

基于嵌入式组件技术的图形软件设计方法

马伟民

(杭州电子科技大学 智能与软件研究所, 浙江 杭州 310018)

摘要:针对嵌入式裸机的图形界面系统(GUI)开发效率低,以操作系统为平台的系统开发资源要求高等问题,提出了一种新型嵌入式开发模式 CCS 组件模型。在开机时动态加载系统所需组件模块以构造系统,实现了图形软件的自动生成。实验结果表明将该方法应用于单色人机界面(HMI)的设计,具有简便、效率高,目标系统灵活配置等特点。

关键词:组件;动态加载;嵌入式系统;图形界面系统;人机界面

中图分类号:TP391.72

文献标识码:A

文章编号:1001-4551(2010)05-0098-04

Design method of graphics software based on embedded technology component

MA Wei-min

(Institute of Intelligent and Software Technology, Hangzhou Dianzi University, Hangzhou 310018, China)

Abstract: In order to solve the problems that the development of embedded graphical user interface(GUI) is inefficient based on the bare system platform, and need more resources based on the operating system platform, a new embedded development model called components configure system(CCS) was presented. The system was constructed when loading the required components modules at the boot-time, the automatic generation of the GUI was realized. This model was applied to design monochrome human machine interface(HMI). The experiment results show that using this method to develop system is simple and efficient, and it can flexibility configure the target system.

Key words: components; dynamic loading; embedded system; graphical user interface(GUI); human machine interface(HMI)

0 引言

在嵌入式环境下开发图形界面软件时,为了满足多任务协调、响应速度、开发周期等因素,引入嵌入式操作系统是一种较好的解决方法,目前大部分嵌入式图形界面也都采用该方法进行设计^[1]。但这种开发方式增加了代码复杂度,需要更多的嵌入式资源(如内存空间),在不扩展外围设备的情况下,中低档单片机很难达到要求^[2-3]。

目前嵌入式图形界面开发大都采用传统的基于代码的程序设计,该方法在编写复杂的界面时往往显得力不从心,并且容易出现错误^[4]。目前主要流行的嵌入式 GUI 系统有 OpenGUI、Qt/Embedded、MiniGUI 和 MicroGUI。但这些系统也都有各自的缺点,如 OpenGUI 内核用汇编语言实现,可移植性较差;Qt/Embed-

ded 是基于 C++ 类库的,因此其资源消耗较大^[5]; MiniGUI 和 MicroGUI 在窗口系统和图形接口方面的功能比较欠缺^[6-8]。

基于以上问题,本研究提出一种能在裸机环境下进行嵌入式图形界面开发的组件模型,称为 CCS 模型。在启动前通过该模型加载所需组件,实现系统的可重构,可以大大降低系统所需的内存资源,并提高开发效率,实现资源的灵活配置。

1 CCS 组件模型

1.1 CCS 组件分层结构

根据嵌入式系统的应用领域,所要求的图形软件底层设备也不同。

CCS 组件模型从上到下分为应用层、内核层、驱动

层和硬件抽象层,如图1所示。应用层组件是系统向用户提供应用服务的惟一窗口,主要用来解析界面数据,并产生相应服务,如串口通讯、GUI等;内核层组件主要包括各种部件显示方法和各种事件,不同的触发事件响应不同的事件驱动;驱动层组件用来驱动硬件系统中的各种设备,如LCD、串口、键盘等;硬件抽象层组件,是系统与硬件电路之间的接口层,用来将硬件抽象化,方便驱动层组件开发。

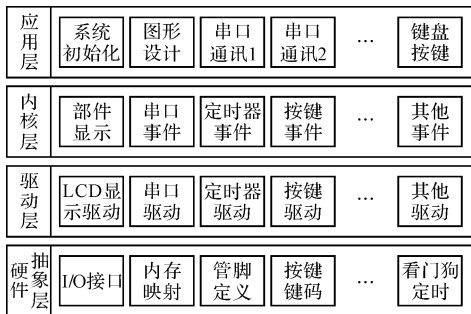


图1 CCS组件的分层结构

每个组件内部分为输入接口、输出接口、内部逻辑3部分,输入接口可以没有,也可以有多个,但输出接口最多只有一个,并采用C语言形式表述。组件与组件之间可以调用,但调用时必须保证接口的一致性。

