

UCLINUX 下自主移动机器人运动系统设计

孙勇智,陈黎明,林江杰

(浙江科技学院 自动化与电气工程学院,杭州 310023)

摘要: 设计了在嵌入式操作系统 UCLINUX 下的自主移动机器人运动系统。该系统主控制器采用了 ARM7 微处理器,操作系统使用 UCLINUX,利用脉冲宽度调制(PWM)技术对直流电机转速进行控制。详细介绍了 UCLINUX 下电机控制器的设备驱动程序设计和移动机器人基本动作的应用程序设计。该系统已经应用于一个移动机器人原型中,能够成功地实现各种动作。

关键词: 嵌入式操作系统;移动机器人;设备驱动程序;脉冲宽度调制

中图分类号: TP242

文献标识码: A

文章编号: 1671-8798(2007)04-0273-04

Motion System Design for Autonomous Mobile Robot in UCLINUX

SUN Yong-zhi, CHEN Li-ming, LIN Jiang-jie

(School of Automation and Electrical Engineering, Zhejiang University of Science and Technology,
Hangzhou 310023, China)

Abstract: Motion system design for an autonomous mobile robot is presented in this paper. The motion system works in an embedded operation system UCLINUX. The master controller is ARM7 micro processor, and takes UCLINUX as its operation system. Pulse width modulation waves are generated for rotate speed control of DC motors. The UCLINUX device driver for motor controller and application program for robot's behavior are described in detail. The system has been applied for a prototype of mobile robot, which can perform various actions.

Key words: embedded operation system; mobile robot; device driver programming; PWM

近年来,随着移动机器人技术的不断成熟,其应用也得到了很大的发展,从实验室中逐渐走向制造工业、医疗卫生、国防军工、家用电器、自动仓储等许多领域。不同领域和不同工作环境的要求使得移动机器人的设计特别是软件设计越来越复杂,同时远程控制 and 操作对机器人的联网能力也提出了更高的要求,建立在传统的监控程序之上的移动机器人控制软件已经越来越无法适应这些需要。

嵌入式系统的发展为这些问题的解决提供了有效的工具。嵌入式系统是以应用为中心,计算机技术为基础,软、硬件可剪裁,适用于应用系统对功能、可靠性、成本、体积和功耗要求严格的专用计算机系统^[1]。目前,高性能嵌入式系统发展的一个重要方向是嵌入式操作系统,它是一种支持嵌入式系统应用的操作系统软件,通常包括与硬件相关的底层驱动软件、系统内核、设备驱动接口、通信协议、图形界

收稿日期: 2007-09-01

作者简介: 孙勇智(1972—),男,山东安丘人,讲师,博士,主要从事智能控制与嵌入式系统研究。

面等。嵌入式系统具有良好的实时性、可靠性、灵活性和经济性,在工业和服务领域中得到了广泛的应用。基于高性能嵌入式微处理器和嵌入式操作系统的移动机器人的开发得到了越来越多的重视。

邹丹^[2]等研究了采用履带式行驶机构的车底检查机器人的航向纠偏算法及其在实时嵌入式多任务控制系统中的具体实现。该机器人主控计算机系统采用基于 ARM7 系列微处理器和 uC/OS-II 的嵌入式实时操作系统,具有处理速度快,功耗低,价格低廉等优点。李渊^[3]等研究了基于 uC/OS-II 的嵌入式实时系统的设计方法,并就其在移动机器人控制系统中的实现进行了研究与讨论。张帆等^[4]介绍了一个能运用于智能机器人视觉导航的嵌入式系统,以三星公司的 S3C44B0X 微处理器为中心构建控制模块,同样也是选择 uC/OS-II 作为嵌入式操作系统,利用了其占用空间少,执行效率高,实时性能优良的优点。采用 uC/OS-II 操作系统实时性好,占用空间少,但是 uC/OS-II 仅仅是提供了一个实时多任务调度功能,对文件系统、网络通信等功能的支持较差,开发工作量相对较大。美国 ActivMedia 公司生产的先锋 II 型移动机器人是以 LINUX 为操作系统,以工业控制机为 CPU 的移动机器人,它功能强大,但是造价较高。Acroname 公司的 Garisa 机器人使用嵌入式 LINUX 操作系统,CPU 采用 INTEL 的 XScale,可以执行测绘和导航功能。目前国内也有研究基于 LINUX 操作系统的移动机器人,但是鲜有报道。

