

高速电梯主控器嵌入式软件设计与实现

苏昊¹ 倪俊龙² 韩尧¹ 许勇¹ 刘瞰东¹¹(厦门大学信息科学与技术学院 福建 厦门 361005)²(南平电业局 福建 南平 353000)

摘要 针对高速电梯控制系统不断增加的功能需求以及电梯智能化、安全性、可靠性等方面不断提高的要求,用 ARM 构建电梯主控制器硬件平台并完成了 BootLoader 及 uCLinux 操作系统的移植;编写了 CAN 总线等相关驱动程序;设计了高速电梯的跑层逻辑控制算法以及基于多目标规划和支持向量机的电梯群控算法;采用 SQLite 实时数据库快速实现电梯运行参数的存储与读取;同时采用 CPLD 和 AVR 单片机设计主、辅两套高速计数电路,保证了电梯在高速运行状态下的稳定性与可靠性。在电梯模拟控制台上的实际测试验证了该电梯主控制器设计的合理性与可靠性。

关键词 ARM uCLinux CAN 总线 电梯跑层算法 电梯群控算法 SQLite 数据库

DESIGN AND IMPLEMENTATION OF EMBEDDED SOFTWARE IN EXPRESS ELEVATOR MAIN CONTROLLER

Su Hao¹ Ni Junlong² Han Yao¹ Xu Yong¹ Liu Tundong¹¹(School of Information Science and Technology, Xiamen University, Xiamen 361005, Fujian, China)²(Nanping Electric Power Bureau, Nanping 353000, Fujian, China)

Abstract In order to meet the increasing function needs of express elevator control system and the growing improved requirements of the elevator in intelligence, security and reliability, in this paper the ARM chip is used for the construction of hardware platform of elevator's main controller. Bootloader and uCLinux operating system are ported on this platform, and corresponding drivers such as CAN-bus are designed as well. Express elevator's operation logical control algorithm and elevator group control algorithm based on multi-objective programming and support vector machine are designed. SQLite real-time database is introduced into this system to implement rapid store and access of elevator operation parameters. CPLD-based main high speed counter is designed while the auxiliary one is implemented with AVR MCU, this redundant architecture of high speed counter guarantees the stability and reliability of elevator system. Practical tests carried out on elevator simulation platform verify the correctness and reliability of the design of elevator's main controller.

Keywords ARM uCLinux CAN-bus Elevator operation algorithm Elevator group control algorithm SQLite database

0 引言

随着现代高速电梯功能的不断增加,电梯运行的快速性、可靠性和安全性等方面面临着更高的要求,传统的继电器控制、PLC 控制已经越来越不能适应电梯发展的需要^[1],在电梯主控制器中采用 32 位嵌入式系统已经成为电梯发展的必然趋势。

目前,国内高速电梯控制系统市场基本上被国外电梯厂商占有,国内普通电梯一般采用单计数方式,不能充分保证高速电梯的运行要求,在高速电梯研制方面进展缓慢。本文开发的电梯主控制器,选择 32 位的 ARM 芯片为主 CPU,采用 CPLD 与 AVR 单片机设计了主、辅两路高速计数电路,在硬件上保证电梯高速运行时的快速性、可靠性和安全性;利用 CAN 总线实现主控制器与内呼板、外呼板之间的高速数据通信,满足高速电梯的控制要求;设计了以太网接口,使电梯参数设定、远程监控更加可靠便利,降低了系统成本;同时,引入 uCLinux 嵌入式操作系统,实现控制过程的多任务处理,增强了电梯系统的可移植性、可扩展性和可维护性;此外,文中针对电梯控制系统设计了高速

电梯跑层控制逻辑算法,并在单梯算法的基础上,设计了基于多目标规划和支持向量机的电梯群控算法;在数据存储方面,本文通过 SQLite 实时数据库对电梯的参数进行实时存取,快速记录电梯运行状态及故障发生的时间、原因等详细信息。本文设计的电梯主控制器对于国内高速电梯设计具有一定的实际参考价值。