为了解决系统的安全性问题,模型规定:只有上层组件才有权调用下层组件,且不能越层调用。如内核层组件可以调用驱动层组件,但不能调用硬件抽象层和应用层的组件。各种组件服务都有其固定的内存区作为数据交换空间。

CCS组件采用XML语言进行描述,具有设计简单、结构清晰、可扩展性强等特点。每个功能从硬件抽象层到应用层都有其相应的XML节点进行描述。如LCD液晶屏显示功能,在硬件抽象层有I/O接口节点对LCD和MCU之间的管脚连接关系进行描述;在驱动层有LCD驱动函数库节点进行描述;在内核层有LCD画点、画线等各种画图节点进行描述;在应用层有画图片、画字符等节点进行描述。

1.2 CCS组件设计流程

CCS组件设计过程如图2所示。整个过程包括以下4个步骤:

(1) 组件设计。根据实际需求设计符合要求的组件,设计完成的组件保存到组件库中。在整体设计过程中,只有当组件库中没有所需组件时才需进行该步骤。

(2) 引擎构造。通过组件配置软件构建目标系统的引擎,CCS组件系统将基于这些配置信息构建引擎。

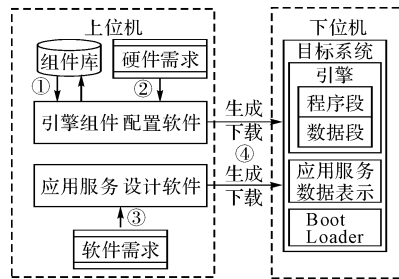


图2 CCS设计过程

引擎构造设计过程可以从硬件抽象层开始从下到上配置,也可以从应用层开始从上到下配置。为了提高系统的稳定性,引入了查错机制,在构造过程中不断进行逻辑判断,保证引擎正确。

(3) 应用服务设计。应用服务设计主要包括界面设计和控制逻辑设计两个部分。通过应用服务设计软件完成设计后自动生成事先约定好的二进制数据格式。为了方便引擎解析,该数据格式除了包括图像界面和控制逻辑数据外还包括索引结构表。

(4) 在启动时加载组件。目标系统中配置有Boot Loader(启动引导)程序,开机后可根据需求加载引擎镜像或者应用服务数据,加载的方式是通过串口进行传送。加载完成后,下位机引擎程序将通过应用服务数据中的索引结构表,快速检索到所需要的数据信息,并通过该信息产生相应的动作。具体加载过程详见下一节。

2 CCS组件技术

由于资源的局限性和实时性要求,在嵌入式系统设计过程中需解决两个重要问题:如何高效管理内存资源和如何实现实时的人机交互。

为了提高内存利用率,将下位机系统的Flash分成了引擎区、应用服务数据区和Boot Loader区3块,各块都具有固定的起始地址和大小。Boot Loader程序主要进行引擎区和应用服务数据区的动态加载,在开机后自动运行。在引擎和应用服务数据下载到下位机系统目标位置后,引擎将开始解析应用服务数据,完成图形界面生成。由于在不同的应用中各个分区所需空间不同,在引擎组件配置软件中加入了一个内存管理组件,该组件可以根据不同的应用,动态调整各分区的大小。

为了提高系统的实时性,本研究在引擎解析应用服务数据的过程中加入图形索引组件。该索引组件可以高效地在应用服务数据中查找所需图形信息,加快引擎生成图形界面的速度。索引组件在设计过程中需

着重考虑以下两方面:①索引结构复杂度应适中,否则引擎将花大量的时间用于解析,降低了程序运行速度;②索引表所占用的内存应尽量小。

图形索引结构如图 3 所示,最高为三级索引。为了压缩索引表的空间,索引表中只包括各索引项指向的地址指针。引擎在内存中只保存了一级索引表,每次查找数据,都必须先从 Flash 中装载二级、三级索引表,并在完成后释放。这样虽然降低了引擎的速度,但大大降低了 RAM 使用量。