UCLINUX 是一种基于 LINUX 内核的嵌入式操作系统,它保留了标准 LINUX 的大多数优点,具有良好的移植性、优秀的网络功能。UCLINUX 针对没有 MMU 的处理器,支持包括 ARM7 在内的多种嵌入式微处理器,它在视频监控^[5]、网络通讯^[6]、数据采集^[7]、语音通讯^[8]等领域得到了广泛的应用。

本文提出基于 ARM7 微处理器和嵌入式操作系统 UCLINUX 的移动机器人控制结构,主要介绍其运动的硬件和软件实现,执行电机采用直流电机,用 PWM 技术对电机转速进行控制,以开发直流电机的设备驱动程序,实现对直流电机的设备独立性操作。目前,该系统已经应用于一个移动机器人原型的控制中,能够成功完成各种基本动作。

1 移动机器人运动控制系统硬件电路

本文开发的移动机器人运动控制系统主控制器采用 Samsung 公司生产的 S3C44B0X 微处理器,

S3C44B0X 微处理器采用 ARM7TDMI 内核,并且提供了丰富的内置部件,包括:8KB 片内高速缓存、LCD 控制器、2 通道 UART、4 通道 DMA、5 通道具有 PWM 功能的定时器、8 通道 10 位 ADC、IIC 接口等。S3C44B0X 丰富的片内资源,使其特别适用于复杂系统的控制。

移动机器人采用两轮驱动方式,由 2 个直流电动机分别提供 2 个驱动轮的动力。为了使机器人能够全方向运动,2 个电动机都可以正反向旋转。

本文采用内部具有双 H 桥的电动机驱动芯片 L298 芯片驱动直流电动机,电路原理图如图 1 所示。在图 1 中,方框内部表示的是 L298 内部电路的一半,由 4 个三极管构成一个 H 桥电路,外接二极管 D1、D2、D3 和 D4 为续流二极管。En 为使能信号,PWM1 和 PWM2 为转速控制信号。当 PWM1 信号为高电平,且 PWM2 信号为低电平时,电动机正转;当 PWM1 信号为低电平且 PWM2 信号为高电平时,电动机反转。

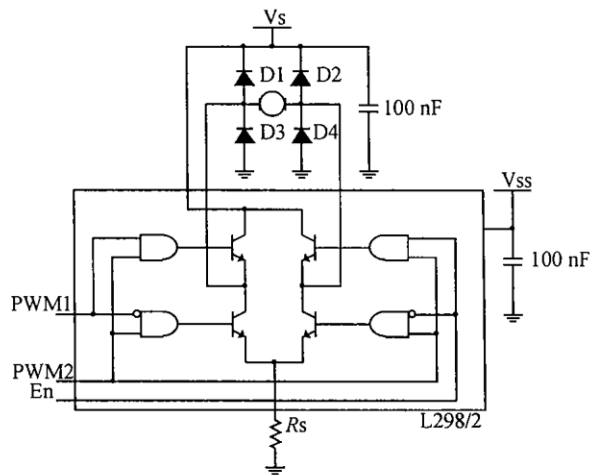


图 1 电机驱动电路原理图

电动机的转速控制原理如图 2 所示,若 PWM1 施加占空比为 $D\%$ 的方波,则 PWM2 应施加占空比为 $1-D\%$ 的方波。在图 2(a)中,PWM1 占空比为 60%,PWM2 占空比为 40%,则施加在电机上的有效脉冲 PWM 的占空比为 20%,电流方向从左至右,使电动机正转;在图 2(b)中,PWM1 占空比为 25%,PWM2 占空比为 75%,则施加在电机上的有效脉冲 PWM 的占空比为 50%,电流方向从右至左,使电动机反转。通过改变 PWM1 和 PWM2 的占空比,就可以控制电机以不同的转速正向和反向旋转。

