

Sensors Measurement Reading for Mobile Robot on ARM-Linux Platform*

Feng Guo, Songdong Xue[#], Jianchao Zeng

Engineering Technology of Internet of Things Research Center, Taiyuan University of Science and Technology, Taiyuan
Email: gf8787@126.com, [#]xuesongdong@163.com

Received: Sep. 15th, 2013; revised: Oct. 8th, 2013; accepted: Oct. 17th, 2013

Copyright © 2013 Feng Guo et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Abstract: The controller design by making use of embedded technology for mobile robot requires managing hardware resources under Linux operating system. Taking the typical proximity sensors amounted on the board for example, considering work principle, sensors and hardware interface of sensors and interface resource of ARM-based microprocessor, an allocation scheme of interfaces is proposed for sensor controlling. Then, the character device drivers of the infrared and ultrasonic sensors are designed by using appropriate cross compiler tools according to the control timing sequences of interface of sensors. Test results show that measurement readings can be transferred correctly to the control system through device driver developed in this paper.

Keywords: Mobile Robot; Sensor; Device Driver; Embedded System

ARM-Linux 平台上移动机器人的传感器检测数据读取*

郭 峰, 薛颂东[#], 曾建潮

太原科技大学物联网工程技术研究中心, 太原
Email: gf8787@126.com, [#]xuesongdong@163.com

收稿日期: 2013 年 9 月 15 日; 修回日期: 2013 年 10 月 8 日; 录用日期: 2013 年 10 月 17 日

摘 要: 使用嵌入式技术设计移动机器人控制器时, 要求在 Linux 操作系统下管理硬件资源。故以移动机器人上配置的典型接近传感器为例, 根据传感器的作用机理、硬件接口及 ARM 处理器的接口资源, 设计了传感器的接口分配方案; 并按照传感器的接口控制时序, 使用交叉编译工具设计了红外、超声波传感器的字符设备驱动程序。测试结果表明, 所设计的驱动程序对传感器测试数据的读取支持良好。

关键词: 移动机器人; 传感器; 设备驱动; 嵌入式系统

1. 引言

移动机器人是机器人学的一个重要分支, 系机械工程、计算机科学、控制工程、电子技术、传感器技术、仿生学等多学科技术的交叉^[1]。按照功能划分,

*资助信息: 国家自然科学基金(61165016); 山西省自然科学基金(2013011019-4); 山西高校科技研究开发项目(20111017); 太原科技大学博士科研启动基金(20112008); 太原科技大学同洲电子科技创新基金(TZ201304)。

[#]通讯作者。

移动机器人可分为感觉系统、决策系统、执行系统等。其中, 感觉系统由一系列传感器及附属电路组成, 不同的传感器能够检测相应的物理量。为了感知外部环境, 机器人通常配置用于安全避障的红外传感器、用于测距的超声波传感器等, 此类传感器亦称为外部传感器; 而光电编码器则用于感知机器人自身的运动状态, 通过检测轮子转过的脉冲数, 间接求得位移增量。除此之外, 机器人的运动控制、定位导航、通信及运

算决策等功能通过控制器发挥作用。控制器的设计可根据需要采用不同的技术实现。对于小型机器人的控制而言，基于嵌入式技术的传感器较为适宜。嵌入式技术是融合了计算机技术、通信技术、微电子技术、语音处理与图像处理技术为一体的综合性技术^[2]。嵌入式系统以实用性为前提，以工业控制为目的，软硬件可剪裁，具有成本低、功能简单、可靠性强、体积小等特点，属于小型的专用计算机控制系统。与通用计算机系统相比，嵌入式系统针对特定的应用而设计，其功能没有通用计算机强大，属于特种计算机系统。欲针对用于目标搜索的小型移动机器人的多任务特性^[3,4]，基于 ARM9+Linux 技术设计控制系统，须首先进行硬件设备的驱动程序设计，以便通过硬件操作完成规定任务。本文以机器人典型配置中的接近传感器为例，阐述驱动程序的设计和测试方法。

