

# OSEK/VDX 嵌入式实时操作系统在汽车稳定性控制器中的应用

郜文<sup>1,2</sup> 李继来<sup>1,2</sup> 梁华为<sup>1</sup>

(1.中国科学院合肥智能机械研究所 安徽 合肥 230031;2.中国科学技术大学 自动化系 安徽 合肥 230027)

**摘要:** 根据 OSEK/VDX 标准设计嵌入式实时操作系统,侧重于任务管理,资源管理部分的实现。并在自主开发的汽车稳定性控制器硬件在环仿真平台上进行验证。实验结果表明,在该操作系统上开发系统可以有效地提高实时性和稳定性,能够满足汽车稳定性控制器的要求。

**关键词:** OSEK/VDX; 嵌入式; 实时; 操作系统; 汽车稳定性控制器

## Application of OSEK/VDX Embedded Real-time Operating System to Vehicle Stability Controller

GAO Wen<sup>1,2</sup>, LI Ji-Lai<sup>1,2</sup>, LIANG Hua-Wei<sup>1</sup>(1. Institute of Intelligent Machines, Chinese Academy of Sciences, Hefei 230031, China; 2. University of Science and Technology of China, Hefei 230027, China)

**Abstract:** This paper designs an embedded real-time operating system according to OSEK/VDX standard. It focuses on the design of task and resource management. Then it is verified on the automobile stability control hardware-in-loop simulation platform. The result of verification shows that developing on this OS can efficiently improve the real-time performance and stability of the system and it can also meet the requirements of automobile stability control system well.

**Keywords:** OSEK/VDX; embedded; real-time; operating system; vehicle stability controller

## 1 引言

OSEK/VDX 标准是当前流行于汽车行业内的汽车电子分布式控制的开放平台。采用符合 OSEK/VDX 标准的嵌入式实时操作系统开发产品,可以提高代码的复用率、降低开发成本、缩短产品开发周期,并增强系统的实时性、可移植性和扩展性。近年来,许多半导体厂商也在自己的产品中融入了支持实时操作系统的结构,例如摩托罗拉公司 freescale 系列微控制器绝大多数都支持实时操作系统,这更加促进了嵌入式实时操作系统在汽车电子中的发展<sup>[1]</sup>。

汽车电子稳定性控制系统(ESC)是目前汽车主动安全技术的一项重要内容,较之传统的控制器,ESC 系统具有更多的传感器,执行器以及更为复杂的控制算法,若采用传统的开发方法,不仅难以严格满足实

时性,也不便于更好的利用控制器本身的硬件资源,同时也会给后续的开发带来一些困难。采用基于 OSEK/VDX 标准的嵌入式实时操作系统来进行控制器的软件开发将有效的解决上述问题。

## 2 OSEK/VDX 嵌入式实时操作系统

### 2.1 OSEK OS 体系结构

按照 OSEK/VDX 规范定义,OSEK OS 按功能分为任务管理和调度、资源管理、警报管理、计数器管理、事件管理和中断管理<sup>[2]</sup>,其结构组成如图 1 所示

OSEK 定义的处理级别为:(1)中断级(2)调度器逻辑级(3)任务级。

优先级规则是:(1)中断级大于任务级;(2)中断级包括一个或多个优先级;(3)静态分配中断服务程序的

收稿时间:2009-07-14;收到修改稿时间:2009-08-25

中断优先级,具体的分配取决于实现和硬件结构;(4)静态分配任务优先级。

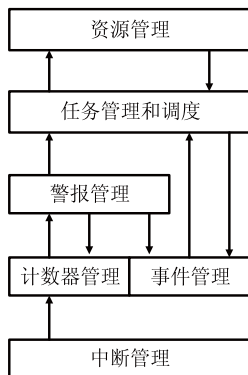


图 1 OSEK OS 结构组成

OSEK 规范中的任务分为基本任务(BT)和扩展任务(ET)。基本任务具有运行态、就绪态和挂起态共三种状态。扩展任务多了一个等待态。基本任务只在开始和结束时才有同步点,需要的资源少,而扩展任务可以对应不同的时间,在运行中可能有多个同步点,所以对环境要求比较高。

相应于基本任务和扩展任务,OSEK 规范定义了符合类的概念,共有 4 种符合类别 BCC1, BCC2, ECC1, ECC2。它们的定义如图 2 所示。

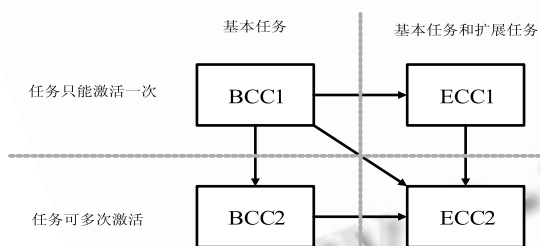


图 2 复合类别定义

## 2.2 OSEK OS 运行机制分析<sup>[2]</sup>

### (1) 任务管理与调度

上面已经提到, OSEK OS 提供两种不同的任务概念:基本任务和扩展任务。

任务调度的机制分为:完全可抢占式调度机制

不可抢占式调度机制 混合抢占调度机制。不管是哪种调度策略,用户可以通过系统服务 Schedule 在任何它想重新分配 CPU 的地方重新调度。除了完全可抢占调度机制的应用程序,其它调度机制都必须要使

