FF 和 CF 的序号 SN 进行重组,FF 隐藏的帧序号 SN 为 0,根据 FF 帧的报文总长度重组完报文后将以指示原语的形式发送到应用层。如果重组失败,即在规定时间内接收不到 FF、CF 帧或序号不符合要求,则发送带有表示错误信息 N\_Result 的 N\_Data.ind 原语到应用层。

网络层软件的实现也是基于状态转换的,网络层有以下状态:准备,网络层预处理,报文重组和报文分组。图 7 为网络层的状态切换图,表 3 为状态转换表。

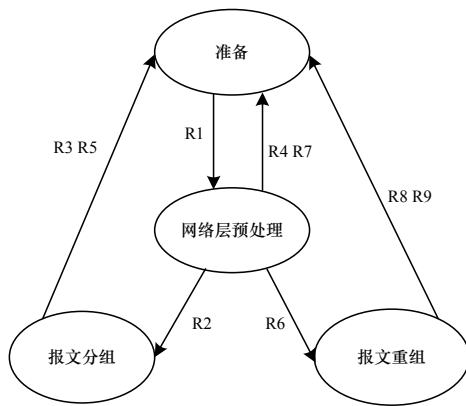


图 7 网络层状态切换图

表 3 网络层状态转换表

条件	描述
R1	接收数据
R2	来自应用层请求,并且数据大于 7 或 6;或接收到 CF 帧
R3	分组成功,发送 N_Data.con 到应用层
R4	来自应用层请求,数据长度小于 7 或 6 单帧发送
R5	分组失败,发送带有 N_Result 表示错误信息的 N_Data.con 到应用层
R6	接收到来自数据链路层的 FF、CF
R7	接收到单帧,直接发送数据到应用层
R8	重组成功,发送数据到应用层
R9	重组失败,发送带有参数 N_Result 表示错误信息的 N_Data.ind 到应用层

网络层预处理是为网络层报文重组和报文分组服务的,其根据接收到的数据首先判断是来自应用层的请求还是来自网络层的数据,如果是应用层的请求,判断请求报文数据域长度,当数据域长度小于 8 或 7 时,即可封装成单帧直接发送,反之进入报文分组的状态;如果是来自数据链路层的数据帧,将数据帧解析成 N\_PDU 的格式并判断是哪种类型的 N\_PDU 帧,若是 FC 帧,则进入报文分

组的状态,若是 SF,则可直接以指示原语的形式发送到应用层,若是 FF 或 CF 帧,则进入报文重组的状态。报文分组和报文重组模块上文已有描述。同应用层,经初始化后网络层先进入准备状态。当线程事件有信号时,接收数据后进入网络层预处理的状态,处理完后根据不同的情况进入到不同的状态。

### 3.3 数据链路层实现

数据链路层的实现是由支持 ISO11898 协议 CAN 控制器驱动来实现的。因此,该部分主要讲述数据链路层向上层提供的接口实现。数据链路层向网络层提供了 3 个接口,见表 4,接口主要实现了网络层数据与 CAN 数据帧之间的映射,并封装了 CAN 控制器驱动所提供的函数。

