

# 基于 MDA 思想的嵌入式机电控制器设计框架

崔竞松<sup>1</sup>, 向永谦<sup>2</sup>, 彭 蓉<sup>3</sup>

(1. 武汉大学计算机学院, 武汉 430072; 2. 62101 部队自动化站, 武汉 430010; 3. 武汉大学软件工程国家重点实验室, 武汉 430072)

**摘要:** 嵌入式机电控制器的硬件耦合关系复杂, 软件受制于运行环境和开发环境, 由此导致该类系统的可移植性和可重用性较差。该文针对软硬件约束, 基于 MDA 思想提出了一种电动车控制器设计框架。该设计框架中的软硬件划分模块和各模块的功能定义使得此类嵌入式应用系统可以顺利地不同硬件平台和软件开发环境之间进行移植并根据用户需求定制。在该设计框架的指导下, 在 EM78P458、P87LPC768、ATMega8L 3 种硬件平台上实现了控制器系统的移植。

**关键词:** 移植; 重用; 嵌入式系统; 控制

## Embedded Mechatronic Controller Architecture Based on MDA Thinking

CUI Jingsong<sup>1</sup>, XIANG Yongqian<sup>2</sup>, PENG Rong<sup>3</sup>

(1. School of Computer Science, Wuhan University, Wuhan 430072; 2. 62101 Automatization Station, Wuhan 430010;  
3. State Key Laboratory of Software Engineering, Wuhan University, Wuhan 430072)

**【Abstract】** The portability and reusability of the embedded system are often suffered from the complex relationships between the hardware components and the constraints of the software running and development environments. An electronic motor controller architecture is proposed based on MDA. The division and definitions of the hardware and software modules in the architecture enable the smooth transplantation between different hardware platforms and software development tools. It also permits the customization. The author has transplanted the controller system between the three platforms: EM78P458, P87LPC768, ATMega8L under the guidance of the architecture.

**【Key words】** Portability; Reusability; Embedded system; Control

### 1 概述

随着芯片制造工艺和工业控制领域需求的不断发展, 嵌入式控制器已经逐步地在许多工业控制场合得到了广泛的应用。但是在一些控制逻辑相对复杂的应用环境中, 嵌入式控制器正在面临着越来越严峻的考验。主要的难度体现在: (1) 用户的应用需求多样化。为了满足用户的需求, 控制系统应具有可配置、可定制的能力; (2) 随着器件和生产工艺的进步, 嵌入式控制系统需要从旧的实现方案向新方案移植。这要求控制系统的设计方案应具有良好的可移植性。然而在运行资源有限、运行平台和开发平台差异较大的嵌入式系统中, 上述系统特性是难以实现的。

MDA<sup>[1]</sup>(Module Driven Architecture)是软件工程领域近年来的重要研究成果。MDA将系统的设计分为4个阶段: CIM(Computation Independent Model), PIM(Platform Independent Model), PSM(Platform Specialize Mode)和ISM(Implementation Specific Model)。其中, PIM给出了一个与平台无关的系统模型, 并可以利用模型验证工具在系统模型的级别对PIM进行验证, 分析其中可能存在的死锁、不可达状态等; PSM根据具体平台的特点, 对PIM进行特化, 得到与平台相关的特性化模型。因此, MDA方法使设计者能够独立于具体平台进行系统设计, 而最终又能够在某个具体平台中实现系统。同时, MDA不仅能够减少最终系统中可能存在的错误, 而且能使得该系统具有良好的可配置性和可移植性。MDA软件设计方法已经在现实的软件工程中得到了应

用。而在应用过程中, MDA方法的关键难点在于从PIM向PSM的转换。

本文参照MDA设计思想, 在对多种相关硬件环境设计经验的基础上, 提出了一种嵌入式机电控制器设计框架(以下简称框架)。该框架对嵌入式机电控制器的功能模块进行了科学的划分, 协调各模块之间的调用关系, 使得设计者能够首先独立于具体的硬件平台设计系统, 而后又能够平滑过渡到某个具体的硬件平台上实现, 并且有能力对产品工作环境的变化和产品本身电气特性的变化进行自适应的调节。

### 2 嵌入式机电控制器设计框架

嵌入式机电控制器(以下简称控制器)通常由软件部分和硬件部分组成。硬件部分一般负责数据采集、对外控制和数据交换; 软件部分一般负责执行算法和控制策略。软件部分与硬件部分是密切关联的。

