

# 基于 OSEK COM 规范的消息传输机制优化

李银国, 李 萍, 蒋建春

(重庆邮电大学汽车电子嵌入式系统研究所, 重庆 400065)

**摘要:** OSEK/VDX 规范在汽车电子控制系统开发中具有重要地位, OSEK/VDX 通信规范(OSEK COM)定义的消息通信传输机制对于系统性能有着重要影响。该文分析 OSEK COM 规范关于消息外部通信过程的 3 种传输机制, 针对传输机制中存在的消息丢失现象, 提出在消息结构中增加不可覆盖属性的消息传输机制优化策略, 并给出实现方法和测试结果。

**关键词:** 嵌入式实时操作系统; OSEK COM 规范; 传输机制; 消息通信

## Optimization of Message Transmission Mechanism Based on OSEK COM Specification

LI Yin-guo, LI Ping, JIANG Jian-chun

(Institute of Automotive Electronic and Embedded System, Chongqing University of Posts and Telecommunications, Chongqing 400065)

**【Abstract】** OSEK/VDX specification plays an important role in the automotive electronic control system, and system performance is significantly affected by message transmission mechanism based on OSEK/VDX communication specification. This paper analyzes three transmission mechanisms in the external communication, and proposes a new optimization strategy for message transmission mechanisms which adds uncovered property in the message structure in order to deal with message loss phenomenon. Implementation method and test results are represented.

**【Key words】** embedded real-time operating system; OSEK COM specification; transmission mechanism; message communication

### 1 概述

OSEK/VDX规范(OSEK)提供了将各种汽车电子设备有机地结合在一起的网络化嵌入式计算机系统的开发规范。OSEK/VDX通信规范(OSEK COM)<sup>[1]</sup>为汽车电控单元应用软件提供了一个统一的通信环境, OSEK COM的接口独立于总线系统, 允许应用软件跨越不同总线平台重复应用。OSEK COM提供了定义明确的API, 可以提供一个可重复使用的通信内核来支持不同类型的应用。此外OSEK COM还支持应用软件的移植性, 在一个专门用于交互任务的通信中, 应用系统可以为任何一个ECU提供一个OSEK兼容的接口。

目前随着集成电路和单片机在汽车上的广泛应用, 越来越多的电控单元(ECU)被应用到汽车控制领域, 如汽车刹车的防抱死系统、动力设备的安全控制等。ECU 内部和 ECU 之间的通信已经成为汽车电子控制系统开发的重要环节, 在汽车电子和其他嵌入式领域有着良好的应用前景。因此, OSEK COM 消息传输机制的研究和优化具有重要意义。

OSEK COM 定义了消息的 3 种外部传输机制, 这 3 种传输机制不能保证每条消息都能够传输到下层总线系统, 存在消息丢失的情况, 这给汽车电子控制的安全性和可靠性带来了较大的影响。

本文深入分析了消息外部通信过程的 3 种传输机制和消息丢失现象的原因, 提出在消息结构中增加不可覆盖属性的消息传输机制优化方案来扩充消息传输机制的功能, 并给出了具体实现。

### 2 OSEK COM 的工作机制

#### 2.1 OSEK COM 通信模型

OSEK COM 通信模型分为 3 个层次: 应用层(Application)

为上层;交互层(Interaction)为中间层;网络层、数据链路层、物理层统称为下层, 如图 1 所示。主要部分是交互层, 该层全权处理内部通信, 并通过调用下层服务协议处理外部通信。

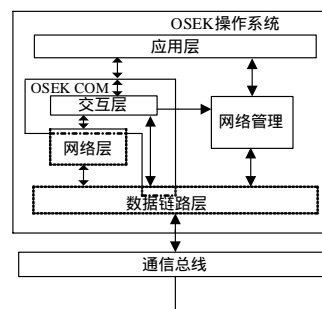


图 1 OSEK COM 及它在 OSEK 体系中的位置模型

#### 2.2 交互层的分析及通信过程

OSEK COM是基于消息对象的通信, 消息及其属性通过 OSEK实现语言(OSEK Implementation Language, OIL)静态配置<sup>[2]</sup>。在内部通信过程中, 应用层调用交互层提供的发送消息API, 将发送方的数据传给交互层的消息对象, 消息对象直接被复制到接收消息对象, 然后接收方调用接收消息API, 从接收消息对象中读取消息数据。在外部通信过程中, 发送方的一个或多个消息对象的数据按比特位对齐被映射到一个