2. 控制需求

本文所述移动机器人，是为搭建用于目标搜索协同控制研究的群机器人系统而设计的成员机器人。此种应用背景，要求机器人具有环境感知、交互通信、行为决策、运动执行等功能^[5]。这些功能，要求控制系统运行来发挥作用。由此，可确定控制系统的软硬件构成，即主要包括主控制器、传感器模块、电机驱动模块、通信模块和软件系统等，详见表 1。其中，在基于 Linux 的软件体系结构中，软件系统可大致分为系统层、接口层、应用层三部分^[6]，如图 1 所示。系统层包括设备驱动程序和 Linux 操作系统，设备驱动程序负责向操作系统提供物理设备操作的逻辑接口；Linux 操作系统完成任务管理、内存管理以及任务通信等基本功能，对所有物理设备以及计算资源等进行统一协调、指挥和控制；接口层所包括的 API 应用程序编程接口，实际上是操作系统为上层应用程序提供的调用接口，应用程序通过调用操作系统的 API 而使操作系统执行应用程序的命令；应用层由应用程序构成，这些应用程序面向用户完成基本的任务以及其他特定任务。

2.1. 要求

移动机器人使用自身配置的传感器检测环境信息，检测数据送控制系统读取后，输入控制算法进行综合分析决策，以决定机器人的运动行为。然后输出

Table 1. Composition of robot control system
表 1. 机器人控制系统组成

名称	构成及参数配置
主控制器	S3C2440, 64 MB 内存, 256 MB FLASH
传感器模块	红外传感器 E18-D80NK, 超声波传感器 HC-SR04
运动模块	电机驱动电路
通信模块	ZigBee, CC2530
软件系统	Linux 操作系统
I/O 元件及接口	按键, LED, 以太网, 串口

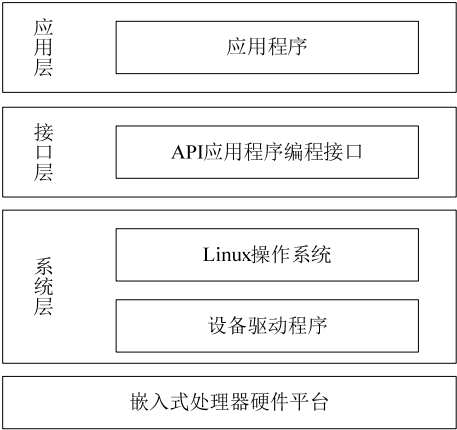


Figure 1. Software architecture of Linux system
图 1. Linux 系统软件体系结构

控制信号，通过驱动模块控制差动电机做前进、后退及转向动作，不同的动作组合可以综合产生期望的行为，完成规定任务^[7]。在群机器人协同控制场景下，还可以根据机器人之间的交互需要，控制系统能够通过通信模块，接收邻居机器人的信息或者发送自己的信息以便共享。为保证控制的实时性，应使用中断方式读取传感器数据。同时，为满足移动机器人二次开发的需要，主控制器应选择高性能 ARM 系列的微处理器。基于此，可对控制器进行选型设计。

2.2. 选型设计

按照控制要求，并综合考虑其他因素，控制器模块选用三星公司的 ARM9 系列 S3C2440 芯片。该模块内部时钟 400 MHz，具有丰富的外部中断资源。配置 256 MB Nandflash 存储芯片，64 MB SDRAM，USB，网口和串口后，可组成基本的嵌入式硬件控制平台，能够满足移动机器人控制算法的实时性要求。例如，红外传感器使用中断的方式提醒 S3C2440，在

传感器的作用范围内遇到障碍物时能够及时响应并处理；再如，超声波传感器的测距需要精确的计时，而该芯片具有 5 个 16 位定时器，能够获得较精确的计数时间。

2.3. 典型传感器配置

为了叙述便利，只考虑接近传感器。红外传感器 E18-D80NK 和超声波传感器 HC-SR04 均可用于环境信息检测，属于外部传感器^[8]，二者的作用距离不同，主要用于避碰控制，见图 2。