因为软件部分与硬件部分在很多方面存在着差异, 所以本框架将控制器的设计分为硬件和软件两个方面。

#### 2.1 硬件设计约束

硬件部分一般指的是产品中有形的部分。包括包装外壳、电路板、器件以及组装等。

**基金项目:** 湖北省科技攻关基金资助重点项目(902130819); 软件工程国家重点实验室开放基金资助项目(SKLSE05-19)

**作者简介:** 崔竞松(1975-), 男, 博士、讲师, 主研方向: 算法优化, 信息安全; 向永谦, 硕士、工程师; 彭蓉, 博士、讲师

**收稿日期:** 2006-02-20 **E-mail:** cui\_js@sina.com

相对于研发成本而言,控制器的硬件部分的生产成本相对较高。因而在满足功能和质量要求的前提下,人们一般会愿意通过多投入研发成本,或者通过其它方式,获得更低的生产成本以提高利润。比较典型的方法是:通过多花时间、人力以及智力,设计出使用器件更少或更便宜的硬件方案。因此,硬件设计框架的一个主要设计目标是:遵循本框架进行设计,能够有效缩短研发周期,减少人力智力的投入,为降低产品生产提供便利。

硬件部分另外一个特点是部件之间的耦合关系复杂。虽然一般的分立元件的直接相邻部件数量不超过3个,但通常一个器件电气性能的变动会影响到整个功能模块的正常运转。如果电路中还有类似于运算放大器、自举控制芯片等需要同时与多个部件直接相连的器件,那么这些器件将会在电路中呈现出高介数<sup>[5]</sup>,成为系统的关键节点。虽然通过指定供应商、老化、测试等手段,可以在一定程度上降低这种不良影响发生的可能性,但是更重要而有效的手段是:固化已有的成熟的功能模块,从优化设计框架的角度,减小在新产品设计过程中发生遗漏和疏忽的可能性,同时降低研发成本。这种优化设计框架也是在MDA方法中从PIM向PSM转换过程中不可缺少的重要技术手段。

## 2.2 软件设计约束

某些控制逻辑比较简单的控制器中可能根本就不存在所谓的“软件”。因为利用现有市场上的分立元件和一些成熟的控制芯片就能够搭建起所需的控制电路,所以不需要设计软件。但随着应用的发展,利用分立元件和现有控制芯片,要想满足用户不断增加的需求越来越困难。实际上,在现有中、高档控制器中,不使用嵌入式软件的,基本上没有。

与硬件部分相比,软件部分的优点是:

- (1)具有计算存储能力,较容易实现复杂的时序逻辑控制;
- (2)设计规范严格,减小了设计失误的可能性;
- (3)调试较方便,调试周期较短;
- (4)模块可重用性强,易于相互搭配;
- (5)生产成本低。

由于相对于软件研发成本而言,软件的生产成本很低,一般忽略不计,因此在不影响软件品质的前提下,设计者通常注重于降低软件的设计成本。软件成本与硬件成本是密切相关的。现实中不乏用提高软件设计成本谋求降低硬件生产成本的例子。比如通过选择设计语言,改变编程风格,减小软件的空间需求,从而降低软件载体的价格;又如通过软件滤波、非线性处理等技术,降低对硬件传感器的要求,从而降低了传感器的价格,简化了传感电路等。

在拥有优势的同时,软件也存在着局限性:

### (1)运行环境的差异

嵌入式软件的运行通常需要依赖于嵌入式硬件平台,而嵌入式系统(硬件)平台的种类非常多。从功能强大的ARM系列,到兼顾速度和成本的Motorola的56XX、TI的TMS320C等系列DSP,到廉价的51、AVR、PIC等系列单片机,还有更底层而高效的门阵列芯片,如Altera、Lattice、Xilinx产的各种CPLD/FPGA芯片。常见的嵌入式硬件平台就有上百种。

虽然现在几乎每个嵌入式硬件平台技术都较为成熟,运行也比较稳定,但却缺乏统一规范。仅在机电控制器的一线主控芯片所经常采用的单片机和DSP范围中,各硬件平台系列从指令集到字长都有区别,而且每个系列之中,因为厂商、型号的不同,还会在硬件资源等方面存在着许多差异。

所以一般的嵌入式软件只能够被设计成为在某个型号的嵌入式硬件平台上运行,而不能像PC软件那样,可以在几乎所有的PC上运行。如果一个嵌入式软件希望移植到另一个硬件平台上,那么从软件体系结构到具体实现方案都将接受考验。