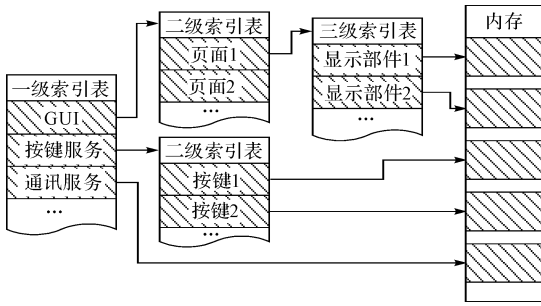


图 3 图形索引结构示意图

3 实验及结果分析

本研究将 CCS 应用于单色人机界面(又称为文本显示器)的设计中。单色人机界面是一种用来连接可编程控制器(PLC)、变频器、直流调速器、仪表等工业控制设备,把通过通讯接收到的数据流信息进行解析后,以文字、数字、图形和动画的方式在显示屏上显示的设备。

系统的上位机设计平台主要包括两个部分:引擎组件配置平台和应用服务图形界面设计平台。引擎组件配置平台提供了 100 多种不同类型的组件可供配置,如:不同的 LCD 型号、不同的串口通信协议、不同的键盘模块、实时时钟模块、EEPROM 模块等。该平台通过简单的鼠标操作实现系统配置后将自动生成相应的 C 语言代码,通过调用编译器,即可生成下位机可执行的文件。该软件平台界面如图 4 所示。

应用服务图形界面设计软件通过简单的图形控件拖拉,实现“所见即所得”的图形界面设计。图形界面中的所有图形控件,在引擎组件配置平台中需有相应的组件库支持。通过该平台设计完成图形界面后,自动生成前文所描述的数据结构。应用服务图形界面设计软件的界面如图 5 所示。

引擎系统是根据硬件环境的需求设计的;而应用服务图形界面和硬件无关。因此在设计系统时,引擎系统只需一次设计完成即可,而应用服务图形界面可



图 4 引擎组件配置平台软件界面

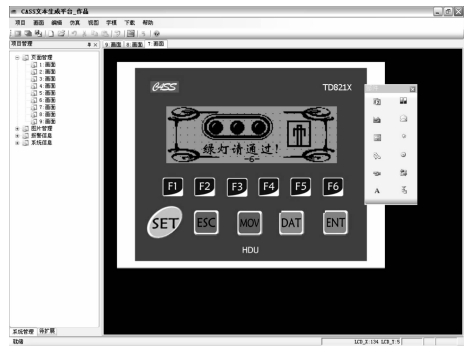


图 5 应用服务图形界面设计软件界面

以根据软件需求的不同而随时改变设计。

下位机基于 Atmega128 芯片设计,其主频为 7 372 800 Hz,驱动液晶屏为 192 × 64 点阵的单色 LCD,并配置了 4 × 4 的按键键盘和 I²C 实时时钟模块。实际运行效果如图 6 所示。

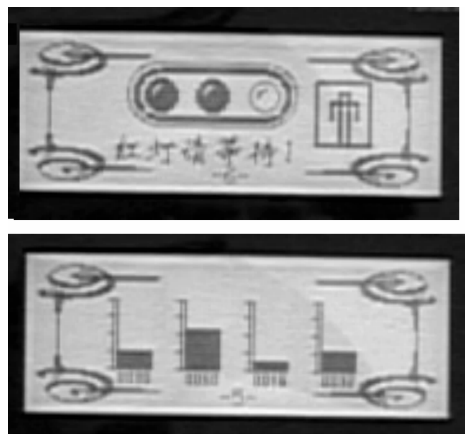


图 6 图形界面实际运行效果

为了能够清楚地了解通过 CCS 组件设计的单色人机界面效果,分别采用 CCS 组件设计方法和传统嵌入式设计方法,设计了相同的人机界面。该图形界面共有 8 个页面,平均每个页面有 6 个图形组件。其设计结果如表 1 所示。

表1 设计结果比较

设计方法	FLASH空间/KB	RAM空间/KB	启动速度/ms	代码重用率/(%)
CCS 组件设计	引擎	63.95		
	界面	7.55	1.72	1.68
	共计	71.50		90
传统嵌入式设计	75.63	1.85	1.63	30