2.4. 传感器检测原理

1) 红外传感器

根据光线反射原理，红外传感器模块检测前进方向上是否存在障碍物。对于物体透明导致无法检测的情形，可由超声波传感器补充。红外传感器 E18-D80NK 集发射与接收于一体，属于漫反射型光电传感器，NPN 常开型输出，测量范围 3~80 cm。传感器尾部安装有一个电位器，可以通过调节电位器改变检测距离。前方无障碍时，输出高电平；有障碍时，输出低电平。传感器与微处理器的外部中断接口相连，检测到障碍物时，传感器输出以外部中断的方式通知微处理器。

2) 超声波传感器

超声波传感器利用声音在空气中的传播速度和传播时间计算出与障碍物距离^[9,10]，所依据的关系是：

$$\text{检测距离} = \frac{1}{2} \times \text{声音传播速度} \times \text{飞行时间} \quad (1)$$

超声波传感器模块 HC-SR04 如图 3 所示，它提供 2 cm~300 cm 的非接触式测距功能，精度可达 3 mm。模块包括超声波发射器、接收器与控制电路。测距由控制端口 TRIG 触发，微处理器提供至少 10 μs 的高电平。超声波发射器自动发射 8 个 40 kHz 的超声波，同时 ECHO 端口向微处理器输出一个高电平信号，微处理器接收到 ECHO 端口的高电平时，定时器开始计数。当接收器检测到返回的超声波，ECHO 端口输出低电平，微处理器的定时器计数停止。而微处理器 S3C2440 的定时器是减计数，减少的计数值就是超声波从发射到返回的时间。如果超声波接收器没有接收到返回的超声波，ECHO 端口经过一段时间自动

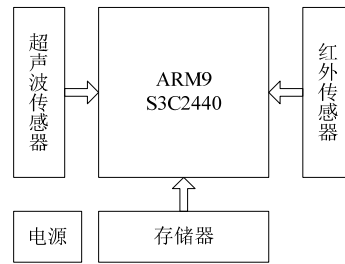


Figure 2. Proximity sensor for mobile robot
图 2. 移动机器人接近传感器的配置示意



Figure 3. Ultrasonic sensor module
图 3. 超声波传感器模块

变为低电平^[11]。

3. 驱动程序设计

Linux 是一款多任务操作系统。由于具有开源性、低成本、多平台支持等特点，在嵌入式方面获得了广泛应用^[12]。基于 ARM9-Linux 嵌入式系统的移动机器人控制器，需要通过设备驱动程序来操作 I/O 设备。作为 I/O 设备的一种，传感器的驱动体系结构如图 4 所示。

3.1. 驱动程序分类

Linux 中使用的基本设备驱动程序，分为网络设备、块设备、字符设备^[13]等几种。块设备驱动不接受应用程序的直接调用，而是以文件系统的形式与应用程序连接，文件系统的硬件采用块单位的处理方式；网络设备是与网络层连接的设备，网络设备驱动程序没有设备文件，不能被应用程序直接使用，只有 ifconfig 等特殊程序能够利用系统呼叫方式来使用；字符设备程序可由应用程序直接呼叫，而应用程序以普通文件方式处理设备文件，达到硬件控制的目的。多数硬件表现为字符设备，移动机器人配置的红外传感器、超声波传感器均属于字符设备。操作设备时，应用程序使用低级文件输入输出函数来请求设备文件，内核通过注册的字符设备驱动程序的 file_operations 结构体信息，调用相应设备文件的函数。字符设备驱