**基金项目:** 国家“863”计划基金资助项目(2006AA11A1C1-3)

**作者简介:** 李银国(1955 -), 男, 教授、博士, 主研方向: 模式识别, 人工智能, 系统辨识与智能控制; 李 萍, 硕士研究生; 蒋建春, 讲师、博士研究生

**收稿日期:** 2008-05-23 **E-mail:** lpylqq@163.com

发送交互层协议数据单元(IPDU)的数据区上,然后交互层调用底层协议(例如CAN协议、串口协议)将数据区发送出去;接收方的接收与发送方的过程相反,在一个接收指示请求后,先由底层协议收取数据到接收IPDU数据区,然后从IPDU数据区分出各接收消息对象的数据,完成接收过程。

### 2.3 外部消息传输机制

当发送外部消息时,在交互层消息依次通过过滤算法,字节顺序转化,最后根据其传输模式封装到相应的IPDU。OSEK COM规范定义了3种不同的消息传输模式<sup>[3]</sup>:

(1)直接传输机制。当IPDU中的消息传输特性定义为触发式且IPDU采用直接传输模式时,从交互层到底层的一个传输请求后,直接传输立即进行。每个IPDU应设置每两次传输之间的最小延迟时间(L\_TMD\_MDT),即前一次传输得到底层的传输确认到下一次传输获得传输请求的时间间隔。如果一次传输立即进行后,在给定的时间间隔内没有得到底层的传输确认即传输死限监控视为错误,那么下一个传输立即开始执行,且不被延迟,如图2所示。

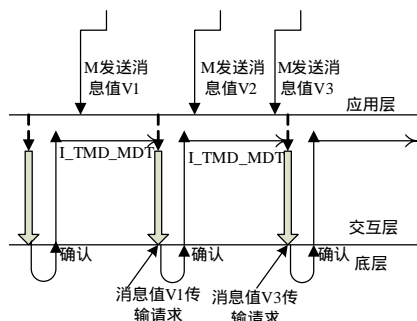


图2 直接传输模式传输机制

(2)周期传输机制。每次调用API函数SendMessage(),SendDynamic Message()的服务更新传输的消息对象,交互层每隔周期传输模式时间间隔(L\_TMP\_TPD)执行一次周期传输请求,传输一个IPDU到底层,当有传输请求时才执行IPDU到底层的消息传输。周期传输模式忽略包含在IPDU里面的所有消息的传输特性。当消息已经发送到底层后,如果在规定的时间内,底层没有返回传输确认,那么立刻调用通知机制通知应用层,消息的周期传输失败。

(3)混合传输机制。混合传输机制是直接传输机制和周期传输机制的结合。交互层每隔混合传输模式时间间隔(L\_TMM\_TPD)执行一次周期传输请求,传输一个IPDU到底层。同时当IPDU中具有触发特性的消息,并获得传输请求时,会立刻产生中间传输。中间传输和中间传输之间,周期传输和中间传输之间必须设置最小的延迟时间(L\_TMM\_MDT)。

### 2.4 3种传输机制的缺陷

本文以直接传输模式为主进行说明,在最小延迟时间结束前,如果同一个消息对象有多次用不同的消息进行更新并传输,那么前面传输的消息对象的消息将会被后面传输的消息覆盖,并没有被传输到底层。直到最小延迟时间结束后,最后一次的传输请求被立即执行。如图2所示:消息对象M在最小延迟时间结束前依次用消息值V2、V3进行更新,这时还没有传输请求,所以V2被V3覆盖,当最小延迟时间结束后,消息对象获得传输请求,V3被立刻传输到底层,V2丢失。假如在最小延迟时间结束前,消息对象M被10个甚至更多的消息值进行更新,当最小延迟时间结束后,只有最后

