

DOI:10.3969/j.issn.1001-4551.2013.11.026

基于 Mobile-Android 小型移动 机器人平台控制系统

李 瑞, 李晓明*

(浙江理工大学 机械与自动控制学院, 浙江 杭州 310018)

摘要:针对目前小型移动机器人控制系统存在的设计成本高、平台搭建困难、过度依赖网络的问题,将软件设计模式中的 MVC 模式应用于移动机器人控制系统设计,同时,在 Android 平台和 Arduino 控制板上使用 Java、C 语言进行了实验。提出了基于 Android 移动计算平台、结合蓝牙通信和无线局域网(WiFi)通信的小型智能移动机器人平台 Mobile-Android。详细地描述了该平台的系统结构、控制软件,包括 GUI 界面模块、运动控制模块、命令配置模块、上位机与下位机通信协议等的设计与实现。在原型系统上初步的测试结果表明,该智能移动平台方案既能够实现智能移动机器人系统硬件平台的快速搭建,又能够充分发挥移动机器人的运动特性与智能手机的强大信息处理能力和控制能力;结合两者长处,获得了一个完整的智能移动机器人系统。

关键词:Android; 蓝牙; 移动机器人; 控制系统

中图分类号:TP242

文献标志码:A

文章编号:1001-4551(2013)11-1414-06

Control system of miniature mobile robotic platform based on Mobile-Android

LI Rui, LI Xiao-ming

(Faculty of Mechanical Engineering & Automation, Zhejiang Sci-Tech University, Hangzhou 310018, China)

Abstract: Aiming at solving problems that the current high cost of small mobile robot control system, difficulty of platform building and excessive dependence on network, the MVC pattern in software design pattern was applied to mobile robot control system design, at the same time, an experiment using C and Java language was carried on Android platform and Arduino board. A small mobile robot platform named Mobile-Android based on Android was put forward. Bluetooth and WiFi communication was applied to the platform. The system structure of the platform and controlling software was described in detail, including GUI module, motion controlling module, orders configuring module and data communication protocol between upper computer and slave machine. The test results of prototype system indicate that this intelligent mobile platform schema can not only build up hardware platform quickly, but bring the movement characteristics of mobile robot and the powerful processing and control ability of the intelligent mobile phone into full play. Combining both of their advantages, a complete intelligent mobile robot system is obtained.

Key words: Android; bluetooth; mobile robot; control system

0 引 言

目前,相对成熟的机器人控制方式主要有触摸屏操控^[1]、基于 GSM/GPRS 网络的远程操控^[2]、基于 Internet 的 PC 机遥控^[3]、基于 WiFi 无线网络的 PC 操控^[4]、红外线遥控^[5]以及语音控制^[6]等,这些操控方

式基本满足了现有机器人的应用需求,但是随着生活水平的提高,人们对生活品质的要求也随之提高,便利、舒适、智能化已经成为家居、办公系统的重要概念^[7]。上述几种控制方式受限于周围是否有网络、PC 便携性较差或是技术尚未成熟(语音控制)等各种因素。人们期望着能实现一种既方便携带又不受网络限

收稿日期:2013-06-28

作者简介:李 瑞(1987-),男,山东烟台人,主要从事 Android 应用、移动机器人方面的研究. E-mail: liruikqn@126.com

通信联系人:李晓明,男,副教授,硕士生导师. E-mail: origin@foxmail.com

制的一种机器人控制系统。

正因此,本研究基于业已成熟的蓝牙技术,针对目前 Android 在移动互联网市场份额的持续增长,实现基于 Android 蓝牙通信的机器人控制系统。该控制系统由 GUI 模块(包括主控模块,设备发现模块,命令配置模块)、控制模块、服务器模块和蓝牙业务等几部分组成,实现方式模块化,各模块各司其职,代码清晰。且该系统成本较低,只需一部 Android 手机和相应的蓝牙模块。经在具有蓝牙从机模块的移动机器人上实验,结果表明:该系统能够实现控制机器人运动的既定目标,且同时实现作为服务器进行移动控制的目的。

1 系统的体系结构设计

该系统由上位机和下位机两部分组成,其中上位机采用带蓝牙通讯功能的智能手机(在该系统中选用安卓操作系统的智能手机),使用 Java 语言开发 Android 应用作为移动机器人的控制软件;下位机采用了带蓝牙通讯模块的典型移动运动平台,包括运动控制单元、电机及其驱动。上位机和下位机之间采用蓝牙通讯。其系统结构如图 1 所示。