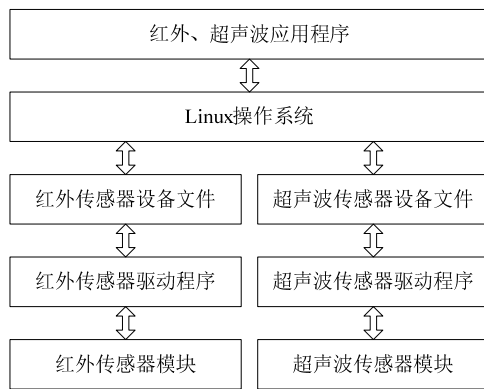


Figure 4. Architecture of device driver
图 4. 设备驱动体系结构

动程序的代码结构，包括注册、注销字符设备驱动程序的模块初始化函数；注册到对应低级文件输入输出的 `file_operations` 结构体的函数；对于使用中断功能的硬件，用中断服务函数^[14]加以处理。

3.2. 红外传感器驱动

红外传感器输出连接微处理器 S3C2440 的 GPF3/EINT3 引脚。根据红外传感器模块 E18-D80NK 的工作原理，需要使用中断方式来处理任务。表现在字符设备驱动程序里，即需要定义中断服务函数 `irq_interrupt()`，同时在设备文件打开函数 `xxx_open()` 中，要使用 `request_irq()` 函数把中断号、中断服务函数注册到系统的中断信息列表中，处理流程如图 5 所示。

这样，当红外传感器设备驱动载入内核，应用程序打开并使用 `read()` 函数读取接口状态，并把状态数据由内核传回用户空间。`read()` 函数关键代码及结构如下：

```

static int sc2440_irq_read(struct file *filp,
char_user *buff, size_t count, loff_t *offp){
wait_event_interruptible(sc2440_waitq, ev_irq);
//等待 GPF3 引脚中断
ev_irq = 0;
err = copy_to_user(buff, ( const void *)
(&value ), min(sizeof(value), count));
//中断发生后，状态值发回用户空间
return err ? -EFAULT : min(sizeof(value), count);}
  
```

当红外传感器检测到障碍物存在时，GPF3/EINT3 引脚相应检测到由高到低的电平变化，将产生中断号为 `IRQ_EINT3` 的中断，系统调用相应的中断服务函

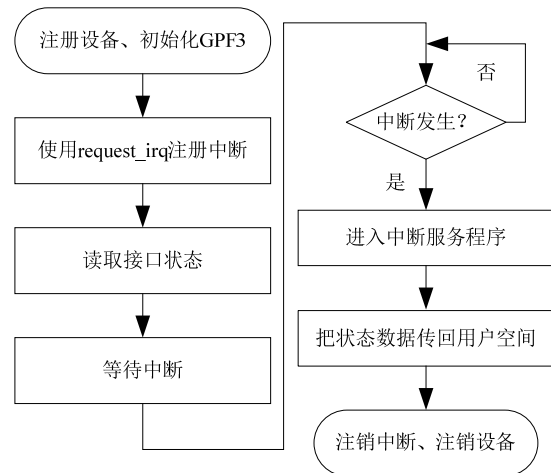


Figure 5. Flow chart of infrared sensor device driver
图 5. 红外传感器设备驱动流程图

数进行避碰处理，中断服务算法流程如图 6 所示。

关键代码及结构如下：

```

static irqreturn_t irq_interrupt(int irq, void
*dev_id ){
struct sc2440_irq_desc *sc2440_irqs = (struct
sc2440_irq_desc * ) dev_id;
int stat;
stat = s3c2410_gpio_getpin(sc2440_irqs-> pin);
//读取引脚状态
if ( stat ==0 ){
//判断是否为低电平
value = 0;
//保存状态值
ev_press = 1;
wake_up_interruptible( &sc2440_waitq );}
//唤醒进程
return IRQ_RETVAL( IRQ_HANDLED );}
//退出中断服务
  
