

信号交叉口本地智能化系统的设计与实现

时柏营, 杨晓光, 朱 彤

(同济大学 ITS 研究中心, 上海 200092)

摘 要: 针对实现信号交叉口本地智能化的问题, 提出一种嵌入平台与信号控制机结合的方案。采用核心板加扩展板结合的方式设计嵌入硬件平台, 介绍应用程序模块结构、嵌入操作系统及文件系统的制作与移植等嵌入平台软件层次模块。对系统进行实现与测试, 结果表明系统运算时间符合实时交通控制需求。

关键词: 信号交叉口; 本地智能化; 嵌入式平台

Design and Realization of Local Intelligence System of Signalized Intersection

SHI Bai-ying, YANG Xiao-guang, ZHU Tong

(ITS Research Center, Tongji University, Shanghai 200092, China)

【Abstract】 Aiming at the problem of the local intelligence, this paper proposes a scheme by combination of embedded platform and signal controller. The embedded hardware is designed into core board and expanded board, and the software modules of the embedded platform, including the application program modules, development and transplantation of the embedded operating system and files system, are introduced. The realization and test of this system are implemented and the test result shows the computation time meets the demand for real-time traffic control.

【Key words】 signalized intersection; local intelligence; embedded platform

1 概述

随着 ITS 技术和检测技术的迅猛发展, 城市交通控制系统的研究开发进入了第 3 代系统阶段^[1], 主要特征表现为: 能够获取更加精确可靠的实时交通信息, 信息更新频率加快, 使优化算法的准确性和适用性大大提高; 以交叉口作为信号控制的主体, 根据各流向流量的变化实现灵活控制, 能对实时交通流变化迅速做出反应与决策, 实现交叉口决策控制的智能。因此, 单个交叉口的智能化是第 3 代交通控制系统的显著特点^[1]。

实现交叉口智能化一般有 2 种方式: (1) 交叉口现场采集到数据后, 上传到控制中心, 由控制中心运算得出下一时段控制方案后下载到交叉口控制机。该集中式控制方式在 20 世纪 70 年代已经得到应用, 如 SCOOT 系统^[2]。但是随着交通网络的扩大、交叉口采集数据量增长以及控制算法越来越复杂, 控制中心运算负荷及通信延迟问题等越来越严重, 实现交叉口本地智能化已经成为重要的发展方向。(2) 本地采集到数据后, 在交叉口现场运算处理得出控制方案, 即实现交叉口本地智能化。但现有信号控制机运算能力不强, 难以对数据进行实时处理与运算。因此, 如何与路口信号控制机结合, 实现信号控制本地智能化, 是现阶段迫切需要解决的问题。

本文提出与路口控制机结合的嵌入式方案, 由嵌入式平台负责现场数据的处理与运算, 得出控制方案后传输给信号控制机执行, 实现本地智能化。

2 嵌入式系统工作原理

系统工作原理是在交叉口信号控制机柜中新增加一个嵌入式平台, 与信号控制机共同组成一个控制单元。嵌入平台

实时接收采集的交叉口运行数据并进行处理, 获得交通控制模型计算所需要的交通特征参数, 如到达流量、交叉口排队长度等, 并以此为基础进行本地交叉口实时交通控制的优化计算, 得出下一时段的信号控制方案, 然后通过通信接口发送给控制机执行, 从而控制交叉口灯色显示。另外, 嵌入平台还可以通过网络与远程控制中心交换数据。嵌入式系统工作原理如图 1 所示。

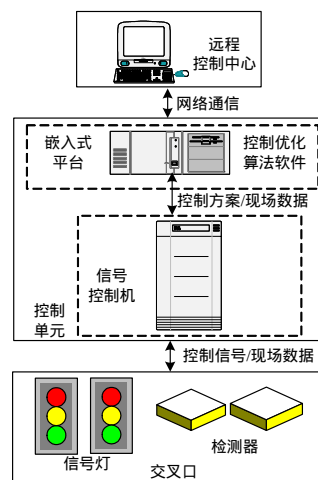


图 1 嵌入式系统工作原理

基金项目: 国家自然科学基金资助重点项目“城市交通网络优化与管理”(70631002)