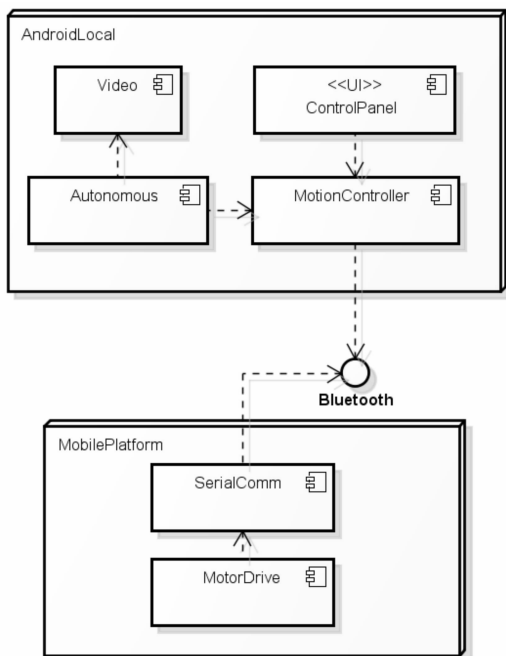


图 1 Mobile Android 系统结构设计

该系统设计方案的优点在于,最大程度上保证了移动机器人系统的运动模块和控制模块的独立性,同时又能充分发挥各自的优势。例如:可以自由升级运动平台,从普通智能小车升级到专业的移动机器人平台;可以充分利用智能手机集成的硬件资源,包括 WIFI 通讯、GPS 与网络定位、加速度和重力传感器、人机界面、摄像头等以及强大的计算能力和软件资源。

该系统核心在于上位机和下位机之间的通讯,在本研究的原型系统中采用蓝牙作为无线通讯的手段。原因在于:①蓝牙串口数传模块成本低,便于对底层运动平台进行扩展;②大多数智能手机均配有蓝牙通讯功能,且标准安卓系统中均支持基于蓝牙的串口通讯功能;③蓝牙通讯距离较短,且为一对一的通讯方式,可以避免远距离时对机器的误操作。

下位机硬件设计较为简单,由于仅仅需要完成运动控制的功能,下位机可以采用单片机配合专门的运动驱动电路实现对电机的控制,从而完成运动平台的行走功能。现有实现方案较多,此处不再赘述。下位机在软件设计上采用串口命令服务器的方式,即不断接收来自串口的运动控制指令,根据具体的命令对电机进行控制。其命令格式如表 1 所示。

表 1 通信协议格式

起始	进给	电机号	方向	速度值	结束
@	j	(motor_id)	[+ -]	(speed_value)	%
@	h	%			

该通讯协议简单明了,第一条指令控制某个电机的具体转速,对于大多数两轮驱动的运动平台而言,需要连续发送两条电机控制指令;第二条指令用来控制所有电机停机,用于应对紧急情况。

2 上位机设计

2.1 上位机软件的总体架构

该控制系统中,一部 Android 手机作为控制器,可通过蓝牙模块直接控制移动机器人;同时该手机也可作为服务器置于移动机器人上,通过 WiFi 通信方式接收客户端的指令,收到指令后通过后台运行的 Service,将客户端的指令发送给下位机,实现远程控制。系统的软件总体框架如图 2 所示。参考了 Android 用户界面框架(Android UI Framework) MVC 模型的思想^[8],采用分层结构,具体分为:表示层、控制层、业务层,分别对应了 MVC 设计模式中的 View、Controller 和 Model。

表示层向用户展示模型的状态,在该控制系统中,是以每个 Activity 对应的页面所呈现,为使界面与控制逻辑分开,其布局以 layout 文件夹下的 XML 文件承载,当然 Android 也允许以 Java 代码方式组织 GUI。控制层负责与业务层的流程控制,一方面将表示层的调用发到业务层请求处理;另一方面也负责与 WiFi 客户端通信。业务层封装了蓝牙数据传输的核心操作,如创建 BluetoothSocket 建立 Socket 通道、传输数据等。

整个控制系统采用这种设计模式,对代码进行了

分层,程序结构更加清晰、代码更加健壮,提高了模块化程度,各模块职责明确,高效率处理相应事件。

该控制系统主要用到的 Android 系统组件为 Activity、Service 和 BroadcastReceiver。Activity 和 Service 及 Activity 之间的消息传递机制使用了 Intent,通过发送 Android 系统级和应用程序级广播的形式将 Intent 发送出去,同时注册了动态 BroadcastReceiver 用以接收消息,并做相应处理。消息处理机制是在主进程中创建了 Handler 对象,用以处理接收到的消息,如界面更新等。