```

3.3. 超声波传感器驱动

根据超声波传感器模块 HC-SR04 的工作原理，传感器的 TRIG 接口与微处理器 S3C2440 的接口 GPB9 连接，而传感器的接口 ECHO 连接微处理器 S3C2440 的接口 GPF6/EINT6。设备驱动程序要直接操作 GPIO 的寄存器，字符设备驱动使用 `xxx_ioctl()` 函数操作具体的硬件设备，操作流程如图 7 所示。

驱动程序操作微处理器的接口 GPB9 产生 10 μ s

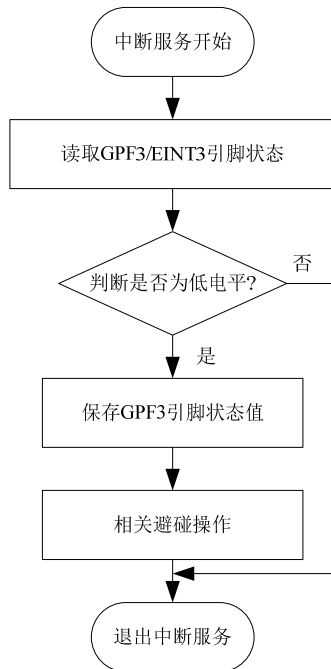


Figure 6. Flow of interrupt service processes of infrared sensor
图 6. 红外传感器中断服务流程

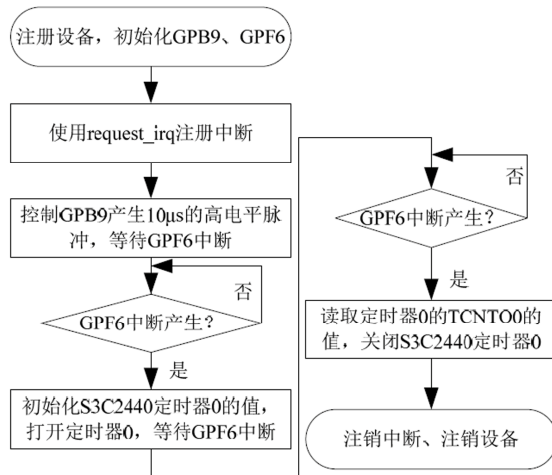


Figure 7. Flow chart of ultrasonic sensor device driver
图 7. 超声波传感器设备驱动流程图

的高电平脉冲, 启动超声波。当接口 ECHO 检测到由低到高的电平变换, 使用中断方式打开处理器 S3C2440 的定时器 0 开始计数。而当 ECHO 检测到由高到低的电平变换, 则使用中断方式读取定时器 0 的 TCNT00 值, 关闭定时器 0。处理器的定时器是减法计数, 定时器初始的装载值减去 TCNT00 的值, 即 ECHO 端口的高电平维持时间, 根据式(1)即可计算得到超声波传感器的测量距离。

ioctl()函数的关键代码及结构:

```

static int tq2440_pwm_ioctl(struct inode *inode,
struct file *file, unsigned int cmd, unsigned long
arg){...
switch( cmd){
case 0:...
case 1:...
s3c2410_gpio_setpin( gpio_table[0], 1);
udelay( 10 );
//GPB9/TRIG 10 μs 高电平脉冲
s3c2410_gpio_setpin(gpio_table[0], 0);
wait_event_interruptible(button_waitq, high);
//等待 ECHO 高电平状态
wait_event_interruptible(button_waitq, low);
//等待 ECHO 低电平状态
tconon=_raw_readl(S3C2410_TCNT0(0));
//读取 TCNT0
tcon = _raw_readl(S3C2410_TCON);
tcon &= ~0x1f;
_raw_writel(tcon, S3C2410_TCON);
//关闭定时器 0
break;}
return 0;}