1 电梯控制系统硬件设计

根据电梯控制系统的开发要求,电梯主控制器需要对内、外呼板信号进行采集、处理,还需要对变频器信号、门机信号、安全回路信号、功能信号等数百个 I/O 信号进行分析、控制;同时,主控制器必须配备以太网接口以实现参数下载和远程监控;而且,电梯控制系统还要达到额定速度为 6m/s 的高速电梯的设计要求。除此之外,电梯还应该具有自学习、消防等近百种功能。

收稿日期:2009-06-22。福建省科技计划重点项目(2008H0060)。刘瞰东,副教授,主研领域:嵌入式智能信息处理与控制。

针对上述电梯控制系统的开发要求,本设计采用基于 ARM7TDM I内核的 LPC2292处理器,其具有三级流水线、哈佛结构、低功耗、高性能、适用于工业控制等特点^[2]。利用该处理器出色的内核性能及丰富的接口,构出电梯控制系统的硬件平台,其组成如图 1所示。

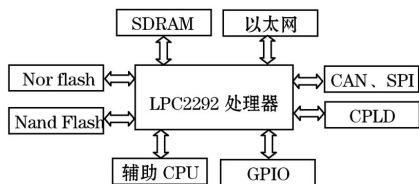


图 1 电梯控制系统硬件平台构成

本设计采用 2MB 的 Nor Flash、8MB 的 Nand Flash 和 16MB 的 SDRAM 组成系统的存储器,保证操作系统和应用程序的运行以及电梯相关参数的存储。ATMEGA128 单片机作为辅助 CPU,用于电梯状态的数码管显示以及辅助编码器采集,单片机与 ARM 之间的通信采用 SPI 总线进行。由于 LPC2292 处理器本身 I/O 口数量的限制,采用 CPLD 芯片用于 I/O 扩展^[3]。LPC2292 本身不具计数功能,本文采用 CPLD 完成主高速计数器的设计,同时采用 ATMEGA128 设计辅助高速计数器,和 CPLD 同时进行编码器值的采集^[4],对外部脉冲计数速度理论值最高可达到 10M bit/s,可以满足电梯高速(6m/s)运行要求,并保证高速运行时的可靠性。CAN 总线接口用来保证主控器与内、外呼板之间的通信。RTL8019 以太网芯片,实现了主控器的参数设定功能和电梯远程监控功能。

2 软件总体架构

根据电梯的硬件平台、电梯的设计功能及电梯实时性、安全性和可靠性的要求,为了降低电梯控制系统的复杂性,增强应用程序的可移植性、可扩展性以及可升级性,本设计中选择了具有内核响应速度快、开源、免费等特点的 uCLinux 操作系统。电梯主控系统的总体软件架构组成如图 2 所示。

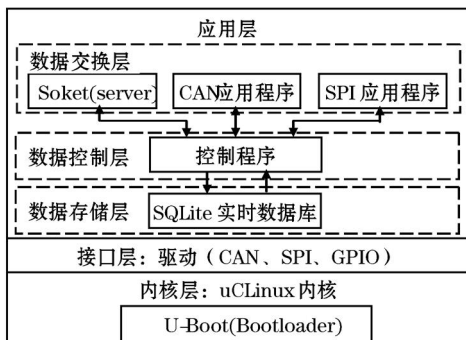


图 2 电梯控制软件总体架构

软件的总体架构分为:内核层、接口层和应用层三层结构。首先由 Bootloader 引导 uCLinux 操作系统的启动;然后接口层实现对硬件设备的封装,减少操作硬件设备的复杂性;最后应用层采用各个模块对接口层进行操作,相当于控制系统的硬件设备。其中,应用层又分为:数据交换层、数据控制层和数据存储层。数据交换层实现对编码器信号的接收、参数设定板参数的设定以及内、外呼信号的采集;数据存储层实现对建筑物楼层的楼层高度以及电梯运行的状态、时间参数、距离参数等数据的存、取;