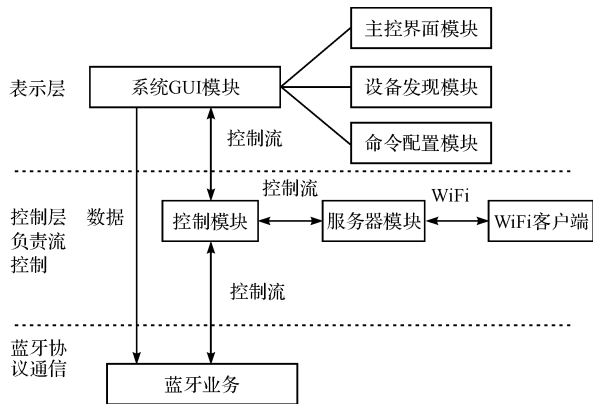


图 2 上位机软件总体框架

2.2 实现蓝牙通信

Android 平台从 2.0 版本开始支持蓝牙协议栈,其应用框架层提供了对蓝牙进行操作的 API,通过这些 API 可实现端到端(peer to peer)的无线连接。本研究用到的 API 用于实现以下功能:

- (1) 查找周围的蓝牙设备;
- (2) 查询与本地蓝牙适配器配对的设备;
- (3) 建立 RFCOMM 信道;
- (4) 通过 Socket 在蓝牙设备之间传输数据;
- (5) 管理蓝牙连接^[9]。

具体使用以下 API:

(1)BluetoothAdapter:该类代表本地蓝牙适配器,是蓝牙交互的入口点。使用该类可以发现其他蓝牙设备,查询已绑定的设备,使用 MAC 地址实例化一个蓝牙设备。

(2)BluetoothDevice:该类代表了一个远端蓝牙设备,使用它可以请求连接到远端蓝牙设备并获取远端蓝牙设备的名称、MAC 地址和绑定状态。

(3)BluetoothServerSocket:该类代表蓝牙服务器端的 Socket,监听来自远程的连接请求。两个蓝牙设备相连,必有一个设备作为服务器打开一个服务器端套接字,当远端蓝牙设备发起连接请求并连接成功时,该

类返回一个 BluetoothSocket 对象。

(4)BluetoothSocket:该类代表蓝牙套接字,是应用程序获得输入输出流与其他设备通信的连接点。

2.3 GUI 模块

本次开发是基于 Android SDK2.2(API 8)平台,最小 SDK 和目标 SDK 都是 Android SDK2.2,如下所示:

```
<uses-sdk
    android:minSdkVersion="8"
    android:targetSdkVersion="8" />
```

本研究所使用的 Android 手机版本为 Android 2.2。开发环境使用 Eclipse 3.7.0,同时使用了用于 Android 开发的 ADT 插件,版本为 21.0.0。要获得 Android Bluetooth 的使用权限,必须在 AndroidManifest 文件中加入权限声明,如下所示:

```
<uses-permission android:name="android.permission.
BLUETOOTH"/>
<uses-permission android:name="android.permission.
BLUETOOTH_ADMIN"/>
```

前者为获得蓝牙开发权限,后者是获得蓝牙的管理员权限^[10]。

控制系统界面如图 3 所示,除了控制器界面作为主显界面外,其他界面通过 OptionsMenu 的菜单项进行启动。用户进入如图 3 左图所示的界面后,点击手机的 menu 按钮,会弹出如图 3 右图所示的菜单选项,在这里用户可以选择相应的交互界面。



图 3 控制系统界面

用户点击“查找周围设备”菜单项会开始查找周围的蓝牙设备,该过程通常持续 12 s。查找结束后,弹出对话框如图 4 所示。

该对话框实际是一个 Activity,在向 AndroidManifest 文件中注册该 Activity 时,声明其 android:theme="@android:style/Theme.Dialog"。从而呈现出一种对话框效果。该 Activity 里包含两个 ListView 控件,一个用于显示已经配对过的蓝牙设备,另一个用于显示最新查找到但还未配对过的蓝牙设备。图 4 中共有 4 个