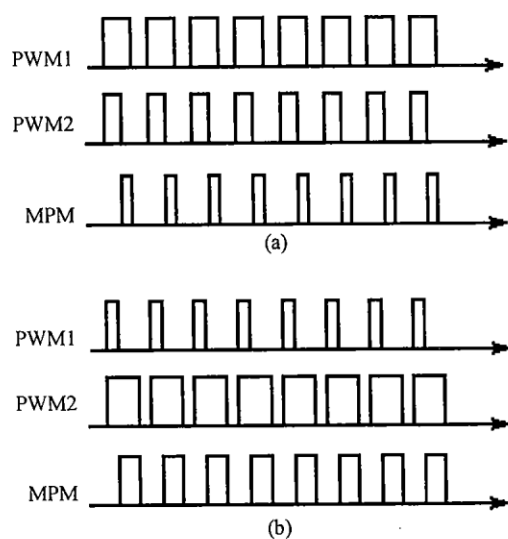


图2 电动机转速控制示意图

2 直流电机设备驱动程序设计

在 UCLINUX 中,设备驱动程序提供了操作系统内核和机器硬件之间的接口,它为应用程序屏蔽了硬件细节。在应用程序看来,硬件设备只是一个文件,可以像操作普通文件一样对硬件设备进行操作。虽然在 UCLINUX 下可以直接对端口操作,但是这样会造成潜在的程序问题,并且程序的可移植性也会受到影响,因此最安全的措施是为新的设备添加驱动程序。

UCLINUX 操作系统下的设备主要有字符设备、块设备和网络设备。字符设备和块设备的主要区别是:字符设备处理信息以字符为单位组织,在对字符设备发出读/写请求时,实际的硬件 I/O 一般就紧接着发生了,如键盘、串行口等设备;块设备处理信息以块为单位组织,利用一块系统内存作为缓冲区,当用户进程对设备请求满足用户要求时,就返回请求的数据,如磁盘和磁带等设备。在本文中,将直流电机设备作为字符设备处理。

在 UCLINUX 中,由于用户进程通过设备文件与硬件交互,所以驱动程序必须处理一些标准的文件操作,如“open”“read”“write”和“close”等。编写驱动程序主要是编写当执行这些文件操作的相应函数。这些函数的入口地址在程序中被定位在 UCLINUX 的一个“file_operations”结构体中(在文件 fs.h 中)。当不考虑直流电动机的反馈信息(如光电编码器等)时,可以将它看作是一个只写设备。可以将“file_operations”结构体定义如下:

```
struct file_operations motor_drv_ops = {
```

```
    write : motor_drv_write,
    open  : motor_drv_open,
    release : motor_drv_release,
};
```

motor_drv_ops 为结构体的名称,motor_drv_write、motor_drv_open 和 motor_drv_release 分别是与打开、写和关闭三个操作相应函数的名称。例如,当驱动程序加载后,如果有应用程序请求打开设备,操作系统会对驱动程序发出“open”请求,驱动程序会按照 motor_drv_ops 中的定义调用 motor_drv_open:

```
int motor_drv_open(struct inode * inode,
struct file * filp)
{
    motor_sys_init();
    MOD_INC_USE_COUNT;
    return 0;
}
```

motor_sys_init 函数对系统的定时器等硬件资源进行初始化。MOD_INC_USE_COUNT 宏使设备计数器加 1,防止文件在关闭前驱动程序被卸载出内核。与之对应的是关闭程序中有一个宏 MOD_DEC_USE_COUNT 使设备计数器减 1。

```
int motor_drv_release(struct inode * inode,
struct file * filp)
{
    MOD_DEC_USE_COUNT;
    return 0;
}
```

对设备的写操作实现两个电动机速度大小和方向控制功能。

```
int motor_drv_write(struct file * filp, const
char * buf,
size_t count, loff_t * f_pos)
{
    int uc[2];
    int * data=(int *)buf;
    copy_from_user(uc,data,count);
    set_motor_state(uc);
}
```

在写操作时,需要与应用程序进行数据交换。由于设备驱动程序工作在内核态,不能简单地用赋值方法与应用程序交换数据,可以使用宏 copy_from_user 将用户应用程序数据传递给内核态数

据。set_motor_state(uc)函数完成对 2 个电机转速进行控制,具体实现从略。其中 uc 包含 2 个数据,分别与左轮和右轮的设定转速(即 PWM 输出波形的占空比)对应。当 $0 < uc[i] < 50$ 时,电机反转,且 $uc[i] = 0$ 时达到最大转速;当 $50 < uc[i] < 100$ 时,电机正转,且 $uc[i] = 100$ 时达到最大转速。

要实现驱动程序的加载,必须定位如下 2 个操作的位置:module_init 和 module_exit,前者在 insmod 的时候执行,后者在 rmmod 的时候执行。关于驱动程序的加载,可参考文献[9]。