作者简介: 时柏营(1976-), 男, 博士研究生, 主研方向: 交通控制, ITS; 杨晓光, 教授、博士生导师; 朱 彤, 博士研究生

收稿日期: 2010-04-18 **E-mail:** sby415@163.com

由上文分析可知,作为运行控制算法并与路口控制机结合的嵌入平台,应该具有如下功能特点:

(1)实时性。运算能力满足现场交叉口信号控制实时性需求,对动态变化交通流实时反应。

(2)功能集成性。具有通信、数据预处理、优化运算等多种功能。

(3)通用性。需要提供与不同型号的信号控制机通信的接口,扩展性强。

根据需求的功能特点,下文给出嵌入平台硬件设计。

3 嵌入式平台的硬件设计

嵌入式平台硬件采用典型的核心板加外围扩展板设计,核心板是一个最小系统,主要包括处理器与存储系统。扩展板上有通信接口部分、人机交互部分、调试接口及电源等。嵌入式硬件平台设计如图2所示。

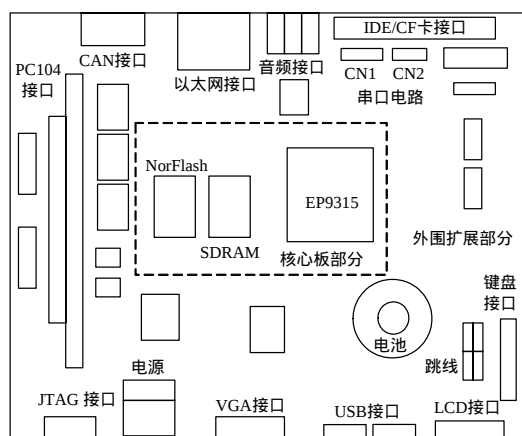


图2 嵌入式硬件平台设计

核心板上主要包括处理器及存储部分。经过比较,处理器选择了 EP9315,它是高度集成的片上系统处理器,拥有 200 MHz 的 ARM920T 的内核、高性能 MMU 存储管理单元,还集成了一个数学协处理器,能够加速浮点运算的速度,满足交通控制实时运算需求。另外,核心板上集成了 64 MB SDRAM、32 MB NorFlash。SDRAM 是程序运行、数据与堆栈的空间,掉电后不能保存数据,但读写速度大大高于 Flash 存储器;NorFlash 用于存储程序,可靠性高、随机读取速度快^[3]。

外围扩展部分主要包括通信接口部分、人机交互部分、JTAG 及串口调试部分、系统辅助资源部分等。由于当前应用的信号控制机型号多样,与外部的通信方式也有串口、以太网接口、CAN 口等多种方式,因此为了保证平台的通用性,扩展部分也集成了多种通信接口。与现场数据采集设备及信号控制机通信可以通过串口或者 CAN 口实现,并且串口设计上将 RS232、RS422、RS485 复用^[4]。与控制中心的通信主要由以太网接口完成。另外为了保证系统应用的扩展性,还集成了一个标准的 pc/104 总线接口。

人机交互部分是工作人员在特殊情况下(如故障状态)现场调试处理的重要接口,主要包括了输出和输入两大部分;在输出接口中包括了 LCD、VGA 显示接口,IDE 和 CF 卡接口以及 USB 接口;输入接口包括 8×8 键盘阵列,PS/2 接口和触摸屏。JTAG 及串口调试部分提供了系统开发调试时的接口。某些串口除了提供通信功能,通过跳线的设置可以在调试阶段实现引导程序下载、程序运行调试等功能。系统辅助资源包括系统电源等。

核心板与扩展板结合构成嵌入硬件平台,可以很好地实现现场数据的接收、处理,完成控制算法运算、存储、通信及人机交互等功能,并具有很好的扩展性。

4 系统软件及系统实现

4.1 软件层次模块

系统软件主要包括 4 个模块:嵌入操作系统(EOS)及文件系统;EOS 启动加载程序及系统初始化程序 bootloader^[5];设备驱动程序;应用程序。首先将 bootloader 下载至板载 NorFlash 的固定位置,系统启动后由它完成系统初始化及 EOS 内核引导工作,之后加载设备驱动程序模块,最后应用程序可以运行执行具体任务。下面说明各个模块功能及具体设计与实现。