图 4 选择设备对话框

蓝牙设备:“RAYLEE-LAPTOP”,“TOROBOT”(本研究的蓝牙机器人)是配对过的设备,显示在上部的 pairedDevicesListView (ListView 对象)中;下部的 newDevicesListView (ListView 对象)显示的“可嘉”,“WindowCE”是最新查找到的蓝牙设备,未配对过。每个蓝牙设备名称下面都显示了其 MAC 地址供用户识别选择,用户点击移动机器人蓝牙模块名称则可实现与其配对,这是控制移动机器人的第一步。

用户点击图 3 中“配置命令”菜单项将弹出的界面如图 5 所示。在该界面中,用户可以输入预先设置好的下位机控制指令。通过该界面输入的指令将保存在 Android 平台的 SharedPreferences 中,SharedPreferences 产生的文件存储在 /data/data/ < package name > /shared_prefs 目录下,该文件是一个 xml 格式文件。

本研究实现的系统亦可作为服务器,实现远程控制和视频传输。实现服务器模式时,用户可点击图 3 中“启动服务器模式”菜单项,启动界面如图 6 所示。



图 5 配置对话框



图 6 服务器模式

进入服务器模式后,用户可输入端口号,点击“开启服务器”按钮将在输入的端口号上监听。如果用户输入

的端口号被占用,该系统将自动选择未被占用的端口号在其上监听,将监听信息显示在 TextView 控件中,上部的 SurfaceView 控件用于预览视频并压缩后传递给客户端,另一台客户端手机可通过 WiFi 连接到该服务器,基于 TCP 协议通过 Socket 实现远程控制。

2.4 控制模块

如图 3 左图所示,界面由上、下、左、右 4 个 ImageButton 和一个 SeekBar 组成,按钮用于控制移动机器人的运动,SeekBar 用于调节前进、后退时电机的转速,避免了用户调节速度时必须回到“配置对话框”界面的不便,用户只需拖动进度条即可控制机器人移动的速度。该过程使用了事件监听模型,以前进按钮为例,流程图如图 7 所示。

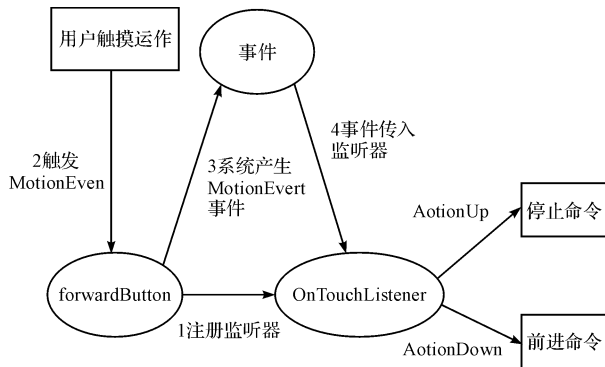


图 7 事件处理流程

forwardButton 添加了 OnTouchListener,当用户触碰前进按钮时,Android 系统将 MotionEvent 事件传递给前进按钮的监听器,在 OnTouchListener 中对 MotionEvent.ActionUp, MotionEvent.ActionDown 进行判断,若为 ActionUp 则发送停止指令“@ h%”;若为 ActionDown 则发送前进指令“@ j#1 + 20% @ j#2 + 20%”。指令发送通过 BluetoothCar (Activity) 类中的 sendMessage (String) 方法进行,该类中保存了指令的全局变量,在该类初始化时会调用 onShow () 方法对声明的指令进行初始化。在 onShow () 方法中获得命令配置模块创建的 SharedPreferences 对象,从该对象中获得用户在图 5 中保存的命令。

该系统作为服务器时,用户首先点击图 3 中“启动服务器模式”菜单项,通过发送一个 Intent 对象打开图 6 所示的界面,用户在该界面配置好监听的端口并点击“开启服务器”按钮后,程序会以 bindService 的方式开启一个后台服务,用于将客户端的命令转发给蓝牙机器人;同时也会开始一个 serverThread 线程,在该线程中监听来自客户端的连接请求,请求连接成功后,通过 Socket 对象获取字节流,并用字符流进行包装。然后系统在一个循环中持续从该流中读取客户端发送