表 4 数据链路层接口

接口函数	含义
LDataReq()	发送请求数据
LDataCon()	发送请求数据后的确认函数
LDataInd()	接收到请求的数据

### 3.4 ISO15765 协议报文处理流程

图 8 是 ISO15765 协议报文处理流程。

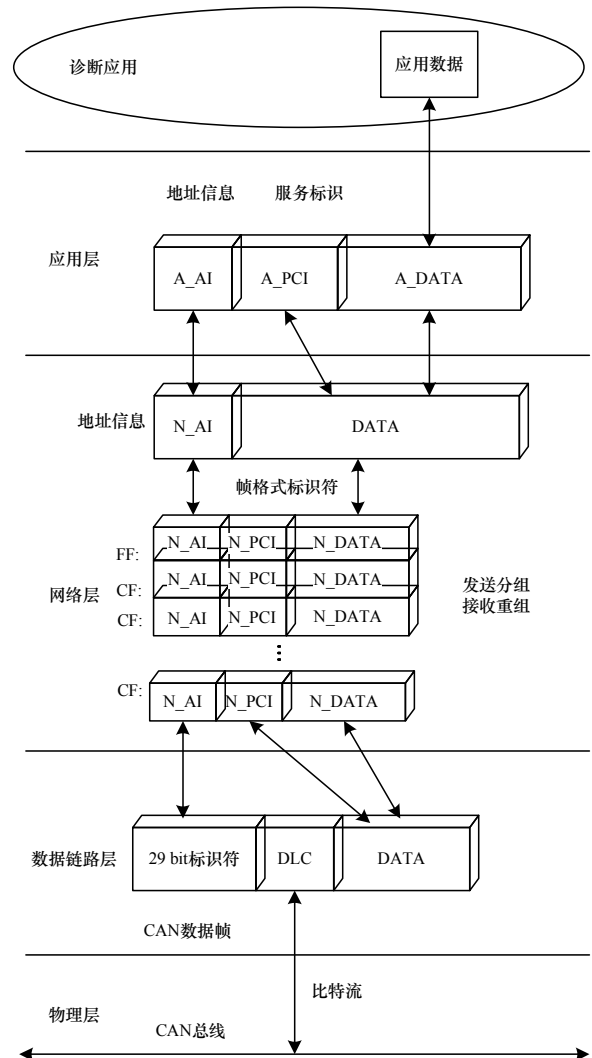


图 8 ISO15765 协议报文处理流程

A\_PDU 包括 3 个部分, 分别是 A\_AI(地址信息——包括源地址和目的地址)、A\_PCI(服务标识符)和 A\_DATA(服务数据)。

N\_PDU 也包括 3 个部分, 分别是 N\_AI(地址信息——包括源地址和目的地址)、N\_PCI(帧格式标识符)和 N\_DATA(帧数据)。

当应用层的数据发送到网络层, A\_AI 映射到 N\_AI, A\_PCI 和 A\_DATA 被封装到网络层服务数据单元的数据域 DATA。当数据域长度大于 7 或 6 时, 先分组, 分组后的每个 N\_PDU 都包含一个 N\_AI, N\_PCI 标识帧类型, N\_DATA 是分组后的数据, 数据长度小于 8。

数据链路层的协议数据单元是 CAN 数据帧, CAN 数据帧也由 3 个部分组成, 分别是 29 位标识符、数据控制信息 DLC 和数据域 DATA。当网络层数据发送到数据链路层时, N\_AI 映射到数据链路层的 29 位标识符, N\_PCI 和 N\_DATA 被封装到 CAN 数据帧的数据部分。DLC 标识该 CAN 数据帧数据域中的有效数据长度。

由数据链路层的 CAN 数据帧到网络层再到应用层是一个反过程。

#### 4 通信验证

为了测试本文设计的有效性, 将该协议应用到 PC 机上, 组装成基于 PC 机的诊断仪, 图 9 中的 ECU 是一个支持 ISO15765 的电子控制单元, 测试诊断仪与 ECU 之间的通信。诊断仪中包括 4 个部分: 测试案例集, 测试封装, ISO15765 通信栈和界面。图 9 是测试环境。

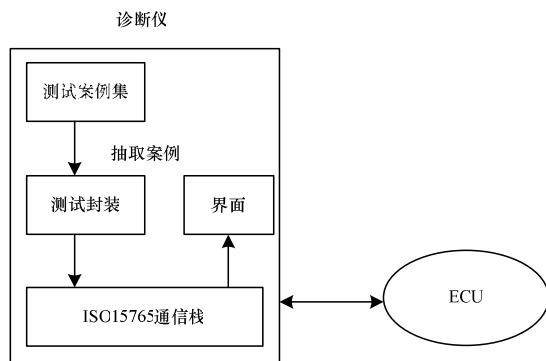


图 9 测试环境

测试封装是对抽取的案例封装成通用的诊断服务请求, 然后调用 ISO15765 的通信服务与 ECU 进行通信。

界面用来显示来自 ECU 的响应的数据信息。其中, 测试案例集覆盖以下的测试要求:

(1)测试应用层各个有关诊断服务发送和接收的情况;

(2)测试网络层数据分组、重组和流控制管理的实现情况;

(3)测试网络层分组失败和重组失败的情况;

(4)观察协议报文从应用层到数据链路层以及从数据链路层到应用层的报文处理流程。

在测试过程中, 通过调试对整个通信过程进行跟踪。测试表明, 该协议栈满足 ISO15765 规定的要求, 也满足一般的故障诊断要求。

#### 5 结束语

ISO14229(UDS)是非常完善的车辆故障诊断协议标准, ISO15765 是一套基于 CAN 总线实现 UDS 的协议标准, 是今后的发展趋势之一。本文研究分析并开发了故障诊断 ISO15765 协议栈。协议栈设计中采用分层接口方式, 下层向上层提供接口函数。协议内部功能采用模块化设计, 确保了协议栈的可剪裁性和可移植性。该协议栈可应用于基于 PC 机的诊断仪和手持式的诊断仪, 同时也可用于支持 ISO15765 车辆内部 ECU 的设计。

#### 参考文献

- [1] 阳宪惠. 工业数据通信与控制网络[M]. 北京: 清华大学出版社, 2003.
- [2] International Organization for Standardization. ISO 14229-2006 Road Vehicles—Unified Diagnostics Services(UDS) Specification and Requirements(v2)[S]. 2006.
- [3] International Organization for Standardization. ISO 15765-1-2004 Road Vehicles—Diagnostics on Controller Area Networks (CAN)—Part1: General information[S]. 2004.
- [4] 田晓川, 王励明, 闫厉, 等. 2 种汽车诊断协议对比浅析[J]. 汽车电器, 2008, (12): 6-8.
- [5] 徐杭, 白瑞林, 严惠. CAN 总线上层协议的设计[J]. 计算机工程, 2007, 33(24): 258-260.
- [6] International Organization for Standardization. ISO 11898-1-2003 Road Vehicles—Controller Area Network(CAN)—Part1: Data Link Layer and Physical Signaling[S]. 2003.
- [7] International Organization for Standardization. ISO 15765-2-2004 Road Vehicles—diagnostics on Controller Area Networks (CAN)—Part2: Network layer services[S]. 2004.
- [8] International Organization for Standardization. ISO 15765-3-2004 Road Vehicles—Diagnostics on Controller Area Networks (CAN)—Part3: Implementation of Unified Diagnostic Services (UDS on CAN)[S]. 2004.

编辑 任吉慧