从表1中可以看出,采用CCS组件设计方案后Flash和RAM空间使用有明显的下降。虽然在界面数据中增加了索引表,但传统嵌入式设计方法必须要花费更多的空间用于描述界面之间的图形组件之间的关系。在界面数据中,索引结构表只使用了208个字节,占整个界面数据的0.3%。由于CCS组件设计的引擎系统除了在上电时初始化硬件外,还需读Flash初始化一级索引表在内的大量数据,启动速度比普通嵌入式设计方法设计的系统要慢。但在人机界面系统中,开机启动时需花大量的时间进行LCD初始化和LCD清屏,因此启动速度只慢了50 μs(该速度是完全可以接受的)。

4 结束语

针对嵌入式裸机的图形界面系统开发效率低等问题,本研究提出了一种新型嵌入式开发模式CCS模型,在启动前通过该模型加载所需组件,实现系统的可重构,实现资源的灵活配置。CCS组件设计方法可以

广泛适用于基于裸机平台的嵌入式系统的开发中,尤其是嵌入式图形设计,可以提高系统的开发效率、减少内存消耗。为嵌入式系统开发提供了一种切实可行的方法。

参考文献(References):

- [1] XU Dong, TENG Qi-ming, CHEN Xiang-qun. Supports for Components Loading and Binding at Boot-time for Component-based Embedded Operating Systems[C]. Second International Conference on Embedded Software and Systems 2005. Xi'an: [s. n.], 2005.
- [2] LINDGREN P, ERIKSSON J, AITTAMAA S, et al. Tiny-Timber, Reactive Objects in C for Real-Time Embedded Systems[C]//The IEEE Computer Society 2008. Munich: [s. n.], 2008:1382-1385.
- [3] BAYNES K, COLLINS C, FITERMAN E, et al. The performance and energy consumption of embedded real-time operating systems[J]. **IEEE Transactions on Computers**, 2003, 52(11):1454-1469.
- [4] COULSON G, BLAIR G, GRACE P, et al. A generic component model for building systems software[J]. **ACM Transactions on Computer Systems**, 2008, 26(1):1-42.
- [5] 廖渊,唐磊,李明树.一种基于Qos的服务构建组合方法[J]. **计算机学报**, 2005, 28(4):627-634.
- [6] 章琛曦,陈耀武.基于计算机网络的嵌入式系统监控软件远程动态加载方法[J]. **仪器仪表学报**, 2001, 22(4):340-341.
- [7] 宋荣.基于Qt/Embedded的嵌入式系统GUI开发[J]. **机电工程**, 2009, 26(4):90-92, 109.
- [8] 于孝辉,陈秋艳,李国义.嵌入式Linux下MiniGUI的移植及软件开发[J]. **辽宁工程学院学报**, 2006, 26(2):90-93.

[编辑:李辉]

(上接第93页)

参考文献(References):

- [1] 谢振伟.基于规则的特装车辆故障诊断专家系统的研究[D].哈尔滨:哈尔滨工程大学计算机科学与技术学院, 2006.
- [2] 陈朝阳,张代胜,任佩红,等.基于故障树分析法的汽车故障诊断专家系统[J]. **农业机械学报**, 2003, 34(5):130-133.
- [3] 纪常伟.基于故障树的汽车故障诊断系统开发[J]. **汽车与动力技术**, 2003(1):52-57.
- [4] 赵艳萍,贡文伟.模糊故障树分析及其应用研究[J]. **中国安全科学学报**, 2001, 11(6):31-35.
- [5] CHEN S M, KE J S, CHANG J F. Knowledge representation using fuzzy petri nets[J]. **IEEE Trans. Knowledge**

and Data Engineering, 1990(3):311-319.

- [6] LI Xiao-ou, YU Wen, LARA-ROSANO F. Dynamic knowledge inference and learning under adaptive fuzzy Petri net framework[J]. **IEEE Transactions on Systems, Man and Cybernetic-Part C: Applications and Reviews**, 2000, 30(4):442-449.
- [7] LI Xiao-ou, LARA-ROSANO F. Adaptive fuzzy Petri nets for dynamic knowledge representation and inference[J]. **Expert Systems with Applications**, 2000, 19(3):235-241.
- [8] 申宇皓,孟晨,傅振华,等.加权模糊Petri网的推理算法研究[J]. **军械工程学院学报**, 2008, 20(3):59-62.
- [9] 鲍培明.基于BP网络的模糊Petri网的学习能力[J]. **计算机学报**, 2004, 27(5):695-702.

[编辑:李辉]