过来的指令。获取指令后本研究通过一个后台运行的 Service 将指令转发给蓝牙机器人;该 Service 具有同 BluetoothCar 类相似的功能,也能实现与远程蓝牙设备的连接,获取通信信道。并能判断客户端发来的是何种指令。例如,如果客户端发来的是“forward”这样一个字符串,则该 Service 知道用户的需求是要机器人前进,于是,该 Service 会将存储在 SharedPreferences 中的前进指令取出,并转发给蓝牙机器人。本研究之所以用 Service 组件,而不是通过发送广播的方式调用 BluetoothCar 中的指令发送方法,是因为现在用户所在的界面是图 6 所示的服务器模式,BluetoothCar 所在的界面被完全遮挡了,而 Android 平台下,一次只允许显示一个 Activity,当某个 Activity 不可见后,该 Activity 将进入其 onStop() 生命周期,一旦系统内存不足,这种 Activity 将首先被系统杀死,所以后一种方法不可行。而 Service 组件则不同,其是 Android 提供的专门在后台长时间运行的无界面组件,而且 Service 的优先级比 Activity 高,所以使用 Service 是符合设计预期的,并且运行结果也确实如此。Service 有两种启动方式: startService() 和 bindService(), 两者区别在于,前者不能获得 Service 的实例,以前者启动的 Service 必须具备自我管理功能;后者可以获得 Service 实例。本研究使用 Service 的主要目的是让 Service 提供一个命令转发的方法,需要获得其实例,才能使用该方法。因此系统应该以 bindService() 方式启动。

服务器界面中 TextView 的更新是通过该 Activity 中 Handler 对象进行更新,消息更新涉及客户端与服务器的连接状态及收到指令后的处理,消息处理流程如图 8 所示。

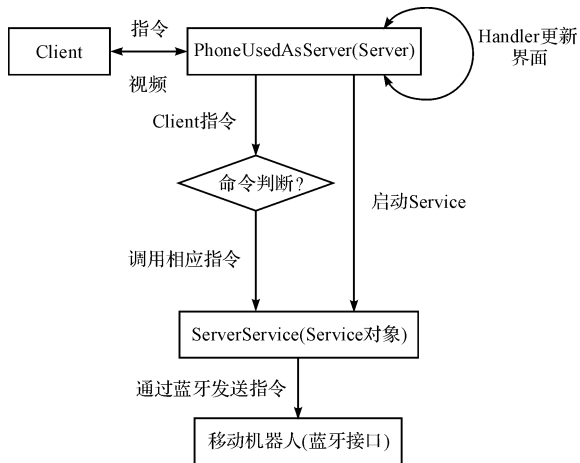


图 8 服务器消息处理流程

2.5 命令配置模块

该模块主要用来实现对移动机器人控制命令的配

置。Android 为开发人员提供了方便的数据存储方法 SharedPreferences。这是一种轻量级的数据保存方式,通过 SharedPreferences,开发人员可以将 NVP(Name/ Value Pair, 名称/值对)保存在 Android 的文件系统中,而且 SharedPreferences 完全屏蔽了对文件系统的操作过程,开发人员仅通过调用 SharedPreferences 中的函数就可以实现对 NVP 的保存和读取。

本研究在 ControlConfig (Activity) 类中通过 getSharedPreferences () 获得一个名为“BluetoothCar”的 SharedPreferences 对象。使用该对象的 putString (String , String) 方法将用户在图 5 界面中输入的指令以键值对的形式保存在 ShardPreferences 中。通过 SharedPreferences 对象的 edit () 方法获得接口 ShardPreferences. Editor 的对象,SharedPreferences. Editor 对象用于提交保存内容只需调用其 commit () 或 apply () 方法即可。这样当用户每次启动程序时,若用户不配置命令则默认读取用户上次配置的命令,并在图 5 界面中进行显示,以便用户查看;若移动机器人的命令有变化则用户只需在图 5 中输入新命令即可,相应的 SharedPreferences 对象的文件将更新。例如根据前述下位机的指令格式可以在“前进”文本框中输入“@ j# 1 + 30% @ j# 2 + 30”,这样当下位机指令变化时只需在该界面中进行更改即可。

3 结果及结论

整套系统设计完成后,本研究在有 WiFi 热点的场所进行了测试(服务器模式需要借助网络测试)。首先进行直接控制测试。测试者打开程序后,首先在菜单选项中选中“配置命令”进行系统命令的配置,配置后点“保存”,将命令保存到系统中。接下来,测试者开启蓝牙搜索并与移动机器人配对,配对完成后即可进行运动控制。系统启动与蓝牙配对界面如图 9 所示。