数据控制层对接收的参数数据进行处理,并实现对电梯的运行控制。

3 内核层的设计实现

3.1 U-Boot启动分析及移植

U-Boot 一个是遵循 GPL 条款的开放源码项目,由于其具有较高的可靠性和高度灵活的功能设置,同时支持较为广泛使用的 CPU,所以本文采用 U-Boot 作为系统的 BootLoader 来引导 uCLinux 操作系统的启动^[5]。

U-Boot 引导 uCLinux 操作系统启动步骤为:与 CPU 相关的初始化和与 CPU 接口设备相关的初始化两个阶段。针对系统硬件平台在 start S 中修改与 CPU 体系结构相关的代码,在 start _ambboot 中修改与平台相关的代码,即可实现 U-Boot 在该硬件平台上的移植。

3.2 uCLinux 的移植、配置和编译

本文主要做的是芯片级的移植,当前 uCLinux 已经支持了无 MMU 的 ARM 体系结构,尽管 uCLinux 中已支持了相应的体系架构,但由于具体处理器设计要求不同,各种基于 ARM7 处理器核的芯片的存储系统不同,片内外设不同,具有的中断源也不同,大部分 ARM7 处理器的中断系统也大不相同,可以说,对于 ARM 平台的芯片级移植的很多工作都集中在对底层中断系统的修改上。由于本文采用 LPC2292 处理器,因此只需添加与 LPC2292 处理器相关的代码即可。主要是修改体系架构和机型相关的代码、中断相关的代码,修改 Makefile 和配置菜单^[6]。内核编译过程如下:

首先到 <http://www.uclinux.org> 网站上下载 uclinux-dist-2004-0408 tar.gz 并且解压,使用 NXP 的 LPC2292 的补丁包 uclinux-dist-lpc22xx patch.gz,对其打补丁,接着使用 make config/make menuconfig/make xconfig 对 uCLinux 的内核进行配置。厂商选择 Philips,产品选择 LPC2292,内核和库版本分别选择 Linux-2.4.x 和 uClibc。根据需要对内核的选项进行选择。最后采用命令 make dep、make 对内核进行编译。

4 驱动程序的设计实现

Linux 驱动程序属于 Linux 内核的一部分,其功能主要是操作硬件设备,为用户屏蔽设备的工作细节,并向用户提供透明访问硬件设备的机制^[7]。

CAN (Controller Area Network) 是一种非常有效的支持实时控制和分布式控制的串行通信网络。CAN 总线以其成本低廉、可靠性高、抗干扰能力强、通信方式灵活等特点,目前已经广泛应用于网络化控制系统中。

本文中 CAN 的设计采用中断加阻塞 I/O 的方式实现,当没有数据可读时,CAN 线程将进入睡眠态,当有数据可读时,在中断服务程序中将读线程唤醒,这种设计可以有效节省资源。CAN 驱动程序的实现使得应用程序操作设备的复杂程度大大降低,同时,也提高了电梯控制系统的快速性和灵活性。实现过程如下:

实现 CAN 操作的数据结构如下。

```
static struct file_operations can_fops = /* driver info */
{
```

```

.....
write: can_write,           //写操作
read: can_read,           //读操作
open: can_open,          //打开设备
release: can_release,     //关闭设备
};