(1)应用程序模块。

系统应用程序主要分为检测器数据预处理模块、信号控制优化运算模块、通信模块等,如图3所示。数据预处理模块通过通信模块1从信号控制机接收采集的信息,经过检错、提取交通特征参数后,传送到优化运算模块计算得出控制方案,再通过通信模块1发送到信号控制机执行。远程控制中心可以通过网络与嵌入平台通信,从而可以直接给控制机发送命令。

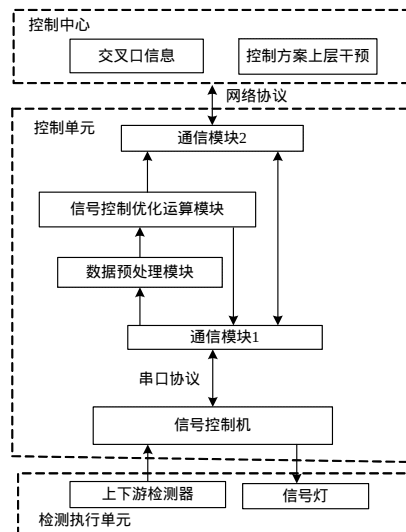


图3 应用程序模块

(2)bootloader 引导程序。

本系统采用 Redboot 作为 bootloader 引导程序,Redboot 不能通用,严重依赖于硬件实现,针对具体的嵌入平台要进行修改与配置。修改完成后重新编译形成 Redboot.bin 文件,在 Windows 操作系统超级终端下将其下载到板载 Norflash 的固定区域保存。

系统启动后 Redboot 开始运行,超级终端下运行界面如图4所示,之后可以在 Redboot 下进行系统设置与内核、文件系统下载等工作。



图4 Redboot 超级终端下的启动界面

(3)嵌入操作系统 EOS 及文件系统。

EOS 采用经过裁剪的 Linux, 本系统内核采用 2.4.21 版本。另外嵌入系统是无盘系统, 需要在内存空间建立根文件系统并且应用程序可以从这里运行。具体过程如下:

1)针对应用的具体需求, 通过内核模块的剪裁和配置文件的修改, 形成配置好的内核文件, 然后制作成内核映像文件 zImage。

2)制作根文件系统 Ramdisk。在制作 Ramdisk 时, 需要考虑把上面编译调试后的应用程序目标文件编入文件系统。在 RedHat9.0 环境下, 把应用程序目标文件加入到指定目录下, 卸载后压缩文件系统, 形成文件系统映像 Ramdisk.gz。

3)运行 Redboot, 利用 TFTP 把 Ramdisk.gz 与 zImage 分别烧写到目标板的 flash 中。系统重启后即可进入系统, 然后应用程序自启动运行, 在 shell 界面可以看到应用程序运行的输出。

(4)设备驱动程序加载。

设备驱动程序既可以嵌入内核随系统一起启动, 也可以作为可加载模块在系统启动之后运行。相比之下, 后一种方式比较简单, 不用修改系统内核, 引入错误的风险性小。因此, 采用加载卸载的方式使用, 这样在具体的应用中, 如采用不同型号的信号控制机时, 对应的通信方式不同, 系统增加或者减少硬件资源非常方便, 增强了系统的扩展性。

4.2 系统实现及测试

本文测试采用德国 SWACRO ITC-1 型信号控制机, 嵌入模块通过串口协议 232 与其通信, 采集数据通过串口 CN1 传输到嵌入平台, 计算生成控制方案通过串口 CN2 发送给控制机。

应用程序在桌面 redhat9.0 下开发, 交叉编译^[6]后形成目标文件待用。应用程序开发中需要注意的是运算模块生成控制方案的表示需要编码成信号控制机可以识别的格式, 例程如下:

```
char msg[50]="F3P1N10=";
char cycr[10]="\x0d";
char cycle[10];
sprintf(cycle,"%d",cy);//设 cy=90s
strcat(msg,cycle);
strcat(msg,cycr);
sendcy=write(*fd,msg,strlen(msg));
```

