

组态化嵌入式软件仿真系统设计

姬 祥, 董云卫, 王宇英, 凌 恒

(西北工业大学计算机学院, 西安 710072)

摘 要: 为缩短开发周期, 提高嵌入式软件仿真系统的设计水平, 提出一种组态化嵌入式系统设计。集成仿真建模工具、仿真模型代码生成工具, 实现仿真运行环境搭建和对仿真数据的实时监控, 采用组态化的仿真方法, 根据用户需求生成监控界面。实验结果表明, 该系统的运行平台快速高效, 监控环境良好。

关键词: 嵌入式软件; 实时监控; 组态软件; 仿真系统

Design of Configuration Embedded Software Simulation System

Ji Xiang, Dong Yun-wei, Wang Yu-ying, Ling Heng

(School of Computer, Northwestern Polytechnical University, Xi'an 710072, China)

【Abstract】For shorten the developing period of embedded software and improve design standards, a configurable software approach for simulation system is proposed in this paper. It includes modeling tool and code generation tool of simulation. Simulation is implemented to customize simulation environment. It can create monitoring interface on the basis of user need. Experimental results show that, the configurable simulation system developed by this approach provides a fast and efficient way for embedded software simulation.

【Key words】 embedded software; real-time monitor; configuration software; simulation system

DOI: 10.3969/j.issn.1000-3428.2011.21.077

1 概述

嵌入式系统软件、硬件开发紧密相关, 而硬件平台开发成本昂贵、开发周期长, 若软件功能调试和性能测试不能及时完成的话, 软件质量无法保障。随着嵌入式软件的规模增大, 系统的结构日益复杂, 如何快速、低成本开发出嵌入式软件, 而又能保障软件的质量, 成为嵌入式系统设计和开发面临的问题。利用仿真技术, 可以快速构建嵌入式软件运行所需的硬件环境和应用的外围环境。嵌入式软件仿真开发平台是支持嵌入式软件开发的一种快速、高效和低成本的工具, 对提高嵌入式软件的开发效率和质量具有重要意义。

模型的重用性使用能够最大限度地使用已有模型, 以避免重复性建模, 达到减少模型开发成本的目的^[1]。构建一个面向嵌入式软件应用的仿真系统, 使其具有通用性和开放性, 使其适用于不同的仿真平台。仿真系统设计完成后, 具有针对专有领域和特定实验环境的特点, 一旦仿真模型或外部环境发生变化, 需要修改仿真监控系统。而内部系统或者监控条件等地频繁变化, 给嵌入式软件开发人员造成了不便。构造一个嵌入式仿真开发系统, 使其集模型开发、过程控制、实时监控和数据处理于一体, 以面向不同的应用领域, 能快速地适应不断变化的仿真模型和监测条件, 是需要研究解决的问题。组态软件伴随着集散式控制系统(Distributed Control System, DCS)的出现而发展并逐渐得到广泛应用。组态软件为自动化工程技术人员提供了一种采用搭积木的方式制作现场控制过程和控制界面的工具^[2]。基于以上内容, 本文采用模型驱动架构(Model Driven Architecture, MDA)的设计思想, 提出组态化嵌入式软件仿真系统设计。

2 仿真系统设计

仿真技术是一种基于模型的技术, 其可以通过仿真建模

快速构建嵌入式软件运行的仿真环境及目标原型。模型驱动架构将建模语言用作一种编程语言, 其在模型软件开发中是非常重要的。模型建立的正确与否直接关系到仿真结果的可信度, 只有模型的正确, 其仿真结果才具有实际的应用价值和意义^[3]。

工程问题通过抽象和简化, 确定变量和参数之间的关系, 可建立数学模型来更方便的解决实际问题。根据建模所依赖的方法不同, 可采用数据建模方法(Simulink)或行为建模方法(Stateflow)。模型建立好后可直接进行模型仿真, 根据仿真结果修改模型。

尽管目前出现的某些大型仿真系统已经基本实现上述功能, 但是仍然存在一些缺陷:

(1)绝大多数的仿真系统因为没有提供完善的仿真过程监控功能, 而为仿真模型的调试与验证带来诸多不便。

(2)在建立仿真模型时, 由于仿真系统模型库中缺乏某些组件为用户的建模和仿真调试带来不便^[4]。

(3)用户在进行半实物仿真时, 由于系统中无法提供相关板卡驱动组件, 因此每进行一次半实物仿真, 用户都需要自行编写一次板卡驱动, 一旦使用者缺乏板卡驱动编写经验, 就必然需要花费大量时间学习程序知识, 故造成整个仿真的过程阻滞。