```

设备类型和设备号的定义如下:

```

#define CAN_MAJOR_NR 130
#define DEVICE_NAME CAN

(1) 设备注册与注销
register_chrdev(CAN_MAJOR_NR, DEVICE_NAME, &can_fops);
第一个参数为此设备动态分配的一个主设备号,第二个参数为设备名,最后一个参数是对各个系统调用的入口点的说明。
unregister_chrdev(CAN_MAJOR_NR, DEVICE_NAME);

(2) 设备的打开与关闭
static int can_open(struct inode * inode, struct file * filp)
//打开设备

static int can_release(struct inode * inode, struct file * filp) //关闭设备

(3) 设备的读与写
static ssize_t can_read(struct file * filp, char * buf, size_t count, loff_t * f_pos)
//内核空间拷贝到用户空间

static ssize_t can_write(struct file * filp, const char * buf, size_t count, loff_t * f_pos)
//数据从用户空间拷贝到设备

(4) 中断过程
request_irq(RQ_CAN1_RX, can0_handler, SA_INTERRUPT, CAN_name[dev ->Minomum], dev); //申请中断
wait_event_interruptible(dev ->inq, (dev ->Recvhead != dev ->Recvtail)) //缓冲区没有数据,线程进入睡眠
void can0_handler(int irq, void * dev_id, struct pt_regs * regs) //中断服务程序
wake_up_interruptible(&(dev ->inq)); //唤醒线程 free_irq(dev ->Inrnum, dev); //注销中断

```

5 应用程序的开发

根据电梯开发设计的要求,本文把整个应用程序划分成数据库模块、通信模块和控制模块的结构,如图 3 所示。

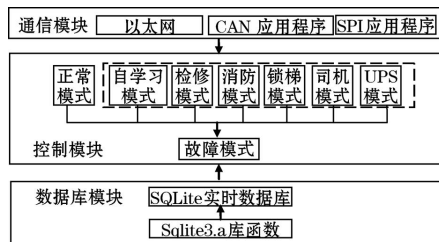


图 3 应用程序总体框架

数据库模块主要读写电梯参数信息,包括:电梯的楼层信息、时间参数、楼高参数、功能性参数等;通信模块包括:以太网模块、CAN 通信模块、SP 通信模块三个部分。以太网模块通过 TCP/IP 协议接受参数设定板的参数信息,采用 Socket 通信模式,与上位机监控软件进行通信;CAN 应用模块主要用来作为与内、外呼之间的信息传递,包括:楼层信息、楼层字符的显示、风扇照明的通断、开关门到位等信息,其中读程序采用中断方式,以提高实时性;SP 通信模块主要用来作为与辅助 MCU 的通信,该通信包括:编码器的值、电梯的状态信息、时钟数据;控制模块是整个系统的核心,该模块包括电梯单梯运行的三大模

式:正常模式、非正常模式、故障模式,在单梯模式之外控制模块还包含电梯群控模块及相关算法。正常模式模块包含电梯单梯运行的相关程序和跑层算法;非正常模式包括:自学习模式、检修模式、消防模式、UPS 运行模式、司机模式、锁梯模式;故障模块保证电梯在出现故障的情况下做出反应,确保电梯安全运行。

5.1 数据库模块设计

5.1.1 SQLite 数据库移植

SQLite 是用 C 语言编写的开源嵌入式数据库引擎,占用资源非常少。在嵌入式设备中只需几百 KB 的内存就可以运行,处理速度非常快。本文采用 v3.8 版本^[8]。

- (1) 在 SQLite 的官方网站 www.sqlite.org/download.html 上下载 sqlite-3.8.1.tar.gz 软件包,解压缩,并修改 main.mk,
- (2) 拷贝 Makefile_linux-gcc 为 Makefile,并作修改。
- (3) 编译后出现静态库 libsqlite3.a。

5.1.2 数据库模块的实现

```

rc = sqlite3_open(数据库名, &db); //打开建立数据库
关闭数据库 sqlite3_close(db); //打开建立数据库
sqlite3_exec(db, SQL 语句, NULL, NULL, &err_msg); //执行 SQL 语句

```

5.2 通信模块

该模块主要完成 ARM 芯片与其它芯片的信息交换,该模块主要由三部分完成:以太网模块采用 TCP/IP 协议,通过 Socket 通信与参数设定板按照自定义的数据格式进行通信,完成电梯参数的设定(包括电梯的时间参数、距离参数、功能性参数等一百多个参数的设定);CAN 通信模块,完成与内、外召板直接的通信,主要用来传输楼层信息,以及语音、VIP 电扇、照明等一系列的控制信息;SP 通信模块主要完成 ARM 与单片机直接的通信。