其中, cy 为传递过来的信号周期参数; 最后串口发送控制机可以识别的字符串"F3P1N10=90"并回车, 表示设置下一时段的信号周期为 90 s。

另外, 应用程序在系统启动后需要自启动运行接收现场数据等工作, 桌面 Linux 系统下一般是通过修改 etc 目录下的配置文件 rc.local 来实现, 但是在嵌入文件系统中没有保留这个文件, 可以打开/etc/inittab 文件, 将 ttyAM0::askfirst:中的 askfirst 修改成 wait, 可以不用输入回车而自动启动应用程序。

系统硬件设计实现及 EOS 内核、Ramdisk 文件系统的制作、下载如上文所述。信号交叉口运行状态信息由信号控制机通过 CN2 发送到通信模块 1, 再通过通信模块 2 把状态信

息通过网络通信发送到控制中心用于显示, 另外控制中心也可以直接发送命令给信号机, 用于手动改变信号灯灯色显示。控制中心运行界面如图 5 所示。

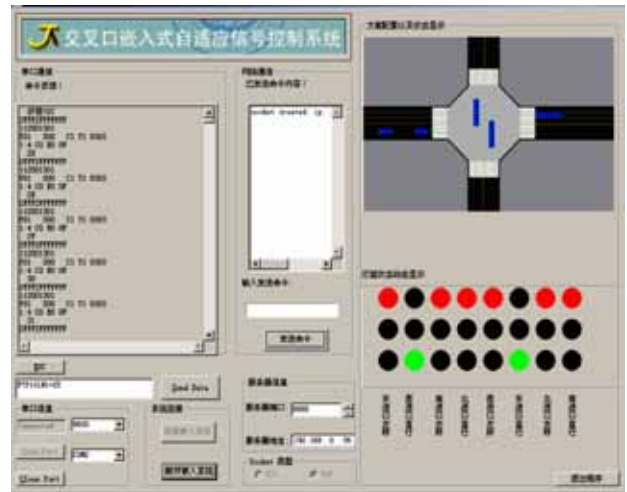


图 5 控制中心运行界面

在系统投入现场运行之前, 必须对系统各项性能就行测试, 主要测试运算模块运算时能否满足现场实时控制需求。由于测试时没有现场数据, 因此编写一个数据产生程序用于模拟现场数据。运算时间与具体控制算法的复杂度有密切关系, 经过运行测试, 自行开发的自适应控制算法, 在各种状态下一一般在 1 000 ms~3 000 ms 之间即可运算完毕输出控制方案, 这样完全可以满足现场控制需求, 对交通流动态变化做出反应。

5 结束语

针对中国交通特性进行的研究已经很多, 也得出了许多有针对性的控制模型与算法。如何把这些模型算法与本地控制机结合是需要解决的问题。嵌入式技术的发展为当前信号交叉口的本地智能化提供了很好的平台, 这种方式实现灵活, 便于扩展, 通用性强。随着嵌入技术的飞速发展及处理能力的提高, 其在交通控制领域也将发挥越来越大的作用。本文对信号交叉口本地智能化系统只进行了初步的研究与实现, 下一步的工作是进行现场的实验与应用。

参考文献

- [1] 杨晓光. 中国城市道路交通实时自适应控制与管理系统研究[J]. 交通运输工程学报, 2001, 1(2): 74-77.
- [2] 全永荣. 城市交通控制[M]. 北京: 人民交通出版社, 1989.
- [3] 陈兵, 张峰, 丁秋林. 基于 ARM 处理器的嵌入式防火墙设计与实现[J]. 计算机工程, 2009, 35(6): 197-199.
- [4] 李驹光, 郑耿, 江泽明. 嵌入式 Linux 系统开发详解——基于 EP93xx 系列 ARM[M]. 北京: 清华大学出版社, 2006.
- [5] 李毅, 李连云, 张伟宏, 等. Bootloader 面向不同结构 Flash 的实现[J]. 计算机工程, 2008, 34(4): 82-83.
- [6] 李曦, 张飞, 时正, 等. 面向 RTEMS 的嵌入式软件集成开发环境[J]. 计算机工程, 2009, 35(4): 41-43.

编辑 顾姣健