用系统服务 Schedule。

### (2) 中断处理

OSEK OS 规定的中断处理分为两类:中断服务程序类别 1(ISR Category 1),中断服务程序不使用操作系统服务,当一个中断服务程序结束时,在被中断的指令处继续执行,也就是说中断对任务管理没有影响,这个类别的中断程序有最小的额外开销;中断服务程序类别 2(ISR Category 2),OSEK OS 提供一个中断框架为用户程序准备一个运行时环境,在系统生成时,用户程序就被分配给这个中断。

### (3) 事件机制

只有扩展任务类型才会用到事件机制,事件机制可以将一个任务从等待状态转换到就绪状态,或从运行状态转换到等待状态。操作系统提供服务来设置、清除、询问事件以及等待事件的发生。在任何情况下,一个事件的接受者都是一个扩展任务,因此一个中断服务程序或者一个基本任务等待一个事件都是不可能的。

### (4) 资源管理

OSEK OS 资源管理内容的一个特点是其避免优先级翻转的方法——优先级天花板协议。其主要内容如下:在系统生成时,对于每个资源来说,它自己的最高限度优先级是被静态分配的。如果一个任务需要一个资源,并且这个任务当前优先级比这个资源的最高限度优先级低,那么这个任务的优先级就被重新设置成这个资源的最高限度优先级。当一个任务释放资源时,它的优先级被动态重置为访问这个资源前的优先级。

### (5) 计数与警报管理

一个计数器实际上是一个计数值,以脉冲为量度,而且一般的计数器都是一个常量。OSEK OS 不提供标准的 API 来直接实现计数。

当一个报警器期满时, OSEK 提供了一些服务来激活一个任务、设置一个事件或者是调用一个报警器反馈程序。一个报警器反馈程序就是由应用程序提供的小函数,当达到预先定义的定时器的值时,这个报警器就期满了<sup>[3]</sup>。

## 3 汽车稳定性控制器硬件平台

主控制器采用摩托罗拉公司主要应用于汽车电子的 Freescale MC9S12DG128 单片机, MC9S12DG

128 微控制器是 Motorola 公司 M68HC12 系列 16 位单片机中的一种,其内部结构主要有单片机基本部分和 CAN 功能块组成,基本结构包括:中央处理器单元 HCS12(CPU),2 个异步串行通信口 SCI,2 个同步串行通信口 SPI,8 通道输入捕捉/输出比较定时器,1 个 8 通道脉宽调制模块以及 49 个独立数字 I/O 口(其中 20 个具有外部中断及唤醒功能)在片内还拥有 128KB 的 Flash ROM,8KB 的 RAM,2KB 的 EEPROM,CAN 功能块包括两个兼容 CAN2.0A/B 协议的 msCAN 控制器组成,这些丰富的内部资源和外部接口资源可以满足 ECU 对各种数据的处理以及 CAN 网络数据的发送和接收要求。

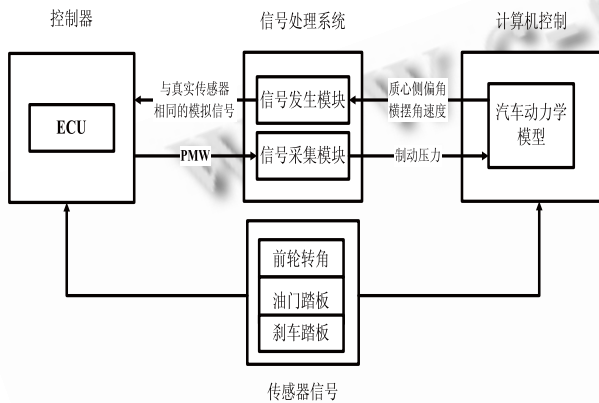


图 3 汽车稳定性控制实验平台结构

由于 FLASH 存储器具有在系统中重新编程的能力,并可用于非易失性数据存储,而且允许现场更新,所以,可以很方便地将操作系统固化到控制器中。

控制单元的主要作用就是通过这些传感器信号对车辆的运行状态进行判断,进而发出控制指令,并自动向一个或多个车轮施加制动力,以使车辆保持在驾驶员所选定的轨迹内<sup>[4]</sup>。