5.3 控制模块

5.3.1 正常模式下单台电梯跑层算法

(1) 主控板接收的呼梯信号分析

内召板的呼梯信号

在电梯位置“前方”的内召板的呼梯信号记为 s_{in_front} ;电梯位置“后方”的内召板呼梯信号记为 s_{in_pass} 。

外召板的呼梯信号

与电梯运行方向在“同方向”且在电梯位置“前方”的外召呼梯信号记为 s_{out_front} ;与电梯运行方向在“同方向”且在在电梯“后方”的外召呼梯信号记为 s_{out_back} ;与电梯运行方向在“反方向”的外召呼梯信号记为 $s_{out_reverse}$ 。

根据电梯的设计要求,电梯应该沿着当前方向前进直到没有前方的呼梯信号再反向,因此程序的数据结构采用 3 个单链表来完成以上四种信号的呼梯请求。本文把存储信号 s_{in_front} 和 s_{out_front} 的单链表记为 $sqlista$;把存储信号 $s_{out_reverse}$ 和 s_{in_pass} 的单链表记为 $sqlistb$;把存储信号 s_{out_back} 的单链表记为 $sqlistc$ 。

“前方”、“后方”的定义说明

考虑电梯在运行过程中有一定的速度,如刹车,必须有一定的刹车距离。“前方”:为目前电梯的运行位置(即当前电梯所在的位置)+电梯的刹车距离,电梯运行同一方向的前面位置;“后方”:除了“前方”之外的为“后方”。

(2) 三组信号的响应流程

当电梯从当前楼层出发,读取 $sqlista$ 的元素作为目的楼层,

到达该楼层后删除链表中的楼层信息,直到 $sq\text{lista}$ 为空;电梯进行转向,相应地三个链表将发生相互转化, $sq\text{listb}$ 转换为 $sq\text{lista}$, $sq\text{listc}$ 转化为 $sq\text{listb}$, $sq\text{listc}$ 清零。具体的响应流程如图 4 所示。

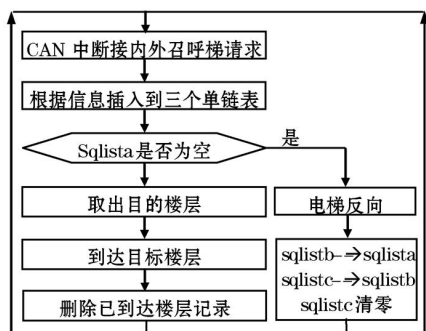


图 4 单台电梯跑层算法流程

(3) 基于预测控制算法的电梯运行曲线

目前高速电梯存在着如下控制难点。首先,电梯控制是非线性系统,难以建立它的精确数学模型。其次,电梯启动与制动段舒适性的调整。第三,电梯下降时,将损失全部势能,并将其转换成热能,频繁使用,将使机械系统温度上升而不得不停机,同时,也使控制系统性能变差,影响舒适性^[9]。

本文采用预测控制算法对电梯运行曲线进行跟踪控制。预测控制算法是目前高速电梯控制系统中应用较广的方法之一,它可以减少系统的启动死区,能够较好地提升高速电梯的运行品质。它对电梯的控制实际上是对电梯理想速度曲线的跟踪控制过程^[10]。如图 5 所示,电梯的运行曲线具有启动加速段,高速运行段,减速制动段,爬行(平层)段。在启动段和减速制动段,速度曲线是变化的。通过预测控制,当接通主、副接触器后电机中有电流输出,接着给变频器方向指令和抱闸输出指令,再根据电梯的逻辑跑层算法给出目标楼层,依据目标楼层和本楼层的高度差选择速度指令的大小;当要到达目标楼层时,电梯给予爬行速度指令;当进入入门区距离时,电梯将进行主抱闸接触器动作,考虑到电梯的惯性作用,电梯将滑行一段距离后停车。