```

中断服务操作的算法流程如图 8 所示。

部分关键代码及结构:

```

static irqreturn_t irq_interrupt(int irq, void
*dev_id ){...
//中断服务函数, 响应超声波传感器 ECHO
down = !s3c2410_gpio_getpin(button_irqs->pin);
if( down == 0 ) {
itcon = __raw_readl(S3C2410_TCON);
itcon |= 0x1;
__raw_writel(itcon, S3C2410_TCON);
//定时器 0 开始计数
high = 1; ev_press = 1;
wake_up_interruptible(&button_waitq);}
//唤醒进程
else{
low = 1;ev_press = 1;
wake_up_interruptible(&button_waitq);}
//唤醒进程
return IRQ_RETVAL(IRQ_HANDLED);}

```

4. 测试

为检验驱动程序设计的正确性,须在驱动程序设计完成后进行测试。与超声波传感器驱动程序相比,红外传感器只使用 `read()` 函数读取接口状态,而超声波传感器的驱动程序操作略显复杂。下面以超声波传感器为例,按图 9 所示的流程编写测试程序,进行功能测试。

操作设备的应用程序,依次使用 `open()` 函数打开设备文件,用 `ioctl()` 函数操作相应的设备,用 `read()` 函数读取定时器 TCNT00 的值,最后用 `close()` 函数关闭设备文件。

主要代码及结构如下:

```
int main(void){
    buttons_fd = open("/dev/SH-OMG", 0);
    //打开设备文件
    ioctl(buttons_fd, 1);
    //启动测距
    read(buttons_fd,&atconon,sizeof(atconon));
    //读取时间值
    printf("the distance is 0x%x.\n", atconon);
    temp = (31250 - atconon) * 173/125;
    //用检测模型计算距离,单位 mm
    printf("the distance is %d mm.\n", temp);}
```

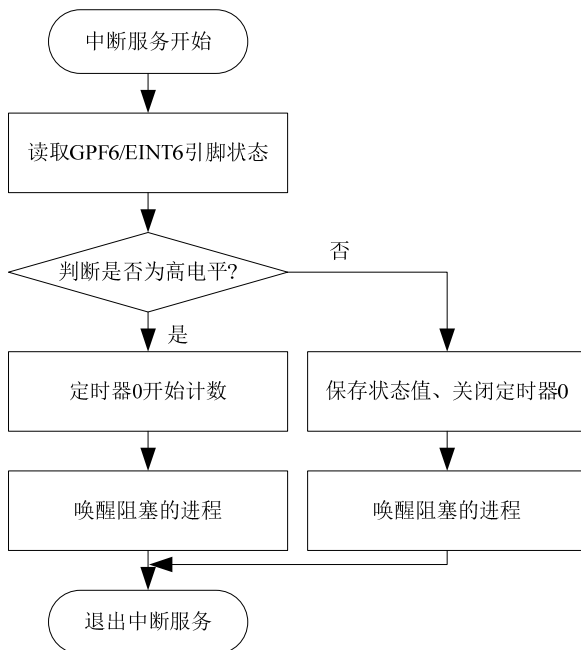


Figure 8. Flow of interrupt service processes of ultrasonic sensor
图 8. 超声波传感器中断服务流程

//终端显示

```
close(buttons_fd);
//关闭设备文件
return 0;}
```

把超声波设备驱动程序以可加载模块方式编译,使用 `insmod` 命令加载进 Linux 内核中,运行测试程序。以不同的距离遮挡超声波传感器,在串口终端上显示出经过处理的距离数据,见图 10。通过与实际距离比较,可知超声波传感器及驱动程序工作正常。这表明,超声波传感器的检测数据能够正确读取并输入控制器,驱动程序的设计能够满足使用要求。

5. 结束语

通过分析移动机器人配置的接近传感器的工作原理,确定传感器与微处理器的连接方式。然后,设计相应的 Linux 设备驱动程序,以操作红外、超声波传感器硬件设备。由于字符设备驱动程序结构上的相