### (2)开发环境的差异

与硬件开发环境不同,软件的开发环境是非常多的。虽然某些嵌入式软件开发环境试图支持尽可能多的开发语言和硬件平台,但可惜的是,即便是目前比较好的开发环境也只能支持某个系列的硬件平台。在开发语言方面,除了汇编语言是默认被支持的以外,开发环境很少支持C语言以外的其它高级语言。而且即使是对C语言的支持也会因为硬件平台的各种限制而不全面。另外各种开发环境在标准C语言所规定的语句语法的基础上,都根据其对应的硬件平台的特点进行了扩展。然而,即便是支持同一个硬件平台的多个开发环境所进行的这种扩展也各不相同。

所以,虽然软件的模块较容易划分,接口也比较规范,但是要想让一段已经在应用过程中被验证为正确的代码能够顺利地地为以后的开发服务,降低开发成本,也并非易事。只有对相关应用领域的各种硬件平台和各种开发环境有了充分的了解,才能够在编制软件时合理地划分功能模块,设定各模块的工作方式以及数据传递接口。对于实时性要求高或者资源非常紧张的应用环境,还需要额外注意各模块代码的大小和在工作时间。

综上所述,为了能够使设计开发的嵌入式控制系统能够在复杂多变的应用环境中具有良好的自适应能力和可移植性,本文基于MDA方法的思想,在多种相关硬件环境的设计经验的基础上,提出了一种针对电动车控制器的设计框架。

## 2.3 设计框架

包括OSACA、OMAC、OSEC在内的多种国际流行的开放系统框架<sup>[2-4]</sup>分别对此领域的控制系统给出了标准或参考。

设计框架宏观上可分为硬件和软件两部分。

硬件部分从物理上通常包括印刷电路板,各种分立元件,控制芯片,接口等。从逻辑功能上看,电动车控制器硬件设计工作将涉及到如下部分,如图1所示。

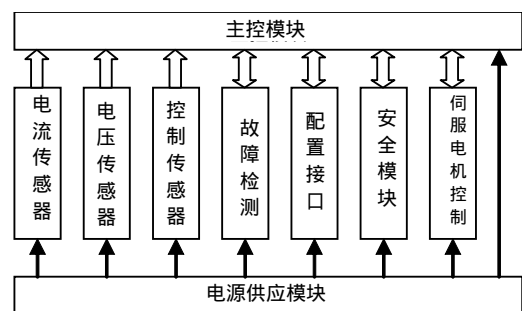


图1 电动车控制器硬件模块划分

(1)电源供应模块 此模块保证整个控制器的各部分都能够得到所需指定电压的稳定电源。此模块通常由一系列稳压电路构成;

(2)主控模块 此模块负责对控制器实施最高层的控制,也是软件的载体。此模块通常由单片机来承担,有时需要辅助芯片;

(3)电流传感器 此传感器负责对通过电机的电流进行检测,并将检测信号传达给主控模块。此传感器通常是通过检

测元件压降来实现的，有时需要信号放大；

(4)电压传感器 此传感器负责对控制器的电源进行检测，并将检测信号传达给主控模块。此传感器通常可以通过分压来实现；

(5)控制传感器 此传感器负责感知用户的意图，并将检测信号传达给主控模块。此传感器通常是霍尔传感器、微动开关等；

(6)伺服电机控制模块 此模块在主控模块的指挥下，控制伺服电机运转；

(7)故障检测模块(可选) 此模块负责检测控制器、电机以及车辆的易损件。如果发生故障，需要报告给主控模块；

(8)配置接口模块(可选) 此模块负责将主控模块与控制器外部的上位系统连接，使上位系统能够对主控模块进行配置和分析。此模块通常由通信转换电路构成；

(9)安全模块(可选) 此模块负责保护控制器不被非法使用和非法复制。

设计框架的软件部分将完成控制器所有上层控制，它的主要硬件载体是主控模块。由于控制功能更加复杂，所以软件部分所涉及的功能模块较多。本框架将这些功能模块分为3个层次，如图2所示。

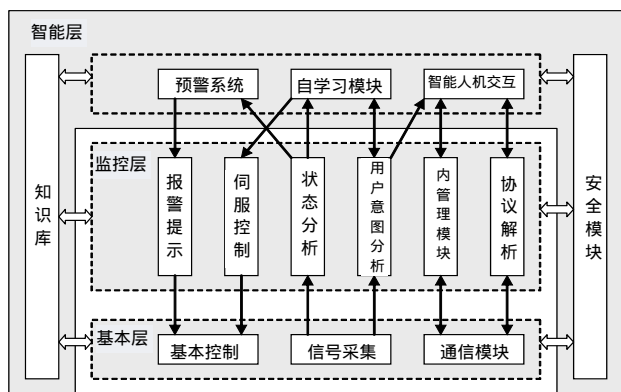


图2 电动车控制器软件模块划分

### (1)基本层

基本层主要负责与硬件平台相关的输入输出功能，包括以下模块：

1)信号采集模块 负责采集各种来自主控模块外部的输入信号，包括各种模拟信号和数字信号。此模块通常需要具备容错滤波，然而容错能力和灵敏度存在着一定的矛盾。此模块的滤波算法需要根据容错能力、响应速度、采样精度、采样口的数量和其它硬件资源限制条件而设计，才能达到最理想的采样效果；