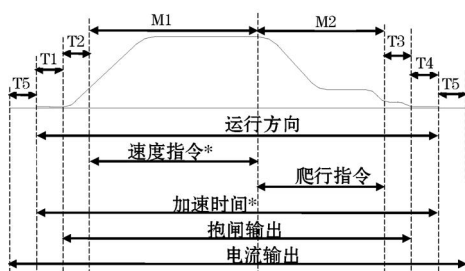


图 5 电梯运行逻辑曲线

算法充分考虑了电梯的服务质量——高效、安全、舒适、节能运行的平衡。在追求速度的同时不放弃对安全性与舒适性的要求。由于算法效率的提高,减少了电梯运行时的能量消耗。

5.3.2 各种模式的设计

(1) 正常模式模块

正常模式是电梯在正常的状态下运行,根据电梯运行的算法对电梯的内、外呼信号进行响应,到达楼层后能够自动执行开门信号。

当电梯从其他状态转换到正常状态的时候,首先应该判断电梯是否处于平层区,即上、下平层信号是否有效。如果电梯不

是处于平层区,电梯首先应该进行平层;当电梯处于平层状态下时,若没有呼梯信号则电梯处于等待状态,一旦有呼梯信号,根据电梯的跑层算法,按呼梯信号的优先顺序进行响应。

(2) 非正常模块

该模块包括:自学习模块、检修模块、消防模块、锁梯模块、司机模块、UPS模块。自学习模式即是电梯学习每个楼层的高度、以及楼层数、向上入门区距离、向下入门区距离的过程;检修模式主要是电梯在检修状态下的运行;消防模式即在电梯发生消防情况时电梯的运行逻辑;锁梯模式即当电梯处于不用状态时应把电梯锁住。司机模式要求跑完内召所登记的召唤后,回到锁梯基站,关闭照明和风扇;UPS模式就是在电梯发生停电时,UPS起作用后,电梯进入 UPS 运行。根据电梯设计要求,各个模块响应的优先级从高到低依次为故障模式、UPS模式、检修模式、自学习模式、消防模式、锁梯模式、司机模式。

(3) 故障模块

为了保证电梯的安全运行,当电梯的硬件系统或者软件系统出现故障的时候,电梯将进入故障模式。此时电梯应该处于停车状态,并将此故障原因写入数据库中,以便检修人员查明故障原因。

5.3.3 电梯群控算法设计

本文设计了基于多目标规划和 SVM (Support Vector Machine) 的电梯群控算法。

(1) 基于多目标规划的电梯群控算法

首先,选取侯梯者的侯梯时间、乘客的乘梯时间以及电梯的能量损耗作为控制目标;定义乘客从按下按钮到电梯 n 到达该楼层、乘客进入轿厢这一过程所经历的时间为侯梯时间 $TOW(n)$;定义电梯 n 中的乘客从当前层站到内召最远目的层站所需要的时间为乘客乘梯时间 $TOR(n)$;设电梯 n 运行的能量损耗为 $EN(n)$ 。而后,根据电梯的实际情况,适当抽象和简化,对上述三个控制目标分别建立目标函数并确立约束条件^[11]。由于 $TOW(n)$ 、 $TOR(n)$ 、 $EN(n)$ 三者的量纲不同,所以必须将三者分别标准化,得到 $\mu TOW(n)$ 、 $\mu TOR(n)$ 及 $\mu EN(n)$,从而,得到评价函数:

$$S(n) = W_1 \times \mu TOW(n) + W_2 \times \mu TOR(n) + W_3 \times \mu EN(n)$$

最终,最优评价函数为:

$$S(j) = \min\{S(1), S(2), \dots, S(M)\} \quad j \in \{1, 2, \dots, M\}$$

根据上式求得的即为最佳派遣单梯号。

(2) 基于 SVM 的交通流模式识别

由于电梯在不同时段交通流模式有很大差别,而我们的多目标群控算法需要在不同的交通流模式下适时调整控制目标的加权系数 W_1 、 W_2 、 W_3 以收到满意的效果,所以本文在多目标规划算法层之上采用 SVM 算法,识别交通流模式。通常电梯的交通流模式可以分为:上行高峰模式、下行高峰模式、随机层间模式和空闲模式。鉴于 SVM 出色的分类能力和结构风险最小化的优势,我们在此采用一对一的多值分类 SVM 算法进行交通流模式识别^[12,13]。