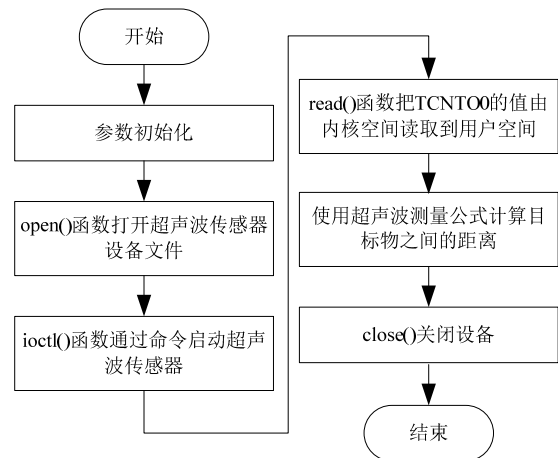


Figure 9. Testing process of ultrasonic sensor
图 9. 超声波传感器测试流程

```
[root@Embedsky rootfs]# ./app_sh1
qianhoustart:
the tcnton is 0x79ac.
the distance is 141 mm.
the tcnton is 0x79ae.
the distance is 138 mm.
the tcnton is 0x798b.
the distance is 186 mm.
the tcnton is 0x7988.
the distance is 190 mm.
the tcnton is 0x797d.
the distance is 206 mm.
the tcnton is 0x797e.
the distance is 204 mm.
```

Figure 10. Screenshot of reading ultrasonic sensor data
图 10. 读取超声波传感器的测试数据截图

似性, 本文只给出典型配置接近传感器的 Linux 设备驱动程序设计方法, 其他功能模块的驱动程序可相应设计。

参考文献 (References)

- [1] 张玫, 邱钊鹏, 诸刚 (2011) 机器人技术. 机械工业出版社, 北京, 1-2.
- [2] 张晓伟, 刘盼盼 (2008) 嵌入式系统设计原理与开发实例. 电子工业出版社, 北京, 3-5.
- [3] Wang, S., Zhang, H., Tan, H.-Q. And Jiang L.-Y. (2012) Implementation of step motor control under embedded Linux based on S3C2440. *Energy Procedia*, **16**, 1541-1546.
- [4] Miah, S. and Gueaieb, W. (2011) Mobile robot navigation using direction-sensitive RFID reader. *Control and Intelligent Systems*, **39**, 254-264.
- [5] 陈剑斌, 田联房, 王孝洪 (2011) 基于 ARM 的自主移动机器人控制系统设计. *机械设计与制造*, **6**, 160-162.
- [6] Sung, A., Choi, B. and Shin. S. (2007) An interface test model for hardware-dependent software and embedded OS API of the embedded system. *Computer Standards & Interfaces*, **29**, 430-443.
- [7] 张玫 (2012) ARM-Linux 嵌入式系统在轮式移动机器人上的应用. *机电产品创新*, **25**, 26-27.
- [8] 王文庆, 张涛, 龚娜 (2013) 基于多传感器融合的自主移动机器人测距系统. *计算机测量与控制*, **21**, 343-345.
- [9] Sv, N. and Ch R. (2007) Calibration and interface of a polaroid ultrasonic sensor for mobile robots. *Sensors and Actuators A: Physical*, **135**, 169-178.
- [10] Ghazali, M.Z., Noor, N.M., Ramly, N. and Putit. S. (2012) Development of microcontroller based mobile gas sensing robot. *Engineering Procedia*, **41**, 1190-1196.
- [11] 单承刚 (2009) 嵌入式 Linux 下超声波避障系统设计. *计算机技术与发展*, **19**, 29-31.
- [12] 黄建明, 刘宝林, 徐秀林 (2011) 基于 S3C2440 多路 A/D 转换 Linux 驱动程序设计. *微计算机信息*, **27**, 110-111.
- [13] 陈涛, 余学才, 朱良销等 (2010) 信号发生模块的 Linux 驱动程序设计. *化工自动化及仪表*, **37**, 88-91.
- [14] 俞永昌 (2008) 设备驱动开发技术及应用. 人民邮电出版社, 北京, 235-245.