传感器记录车辆的一些物理变量,包括:前轮转角、油门踏板和刹车踏板。基于这些数据,通过 ECU 系统分析驾驶员对方向盘的操作方向,并计算车辆是否遵照驾驶员提出的转向要求行驶,然后有针对性地对各个车轮实施制动。同时这些制动压力与物理传感器信号通过计算机内的汽车动力学模型计算出质心侧偏角以及横摆角速度,通过信号发生装置转换为模拟信号后,也作为 ECU 的输入,用于产生制动压力。这样就形成了汽车稳定性控制的硬件在环实验平台。

#### 4 基于 Freescale MC9S12DG128 设计 OSEK OS

OSEK OS 的设计参照嵌入式操作系统 uC/OS-II 的架构,为了将来移植到其他硬件平台的方便,首先把内核分为硬件无关,硬件相关和任务相关三个部分,与应用相关的内核配置选项放置在任务相关部分<sup>[5]</sup>。

我们这里设计的操作系统,遵循 BCC1 级别的定义。在这个级别中任务类型为基本任务。每个优先级对应唯一的一个任务,每个任务只能一次激活。根据这个级别设计的操作系统适合于 MC9S12DG128 上应用程序的开发。操作系统服务 ActivateTask 或者 ChainTask 完成激活一个任务的操作,激活之后,任务就由初始状态转换到准备执行状态。为了提高效率,不支持优先级的动态管理方法,也就是由用户静态定义的优先级,在系统执行过程中是不能被修改的。

每个任务都由一个任务控制块(TCB)来管理,在任务管理模块中,实现 OSEK 标准 API:激活任务、终止任务、连接任务、调度、捕获当前运行任务 ID 和获得任务状态。资源管理模块实现 OSEK 标准 API:获得资源、释放资源。计数器管理没有标准 API,警报管理结构由警报和警报行为组成。其标准 API:获得警报信息、获得警报到期所需时间、设置相对警报、设置绝对警报和消除警报。事件管理模块实现的 OSEK 标准 API:等待事件、设置事件、消除事件和获得任务的时间状态。实现中断管理的标准 API:开/关所有中断和开/关第二类中断。

在资源管理中,OSEK OS 是通过采用优先级天花板协议(PCP)来避免优先级反转和死锁问题的。所以在系统生成期间,静态分配每一资源的最高限度优先级,使设计的 OSEK OS 执行 PCP 协议。这样,OS 中任务调度的原则就是:当任务不占有任何资源时,运行在原来的优先级上。若任务占有资源,每个占有资源的任务的优先级等于它所占有的所有资源的优先级中最高的那个。任务分配的规则就是不管什么时候任务请求资源都予以分配。

值得注意的是:MC9S12 系列单片机有中断自动保存寄存器的功能,所以在操作系统中断级别的任务调度时,并不需要在程序中手动保存任务上下文。

#### 5 实验验证与对比分析

我们在前述的汽车稳定性控制器硬件在环仿真平

台上基于所设计的 OSEK OS 进行软件编程,采用线性二次型最优控制算法,利用操作系统提供的任务管理功能分模块分任务编写控制软件,可以使程序思路更加清晰,更有效的利用系统硬件资源[6]。同时,中断管理,时钟管理等功能可以有效的增强系统的实时性和稳定性。

### 5.1 任务系统设计

汽车稳定性控制器的实时多任务平台设计如图 4 所示,共包含监控任务,控制任务,诊断任务,信号采集任务等 4 个部分[7]。

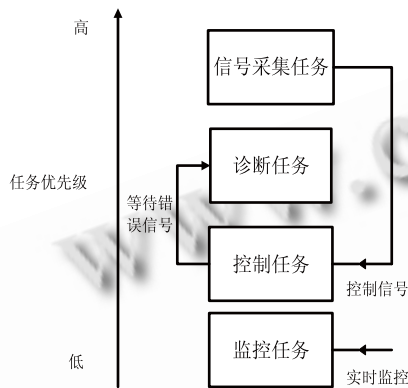


图 4 实时多任务平台

其中,监控任务负责整个控制器运行期间关键变量的实时监控,并与上位机进行通信。控制任务接收信号采集任务结束发送的控制指令激活,调用算法处理程序,输出相应的控制信号量。诊断任务由错误诊断信号激活,负责诊断系统级和算法级的错误,并向上报告。信号采集任务由定时器激活,进行模拟和数字信号的扫描和采集,并发送算法执行信号和控制指令。

### 5.2 仿真实验验证

仿真实验对有操作系统的控制平台和无操作系统的控制平台做了对比。仿真条件如下:

假设在水平路面上,车辆匀速前进,驾驶员给出一个阶跃的前轮转角信号。车辆参数为:

- 整车质量 1395 kg
- 质心至前轴距离 1.2 m
- 质心至后轴距离 1.35 m
- 车辆绕垂直轴转动惯量 900 kg\*m<sup>2</sup>
- 前轮侧偏刚度 24600 N/rad
- 后轮侧偏刚度 19000 N/rad
- 时间单位 5ms

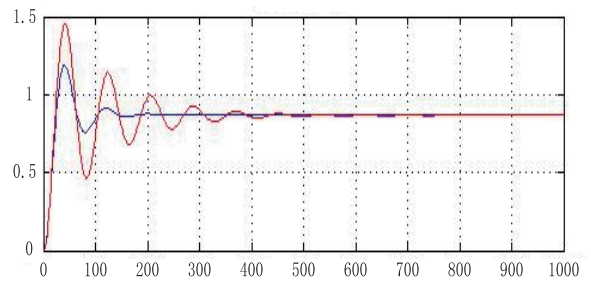


图 4 理想情况下横摆角速度控制仿真曲线(无操作系统)

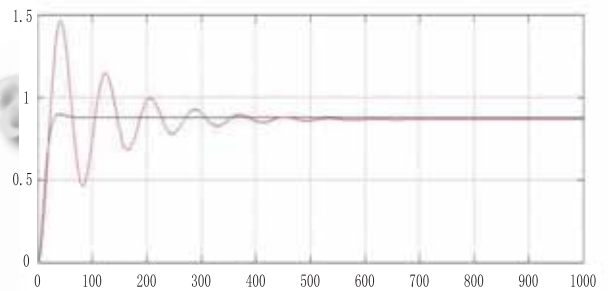


图 5 理想情况下横摆角速度控制仿真曲线(有操作系统)

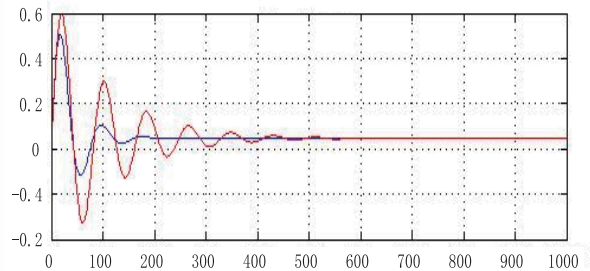


图 6 理想情况下质心侧偏角控制仿真曲线(无操作系统)

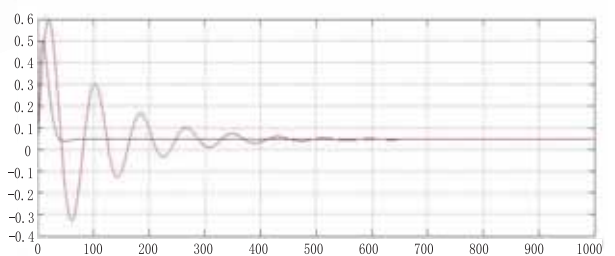


图 7 理想情况下质心侧偏角控制仿真曲线(有操作系统)

## 6 结语

图 4, 6 是在无操作系统的情况下采用传统方法的控制器的控制效果,控制信号在 150 时间单位左右达到稳定状态。图 5, 7 是在有操作系统的控制平台

(下转第 161 页)

(上接第 151 页)

上得到的控制效果，控制信号在 50 时间单位左右达到稳定状态。很明显，加载了操作系统的控制系统实时性得到了极大的提高。

实验表明：在 OSEK 嵌入式实时操作系统上开发的控制平台系统能可靠的完成硬件在环仿真中的信号采集，信号发生，制动控制等功能，控制效果及实时性良好，且程序具有可靠的稳定性。加载了实时操作系统之后，我们的控制算法只需经过很小的改动就可以移植到新的平台上，提高了控制软件的开发效率。

### 参考文献

- 1 张宝民,孙晓民.基于 OSEK 规范的嵌入式实时操作系统研究.计算机应用研究, 2004,4:32 - 35.
- 2 The OSEK/VDX Group. OSEK/VDX Operating Sys-

tem Version 2.2.3. <http://www.osek-vdx.org>, 2005.

- 3 黄鹏.基于 OSEK/VDX 的嵌入式车用操作系统研究.武汉理工大学学报, 2005,27(5):218 - 221.
- 4 李幼德,刘巍,李静,赵健,宋大风,沙宏亮.汽车稳定性控制系统硬件在环仿真.吉林大学学报(工学版), 2007, 37(4):737 - 740.
- 5 Labrosse JJ.邵贝贝等译.嵌入式实时操作系统  $\mu$ C/OS-II.北京:北京航空航天大学出版社, 2003.
- 6 张宝民,孙晓民.基于 MPC555 硬件平台的 OSEK 车用嵌入式实时操作系统设计与实现.中国计算机大会, 2003.
- 7 李秋华,张天宏,邓志伟.实时操作系统在发动机数字电子控制器中的尝试.航空发动机, 2006,32(2):53 - 55.