模型驱动的嵌入式软件仿真流程可以概括为:

(1)使用建模语言生成模型。

基金项目: 国家自然科学基金资助重点项目(60736017); 国家“863”计划基金资助项目(2009AA01Z147)

作者简介: 姬 祥(1985—), 男, 硕士研究生, 主研方向: 嵌入式软件仿真; 董云卫, 教授; 王宇英, 讲师; 凌 恒, 硕士

收稿日期: 2011-05-30 **E-mail:** jixiang@yahoo.cn

- (2)将模型转化为高级语言代码。
- (3)高级语言代码被编译成目标代码,下载到目标机上。
- (4)目标代码在实时操作系统中运行。
- (5)实时操作系统驱动嵌入式硬件。
- (6)仿真运行,调试及验证。

嵌入式软件仿真系统开发流程如图 1 所示。

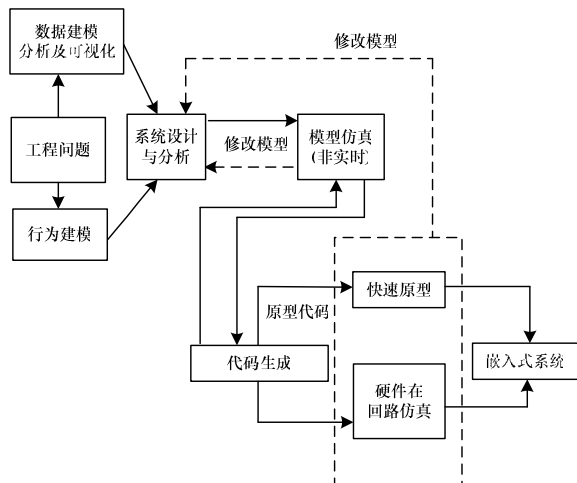


图 1 嵌入式软件仿真系统开发流程

针对上述问题,在系统一体化理论的基础上,设计集建模、仿真和分析处理等功能的通用一体化系统平台。根据嵌入式软件开发流程,仿真工具满足以下功能:

- (1)能够为代码的运行提供硬件支持,具备构建快速原型及环境系统的能力。
- (2)具备从模型到代码的自动生成能力。
- (3)提供仿真过程控制功能,能够加载编译仿真模型,并能控制仿真,即启/停。
- (4)提供组态化仿真监控界面配置功能,能灵活配置仿真监控界面。
- (5)提供仿真数据监控功能,能够直接通过宿主机对仿真机所运行实例的输出进行监控,即实时观测仿真动态。
- (6)提供通信配置功能,为宿主机与目标机的通信连接提供多种选择。

3 系统功能及结构设计

嵌入式软件仿真系统软件设计框架如图 2 所示。

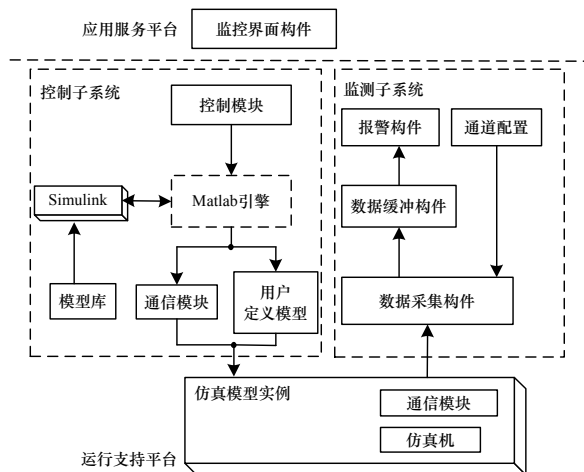


图 2 嵌入式软件仿真系统设计框架

由图 2 知,嵌入式软件仿真系统的硬件部分由主控计算机、仿真计算机、信号调理设备和被控对象等组成。软件部

分集成仿真建模工具和仿真代码生成工具,实现仿真运行环境的定制、仿真监控等功能。嵌入式软件仿真工具是仿真系统结构中抽取出的功能相对独立的构件,如界面配置,数据采集、缓冲和报警等构件,通过在一定框架下,将这些构件组合,来形成仿真应用的软件。构件之间通过标准接口相互通信,这样便实现操作命令的传递和数据共享。底层为运行支持平台,而上层为应用服务平台。软件系统划分为 3 个子系统,即组态监控界面子系统、控制子系统、监测子系统。

图 2 的相关说明如下:

(1)Simulink 是 Matlab 的一个集成工具箱,是一种高度图形化的建模仿真工具,用来对动态系统进行建模、仿真和分析。各子系统模块采用分层结构,通过模块之间的组合完成系统功能的实现,即可以方便地通过更改、替换子模块,修正系统。这种灵活的仿真建模语言还支持外部模式下修改仿真参数,这为在线仿真时模型的调试和修改提供了支持。

(2)监控界面构件实现实时监控的功能。用户将仪表、示波器和数据表等控件,通过将其添加到监控界面,进行参数配置,完成将接受到的实时数据或历史数据显示到运行着的监控界面的功能,并且获得动态化监控效果。

(3)数据采集构件,将仿真运行获得的实时数据进行必要的加工,并将其发送到主控机的数据缓冲区中。数据采集构件通常有以下 3 种实现方式:

- 1)双机系统中主机和从机之间的通信。
- 2)以太网、串口或网络实现的 C/S 架构,其中,Server 是数据提供方;Client 是数据访问方。
- 3)基于 Internet 的 Browser/Server 应用。
- (4)报警构件,对缓冲区接收到的每组变量,根据已经设置好的报警界限做出判断,若在报警界限以内,则继续运行;若超出报警界限,则根据设定方式发出警告信息或者终止仿真执行。

(5)通道配置构件与监控界面相关,通过一定的绑定(对应)规则将每个控件与模型的各种输出数据进行对应。

(6)数据缓冲构件,将缓冲区的数据进行选择和处理,以图形的方式动态显示在计算机屏幕上,或者以 XML 的方式实现数据转存到数据库的功能。

采用以上思想设计和开发的仿真系统,可以满足嵌入式软件从开发到运行和测试的各阶段需求。同时为系统的扩展性,支持开放标准和适应不同实时仿真平台提供保障。

4 组态化仿真系统

组态软件为搭建满足用户需要的仿真环境,提供了理论支持和过程支持。多任务、实时性是该组态化嵌入式软件仿真系统的主要特点。在多任务环境下,将功能相对独立的任务以组件的方式实现,各组件在配置好参数后,可进行数据交换。把不需要立即做出反应或对实时性不产生损害和影响的事件,作为顺序执行的任务;对保存时间很短,需要计算机立即做出反应的事件作为中断请求或事件触发信号,对其专门编写程序,以便计算机对这类事件做出快速的响应。

组态软件一般是由系统开发环境和系统运行环境构成。系统开发环境是进行应用程序系统生成工作所必须依赖的工作环境,由若干组态程序组成,如图形界面组态程序、数据通信组态程序等。系统运行环境下,目标应用程序被装入计算机内存并投入实时运行。该嵌入式软件仿真工具,集开发环境和运行环境于一体。如图 3 所示,Host 为开发环境提供了可定制仿真环境和个性化监控界面的能力,运行环境由

Target 完成, 是仿真程序运行所依托的外部条件, Target 通过串口或以太网与 Host 进行实时数据的传输(此处的 I/O 驱动、模型代码等模块均运行在仿真内核之上, 其为一个高效的实时 OS)。

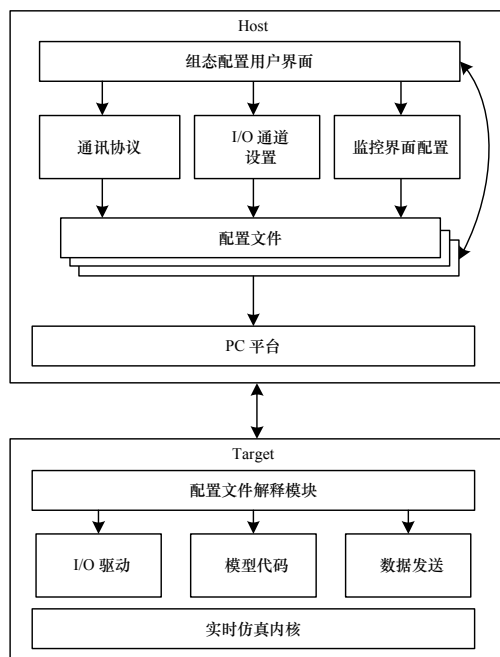


图3 组态化仿真工具开发和运行环境

由图3知, 在 Host 中, 组态配置用户界面根据需要定义仿真组件相互之间数据交换的界面控制元素, 系统通过此界面向仿真应用发送现场数据, 仿真应用接收这些数据, 通过对不同功能模块进行参数配置, 完成为用户针对具体的仿真环境和数据监控需要提供可定制的能力, 这些模块包括通信协议模块, I/O 通道模块和监控界面模块。界面元素采用事件驱动和多线程控制机制。