一个消息值传输到底层,前面所有消息均丢失了。消息的丢失现象同样出现在周期传输模式和混合传输模式中。3种传输机制均不能保证重要消息不会被覆盖。

## 3 消息传输机制的改进

### 3.1 消息传输机制改进的必要性

汽车上的控制系统直接关系到人的生命安全,即使出现丝毫的差错也会危及生命。因此,在实际的汽车电子控制系统的应用中,电控单元间的通信需要及时和准确的完成。在现有的传输机制中,消息之间的传输设置了最小延迟时间,周期消息的传输设置了周期传输时间间隔,在时间间隔内同一个消息对象的消息允许被覆盖,给消息的丢失现象留下了隐患。

### 3.2 消息传输机制改进方案

本文阐述了直接传输机制的改进思想和算法。在周期传输模式和混合传输模式中,该算法也适用。

在直接传输过程中,在最小延迟时间结束前,当消息传输时,判断消息是否具有可覆盖性,若消息允许被覆盖,直接按照OSEK COM中传输机制进行传输,若消息不允许被覆盖,将消息放到先入先出消息队列中。待最小延迟时间结束后,首先将消息队列中第1条消息取出,放入IPDU结构中,并调用底层函数,其功能是:将消息传输到底层。在得到底层确认后,表明该消息传输成功,实现了电控单元间的通信。再取出消息队列中第2条消息进行传输。若第1条消息在给定的时间间隔内没有得到底层确认,表明该消息传输失败,立刻取出消息队列第2条消息进行传输,直到取出该消息队列中最后一条消息,调用直接传输函数,其功能是:将消息传输到底层,记录当前系统时钟节拍,给直接传输请求的标志清零<sup>[4]</sup>,该方案流程如图3。

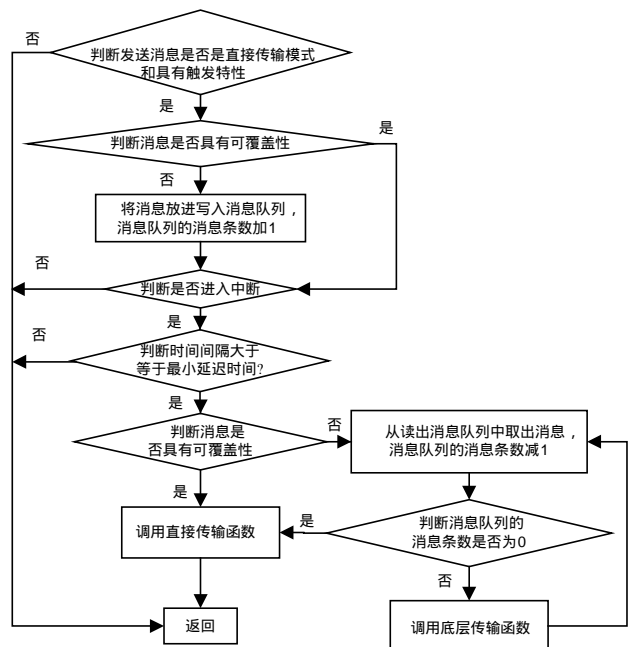


图3 改进后的直接传输机制流程

### 3.3 直接传输机制改进方案的具体实现<sup>[5]</sup>

(1)消息结构的改动,在消息的结构体中增加可覆盖性属性:

```
typedef struct COM_MSG {
    INT8U MSGcover;
}COM_MSG;
```

(2)增加先入先出的消息队列的宏定义结构：

```
/*消息队列首地址*/
#define OSFIFOFIRST(m) OsMsgPtr[m]
/*消息队列末地址*/
#define OSFIFOLAST(m) OsFifoLast[m]
/*消息队列最大的消息条数*/
#define OSFIFOLENGTH(m) OsFifoLength[m]
/*消息写入消息队列指针*/
#define OSFIFOWRITE(m) OsFifoWrite[m]
/*消息读出消息队列指针*/
#define OSFIFOREAD(m) OsFifoRead[m]
/*消息队列中已有的消息条数*/
#define OSFIFOCNT(m) OsFifoCnt[m]
```

采用写入消息队列指针将消息从首地址依次放入消息队列，当到达该队列的末地址后，下一条消息又放入消息队列的首地址，形成一个由指针组成的循环缓冲区，可以用读出消息队列指针以同样的顺序取出消息。