对于一对一的 k 类多值分类问题,第 i 类和第 j 类的分类超平面可描述如下:

$$\min_{ij} \left[\frac{1}{2} \tau_{ij} + C \sum_t \eta_{ij,t} \right]$$

$$s.t. \quad \tau_{ij} (x_t) - 1 - \eta_{ij,t} \quad \text{若 } y_t = i$$

$$\tau_{ij} (x_t) - 1 + \eta_{ij,t} \quad \text{若 } y_t = j$$

其中, $i, j \in \{1, 2, \dots, k\}$, τ_{ij} 为权值向量, η_{ij} 为松弛变量, C 为惩罚系数, (x_t) 表示从输入空间到特征空间的变换集合。

对电梯交通流模式分类问题,本设计以电梯乘客总量、进入建筑物门厅人数及离开建筑物人数这三者为输入特征向量,得到的输出为前述四种电梯交通流模式。对这个 3 维输入特征向量的四分类问题,构建如前所述的 6 个子分类器,采用投票规则 (Voting Strategy) 进行分类。在电梯运行中,调用本算法判别交通流模式的分布时段,而后确定多目标规划中控制目标的权值 W_1 、 W_2 、 W_3 ,调用多目标规划算法,计算出最优派梯方案。

6 软件测试

本软件的测试主要分为为驱动程序的测试和应用程序的测试两个部分进行。

驱动程序的测试包括 CAN、SPI、GPD 等驱动程序的测试,主要采用在用户态编写应用测试程序,来判断驱动程序是否正确,本文在测试的过程中采用两个线程进行,即一个线程发送数据,一个线程接收数据,在测试过程中 CAN、SPI 总线均能够 100% 接收到信息。

应用程序测试在厦门汉京电梯控制系统有限公司的电梯模拟台上进行。电梯在正常模式下能够按照跑层算法完成对内外呼信号的响应,而且电梯的舒适度能够根据电梯的速度切换进行调整,除单梯模式外,电梯群控算法也进行了测试,收到了较好的效果。另外,电梯在各个非正常模式下均能够正常运行。电梯在低速、中速、高速运行下测试,结果如表 1 所示。

表 1 不同速度下的电梯运行情况

速度	2.3m/s	4.0m/s	6.1m/s
正常模式	响应正常	响应正常	响应正常
异常模式	进入/退出正常	进入/退出正常	进入/退出正常

7 结语

本文设计的高速电梯主控器,能够满足电梯控制系统功能日益增多和系统安全性、可靠性不断提高的要求;采用 uCLinux 嵌入式操作系统能够满足高速电梯的响应速度,降低系统设计的复杂性,增强系统的可升级性;本文中设计的跑层算法和群控算法在保障电梯正常运行的同时,满足了乘客的舒适性、候梯时间等各方面的指标要求。综上,本文设计的电梯主控器,对国内电梯控制系统的开发设计具有良好的实际参考价值。