通信协议模块提供通信方式的选择和通信协议的配置。I/O 通道设置模块是板卡应用有关的单元。在半实物仿真中, 仿真程序和外部信号需要板卡的调理和转换, 才能进行数据交换。对板卡进行配置, 选择采样率和调节转换电压, 并选择工作通道, 以完成 D/A、A/D、DIO 的变换。。

Target 为配置数据解释模块, 负责将接收到的配置文件进行解释, 并送给 I/O 驱动模块、模型代码或数据传输模块。I/O 驱动模块接收对 I/O 板卡应用的配置参数数据, 根据设置内容完成数据的 D/A、A/D、DIO 转换, 模型代码是运行的仿真模型程序, 可以接收仿真控制命令和在线修改仿真参数, 数据发送模块在设置好通信协议后, 将仿真数据实时发送到 Host 中。

定制好仿真环境和监控界面后, 模型代码就可以运行在 Target 中, 仿真数据实时发送到 Host 中, Host 发送控制命令到 Target, 以此实现仿真过程的控制。

5 仿真工具的实现

程序可靠和稳定的运行, 是开发组态软件所要达到的目标。仿真工具的实现采用 Visual C++, 其是一种常见应用程序开发环境, 特点是代码运行效率高、运行速度更快、实时性更好。应用框架采用微软的 MFC 工具, 其具有功能全面、性能稳定的优点。

为接收到的每一组仿真数据建立一个缓冲区, 用于监控显示的进程从缓冲区读取数据, 然后传递给对应的监控组

件, 最后由监控组件完成显示和更新, 以此完成示波器控件、仪表控件和数字表格控件组态化配置的实现。其中, 示波器控件参数配置实现类如下:

```
Class Scope:Public MonitorComponent
{private:
    CString name;
    RECT size;
    Pos position;
    Axis x_axis;
    Axis y_axis;
    int num;//numbers of variable
public:
    void NameChange(CString);
    void SizeChange(Rect);
    void PositionChange(Pos);
    void AxisChange(Axis,Axis);
    bool Refresh();//refresh GUI of Monitor}
```

配置参数是组态软件实现的关键, 它将系统开发环境和系统运行环境联系起来。参数配置通过图形化界面进行, 配置参数以 XML(Extensive Makeup Language)的方式记录并保存, 如图4所示。

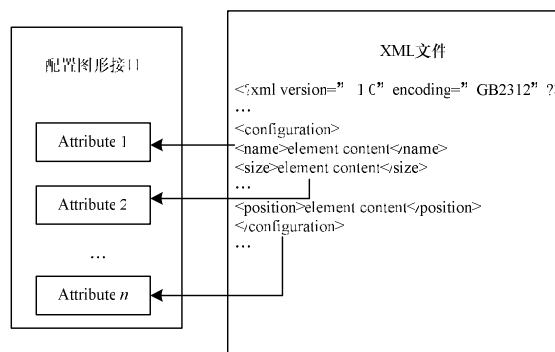


图4 组态化参数配置实现

用于实时监控的图形化显示控件采用 ActiveX 控件的技术, 可以很方便地应用到 VC++开发的程序中。仿真工具还集成了 LabVIEW 提供的专业化虚拟仪表和示波器显示控件。配置参数通过监控组件类的函数调用, 将配置好的参数映射到图形化显示控件对象属性值上。

Matlab 引擎, 是一组 Matlab 提供的接口函数, 供用户在其他编程环境对 Matlab 控制的实现^[5]。仿真工具通过调用 Matlab 引擎, 集成了仿真建模和模型代码编译链接的功能。文件 enging.h 中, 声明了引擎的 API 函数说明和所需数据结构的定义。

RTW(Real_Time Workshop)为基于 Simulink 的代码自动生成环境, 其可以将模型自动转换为代码, 在硬件上运行动态系统的模型, 并且支持基于模型的调试。目标机运行 xPC 实时系统, 它是一个基于 RTW 体系框架的主从式结构的实时仿真技术的附加产品。仿真工具集成代码自动生成功能, 在 Simulink 实现建立模型之后, 通过调用 RTW 工具将模型编译成可执行的 C 代码, 并下载到目标机上运行。

6 工作过程和应用实例

使用该仿真工具的工作流程如下:

- (1) 用户针对目标系统建立仿真模型。
- (2) 设置通信协议, 并将编译链接后的仿真代码加载到仿真机; 根据使用需要, 对使用的板卡 I/O 通道进行参数配置。

(下转第 231 页)