3 移动机器人基本动作实现

设备驱动程序加载到内核之后,可以非常方便地通过 UCLINUX 的系统调用来实现移动机器人的基本动作。例如,移动机器人“全速前进”的动作可以通过如下程序实现:

```
#define MAX_PWM 100
void move_forward(void)
{
    int fd;
    ini value[2];
    value[0] = MAX_PWM;
    value[1] = MAX_PWM;
    fd = open("/dev/motor_drv", O_RDWR);
    write(fd, value, 2 * sizeof(int));
}
```

改变上述程序中 value[0]和 value[1]的数值,可以实现不同速度和不同方向的运动。当 value[0] = 0 且 value[1] = 0 时,移动机器人全速后退;当 value[0] = MAX_PWM 且 value[1] = 0 时,移动机器人迅速右转;当 value[0] = 0 且 value[1] = MAX_PWM 时,移动机器人迅速左转;当 value[0] = MAX_PWM/2 且 value[1] = MAX_PWM/2 时,移动机器人半速后退;当 value[0] = 0.75 * MAX_PWM 且 value[1] = 0.75 * MAX_PWM 时,移动机器人半速前进。

目前,该控制系统已经应用于一个移动机器人原型中(图 3)。该系统能够实现各种基本动作以及反应式避碰行为,更复杂的行为目前正在开发中。

4 结 语

设计了一个基于 ARM7 微处理器的移动机器人运动系统,采用 PWM 技术控制直流电动机的速度大小和方向。目前完成了 UCLINUX 下电动机

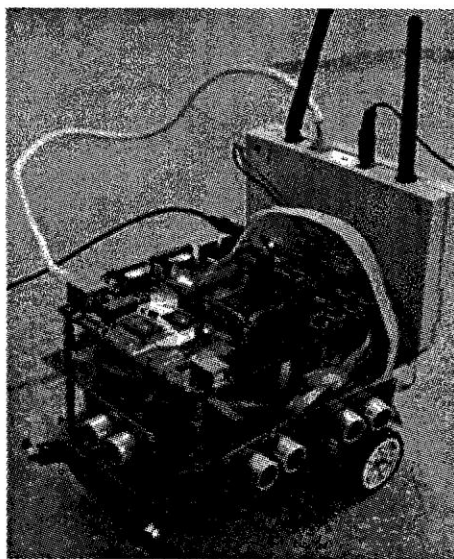


图 3 移动机器人原型

控制电路的设备驱动程序设计,能够实现一些基本动作。该系统采用流行的 32 位嵌入式处理器和嵌入式操作系统,既便于实现远程网络控制和其他系统功能的扩展,又保持了较低廉的成本。应当说明的是,目前的控制系统仅仅是一个开环控制系统,如果跟光电编码器、罗盘等传感器结合起来,可以实现更为丰富的控制功能。目前正在进行各种传感器驱动程序的设计和无线通信系统的设计。

参考文献:

- [1] 王田苗. 嵌入式系统设计与实例开发——基于 ARM 微处理器与 uC/OS-II 实时操作系统[M]. 2 版. 北京: 清华大学出版社, 2003.
- [2] 邹丹, 王田苗, 陈殿生. 车底检查机器人导航控制研究[J]. 计算机工程与应用, 2005, 41(35): 1-3.
- [3] 李渊, 言勇华. 嵌入式实时系统在移动机器人中的应用研究[J]. 微计算机信息(嵌入式与 SOC), 2006, 22(14): 220-223.
- [4] 张帆, 高富强, 顾兴海. 嵌入式系统在机器人视觉中的应用[J]. 自动化与仪器仪表, 2005(1): 38-39.
- [5] 景绍学, 伍超. 基于 ARM7 的网络视频监控系统的的设计[J]. 现代仪器, 2006(4): 46-48.
- [6] 郭玉辉, 王彦瑜. 基于 ARM7 内核芯片网络通讯设计[J]. 计算机工程与设计, 2006, 27(16): 3043-3044.
- [7] 吴平, 曹晓琳, 丁铁夫. 基于嵌入式以太网的语音通讯系统设计[J]. 计算机工程与应用, 2006(16): 163-166.
- [8] 刘凯, 余立建. 基于嵌入式 Linux 系统的远程数据采集[J]. 计算机应用, 2006(6): 264-265.
- [9] Alessandro Rubini, Jonathan Corbet. LINUX DEVICE DRIVERS[M]. Berlin: O'Reilly & Associates, 2002: 10.