2)基本控制模块 负责控制主控模块对外的输出信号，包括低频输出和高频输出。此模块对主控模块硬件平台进行了一次逻辑封装，对设计框架中的更上层模块提供了更统一、更数学化的控制接口；

3)通信模块(可选) 负责通信协议解析，实现主控模块的内外数据交换。此模块对于配置控制器、在线升级及调试都是至关重要的。

### (2)监控层

监控层主要负责与硬件平台基本无关的管理和控制功能，包括以下模块：

1)内管理模块 负责主控模块内部的基本维护功能。包括自检、WatchDog、系统时钟、中断管理、数据一致性维护；

2)用户意图分析模块 负责从相关数据信号中分析得出用户的意图。用户的意图可能包括运动的目的地或方向、车辆的速度、加速度或驱动力等；

3)状态分析模块 负责分析由信号采集模块所采集到的所有数据，并从中得出目前系统的工作状况。系统工作状况可能包括：

车体状况 车辆当前的位置、角度、姿态等；

电机状况 电机的转速、相位、电流等；

电池状况 电池的电压、电量、充放电电流等；

控制器状况 控制器本身的温度、配置信息等；

故障状况 整车当中的某些易损件的工作状况，容易发生错误的配置开关等；

环境 环境温度、路面状况等；

4)报警提示模块 负责用声、光等形式向用户发出不同的报警或功能提示；

5)伺服控制模块 负责根据用户意图和当前系统状况，通过基本控制模块对车辆的运行进行调整，并实施必要的各种保护功能。控制对象包括：伺服电机、开关等；

6)协议解析模块 负责完成对通信数据格式的解析、封装和纠错。

### (3)智能层

智能层主要负责复杂算法和功能，包括以下模块：

1)自学习模块 通过对过往历史数据进行分析整理、数据挖掘，发现在设计阶段和生产阶段所不容易或无法得到的重要信息，以调整未来的控制参数，优化控制效果，使用户达到更佳舒适度；

2)知识库 为全系统提供算法、控制模式和经验参数，以满足控制器在不同应用环境下的需求；

3)智能人机交互 通过对用户行为的分析，发现用户的操作习惯和潜意识意图，可为用户提供更全面更贴心的服务。除了最终用户以外，生产、检测、维修人员也是经常与控制器交互的用户。为他们提供良好交互界面将有助于降低他们的工作量和定位错误的时间；

4)预警系统 在为用户提供足够的健壮性和容错能力的同时，还应及时地对系统中可能发生的问题进行预警；

5)安全模块 为保障厂商和用户的合法权益，安全模块将与硬件部分中的相应模块进行配合，使得非授权用户不能使用控制器，使得非原厂厂商无法复制控制器。

## 3 框架应用

在上述设计框架的指导下，笔者先后在 EM78P458、P87LPC768、ATMega8L 3 种单片机硬件平台上，分别使用 CWice、Keil uVision、CodeVision、AVR studio 等多种开发环境，结合相应的调试工具，利用一套可配置、可重用的 C 语言源代码，在相应的硬件电路板上，为不同功能和功率需求的应用环境，开发了 10 余种不同款式的电动车控制器。由于篇幅有限，不再具体举例。本研究成果亦可应用于军车平稳行驶的控制，将有助于军车在移动过程中对目标的瞄准操作。

## 4 结论

在将一套硬件电路设计和相关软件从一个应用环境向另一个移植的过程中，合理有效的软硬件模块划分方案是至关重要的。本文在 MDA 思想的基础上，针对嵌入式机电控制器系统在移植和重用过程中所受到的来自于软件和硬件方面的制约，设计了一种系统设计框架，给出了软硬件的模块划分和功能定义，最后给出了在该设计框架指导下完成的两个应用实例。利用本框架实施设计与开发能够有效地提高系统的可重用性和可移植性。

## 参考文献

- 1 Oliver I. Applying UML and MDA to Real Systems Design[C]. Proc. of the Conference on Design, Automation and Test in Europe, 2005: 70-71.
- 2 Nacsas J. Comparison of Three Different Open Architecture Controllers[C]. Proc. of IFAC, MIM, Prague, 2001-08-02.

(下转第 272 页)