图 9 系统启动与蓝牙配对界面

当成功与蓝牙机器人配对后,主控界面顶部的 Title 右侧会显示“已连接到蓝牙机器人”,提示用户控制器已与蓝牙机器人连接成功,可以进行运动控制。同

时当开启服务器模式时,系统会在用户给定的端口上监听。用户的操作信息都会及时显示 EditText 控件中,服务器模式界面如图 10 所示。



图 10 服务器模式界面

当未连接时蓝牙模块指示灯一直闪烁,连接成功后指示灯将不再闪烁,两种控制画面如图 11 所示。

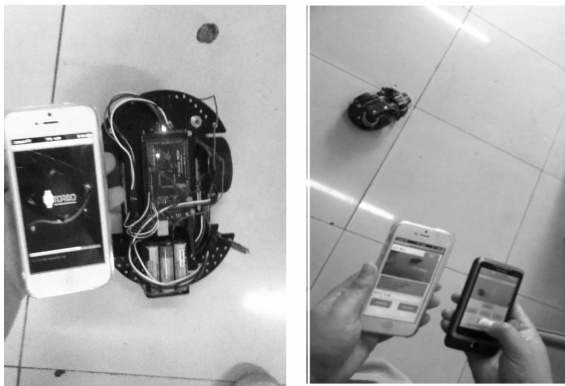


图 11 两种控制方式

由于蓝牙技术本身传输距离有限,经过现场测试,本研究的控制系统在 10 m 左右范围内能顺利达到预期目的,实现蓝牙机器人的运动控制,远程控制方式下借助视频传输可以实现远程控制。

4 结束语

Android 因其性能优异、平台开放的特点,在智能手机、平板电脑、车载导航仪和数字电视等领域获得普

遍应用。本研究提出了一种基于 Android 平台的移动机器人控制系统的设计和实现,分析了各个功能模块的设计和具体实现方法。测试结果表明,该控制系统操作简单,运行稳定,在十几米范围内能实现有效控制。可以预见,该控制系统可以应用于家用机器人上,解决目前家用机器人操作不方便、便携性差、使用成本高等问题,而且蓝牙十几米的传输范围在一个家庭范围内基本够用,因此本研究在家用机器人领域有较为广阔的应用前景。

参考文献(References):

- [1] FANG Zheng, TONG Guo-feng, HUA Yi-min. An Em-bedded Platform for Intelligent Mobile Robot[C]// Proceedings of the 6th World Congress on Intelligent Control and Automation. Dalian: IEEE Press, 2006: 9404-9108.
- [2] 李瑞峰,顾义坤. 基于 GSM/GPRS 网络的家用机器人远程监控系统设计[J]. 制造业自动化, 2006, 28(10): 71-74.
- [3] 陆建飞,赵福臣. 基于 Internet 的机器人遥控操作研究[J]. 机械工程师, 2008(8): 119-121.
- [4] GUPTA G S, MUKHOPADHYAY S C, FRENCH J R. Wireless communications and control module of a Web-enabled robot for distributed sensing applications[C]// IEEE International Instrumentation and Measurement Technology Conference: I2MTC 2008. Victoria, Canada: IEEE, 2008: 393-398.
- [5] 钟伟浩,肖南峰. 家庭服务机器人红外线遥控系统的设计与实现[J]. 低压电器, 2007(22): 19-22.
- [6] 李瑞峰,李 麟. 基于 RSC4128 的家用机器人语音人机交互系统的设计[J]. 制造业自动化, 2007, 29(10): 31-33.
- [7] 马 鸣,张 华. 基于 Android 的蓝牙远程控制 PC 系统的设计与实现[J]. 电脑知识与技术, 2012, 28(8): 6833-6835, 6952.
- [8] 王向辉,张国印,赖明珠. Android 应用程序开发[M]. 2 版. 北京:清华大学出版社, 2012.
- [9] 燕 琦. 基于 Android 的蓝牙智能无线控制系统设计[J]. 电脑编程技巧与维护, 2012(20): 95-96.
- [10] 侯新琦,李 佳. 基于 Android 蓝牙通讯的研究[J]. 电子世界, 2012(11): 84-85.

[编辑:李 辉]

本文引用格式:

李 瑞,李晓明. 基于 Mobile-Android 小型移动机器人平台控制系统[J]. 机电工程, 2013, 30(11): 1414 - 1419.

LI Rui, LI Xiao-ming. Control system of miniature mobile robotic platform based on Mobile-Android[J]. Journal of Mechanical & Electrical Engineering, 2013, 30(11): 1414 - 1419.

《机电工程》杂志: <http://www.meem.com.cn>