参 考 文 献

- [1] 王恒升,李丹峰,龙迎春. PLC 电梯控制系统设计[J]. 工业仪表与自动化装置, 1999(6): 25-27.
- [2] 周立功,等. ARM 嵌入式系统基础教程[M]. 北京:北京航空航天大学出版社, 2005: 33-34.
- [3] 彭可,陈岚,李仲阳,等. 基于 CPLD 和多处理器结构的控制网络节点设计[J]. 计算机工程与应用, 2005, 23: 209-212.
- [4] 李彬,萧德云,张正芳. 基于单片机和 CPLD 的智能 I/O 模块设计[J]. 计算机工程与应用, 2006, 36: 66-69.
- [5] DENX Software Engineering [EB/OL]. [2009] http://www.denx.de/wiki/U-Boot
- [6] 周立功,陈明计,陈渝. ARM 嵌入式 Linux 系统构建与驱动开发范例[M]. 北京:北京航空航天大学出版社, 2006: 71-72.
- [7] Jonathan Corbet, Alessandro Rubini, Greg Kroah-Hatman. Linux device driver[M]. Beijing: O'Reilly, 2006: 1-14.
- [8] Hwaci [EB/OL]. [2009] http://www.sqlite.org/.
- [9] 周志翔,刘剑. 超高速电梯发展中存在的问题与研究方向[J]. 控制工程, 2003(10): 1-4.
- [10] 诸小鹏,张广明. 基于 GPC 的电梯运行速度控制[J]. 控制工程, 2004, (11): 180-182.
- [11] 李彦华. 智能电梯群控系统的设计[D]. 厦门大学, 2008, 5.
- [12] 许玉格,罗飞. 基于 SVM 的电梯群控系统交通流模式识别[J]. 华南理工大学学报:自然科学版, 2005, 6, 33(6): 32-35.
- [13] Chi-Wei Hsu, Chih-Jen Lin. A Comparison of methods for multiclass support vector machines[J]. IEEE transaction on neural networks, 2002, 13(2).

(上接第 97 页)

在本案例中,知识可以通过不同的终端进行分发,包括 Web 浏览器、PDA 和手机等。丰富的终端分发手段使得销售人员可以随时随地得到所需知识; eLearning 系统可以根据不同的终端设备格式进行学习对象分发。

在本案例中,负责组织知识学习的教学设计者在智能化的课程创作软件支持下,工作效率会很高;知识经理与知识组织软件则负责知识组织工作。在该软件公司,少数几个知识经理和教学设计者就能够创作大量的学习对象供员工进行学习。

4 结语及未来的工作

本文描述了 eLearning 如何应用于企业知识管理,提出了基于 eLearning 的企业知识管理模型,并给出了应用案例。研究工作和应用案例表明,基于 eLearning 的企业知识管理模型能更好地发挥 eLearning 和知识管理的优势,整合 eLearning 和知识管理将有效提升企业的学习能力和工作效率。建立企业知识本是企业员工提供一个能自由访问多个信息源的统一的知识门户的关键,也是未来研究工作的重点之一。

参 考 文 献

- [1] eLearning glossary[EB/OL]. 2008-9-20 http://www.astd.org/LC/glossary.htm
- [2] 迈克尔·波兰尼. 个人知识—迈向后批判哲学[M]. 许泽民,译. 贵州:贵州人民出版社, 2000.
- [3] Nonaka I, Takeuchi H. The Knowledge-Creating Company[M]. Oxford Univ. Press, NY, 1995.
- [4] Schmidt A. Knowledge Maturing and the Continuity of Context as a Unifying Concept for Knowledge Management and E-Learning[C] // Proceedings of I-KNOW '05. Graz, Austria: [s.n.], 2005: 122-136.
- [5] Maurer H, Sapper M. E-Learning Has to be Seen as Part of General Knowledge Management[C] // Proc. of ED-MEDIA 2001 World Conference on Educational Multimedia, Hypertext & Telecommunications 2001: 1249-1253.
- [6] Schmidt A. Bridging the Gap Between Knowledge Management and E-Learning with Context-Aware Corporate Learning Solutions[C] // 3rd Conference on Professional Knowledge Management - Experiences and Vision. Kaiserslautern, Germany: [s.n.], 2005.
- [7] Yang S, Chen I, Shao N. Ontology Enabled Annotation and Knowledge Management for Collaborative Learning in Virtual Learning Community[J]. Educational Technology & Society, 2004, 7(4): 70-81.
- [8] Staab S, Schnuur H, Studer R, et al. Knowledge Processes and Ontologies[J]. IEEE Intelligent Systems, 2001, 16(1): 26-34.