(3)消息放入消息队列的实现：

```
if( ! TOMSG[MSG]. MSGcover) //如消息具有不可覆盖性
{
if(OSFIFOCNT(m)==
OSFIFOLENGTH(m)) //如消息队列中已有的消息数为可容纳的
//最大消息条数
{
返回; //说明队列已满
}
else
{
*OSFIFOWRITE(m)=
*(TOMSG[MSG].MsgData); //将消息写入到消息队列
if(OSFIFOWRITE(m)==
OSFIFOLAST(m)) //如写入消息地址是消息队列末地址
{
OSFIFOWRITE(m)=
(void*)OSFIFOFIRST(m);
//将写入消息地址指向消息队列首地址
}
else
{
OSFIFOWRITE(m)++; //将写入消息地址后移
}
OSFIFOCNT(m)++; //更新消息队列中已写入的消息条数
}
}
```

4)消息从消息队列取出的实现：

```
if( ! TOMSG[MSG]. MSGcover)
{
*(TOMSG[MSG].MsgData)=
*OSFIFOREAD(m); //从消息队列中取出消息
*OSFIFOREAD(m)=0; //并将该地址的消息值清零
OSFIFOCNT(m)--; //更新消息队列消息条数
while(OSFIFOCNT(m) != 0) //当消息队列消息条数不为 0
{
```

```
Low_sendMsg(Msg); //调用底层传输函数
if(OSFIFOREAD(m)==
OSFIFOLAST(m)) //如读出消息地址是消息队列的末地址
{
OSFIFOREAD(m)=
(void*)OSFIFOFIRST(m);
//将读出消息地址指向消息队列首地址
}
else
{
OSFIFOREAD(m)++; //将读出消息地址后移
}
*(TOMSG[MSG].MsgData)=
*OSFIFOREAD(m); //从消息队列中取出消息
*OSFIFOREAD(m)=0; //并将该地址的消息值清零
OSFIFOCNT(m)--; //更新消息队列消息条数
}
if(OSFIFOCNT(m)==0)
{
COMDirecttransmission(Msg); //调用直接传输函数
}}
```

#### 4 改进算法的效果分析和验证

在实际应用中,改进算法的关键为消息队列长度的配置。消息队列长度太长,会占用多余内存;消息队列长度太小,同样会发生消息放入消息队列后还没有被取出就被后面放入消息覆盖的现象,所以,需要根据实际应用,合理配置参数。

本文选择 CAN 总线作为底层通信协议,在硬件上选择 16 bit 单片机 Mortolona HCS12DP256B,并用 ZLGCANTest 测试软件的界面来显示消息从应用层发送到底层消息接收的情况。经多次对比测试,改进算法在 3 种传输模式下消息的接收率可以达到 94%以上,测试验证该方法的可行性和正确性。

#### 5 结束语

本文在对 OSEK COM 规范的 3 种传输模式进行分析的基础上,提出了新的传输机制改进方法,即增加消息的不可覆盖性属性和简单的消息队列,占用的内存较小,对该规范的改动较小,对外部通信的 3 种机制均可适用。实际测试验证了改进后消息的丢失情况得到显著改善,对提高汽车电子控制系统中电控单元的通信效率和效果具有重要的意义。

#### 参考文献

- [1] OSEK/VDX Organization. OSEK/VDX Communication Specification 3.0.3[EB/OL]. (2004-07-20). <http://www.osek-vdx.org>.
- [2] OSEK/VDX Organization. OSEK/VDX System Generation, OIL: OSEK Implementation Language[EB/OL]. (2004-07-01). <http://www.osek-vdx.org>.
- [3] Joseph L. OSEK/VDX 汽车电子嵌入式软件编程技术[M]. 罗克露,译. 北京:北京航空航天大学出版社,2004.
- [4] Moron T D. 嵌入式微控制器[M]. 严隽永,译. 北京:机械工业出版社,2005.
- [5] Labrosse J J. 嵌入式实时操作系统 uC/OS-II[M]. 邵贝贝,译. 北京:北京航空航天大学出版社,2003.