

基于 STM32 的 USB HID 接口多功能控制器设计

王晓东

(唐山学院 计算机科学与技术系, 河北 唐山 063000)

摘要:在 STM32 处理器平台上使用 USB HID 接口协议设计了一种低成本的多功能控制器, 控制器通过 USB 接口收到上位机指令后可进行变频、变占空比的 PWM 信号输出、GPIO 读入和输出、模拟量 AD 采集等多种功能; 上位机软件采用一组通用的 HID 接口的 API 函数实现对下位机硬件的读写控制, 而且免驱安装, 能显著降低开发成本。实验验证的结果表明, 系统可以稳定运行。

关键词:通用串行总线; 人机接口设备; 控制器; STM32 处理器

中图分类号:TP368.2 **文献标志码:**A **文章编号:**1672-349X(2020)03-0021-06

DOI:10.16160/j.cnki.tsxyxb.2020.03.005

Design of Multi-Purpose Controller with USB HID Based on STM32

WANG Xiao-dong

(Department of Computer Science and Technology, Tangshan University, Tangshan 063000, China)

Abstract: On the platform of STM32, a multi-purpose controller is designed with USB HID protocols, which can change the frequency, output the PWM with the changed duty ratio, read and output the GPIO, collect the analog quantity AD data and etc. after receiving the command from the master computer through the USB. The software of the master computer applies a group of API functions of the universal HID to control the reading and writing of the slave computer, which can significantly reduce the development cost with no driver needed. The experimental results show that the system can run stably.

Key Words: Universal Serial Bus (USB); Human Interface Device (HID); controller; STM32 processor

1 概述

USB 总线协议是由 Intel, Compaq, Microsoft 等七家公司共同制定的串行接口标准^[1], 协议内部包含完善的主从机通讯机制, 支持从高速视频传输到低速查询传输多种通讯模式, 传输速率高, 从 USB1.1 协议发展到 USB3.0 协议, 速度已达到 5.0 Gbps, 而且支持设备的带电热插拔, 因此支持 USB 总线协议的设备使用起来十分方便, 被广泛应用在通用计

算机和边缘计算领域。如文献[2]作者使用 Windows 的 DDK 开发工具, 基于 USB 总线的 HID 人机接口协议, 开发了一种 USB 接口自动测试设备。但对于工业系统开发者和厂商来说, USB 驱动程序开发过程十分复杂, 在 Windows 环境下需要使用 WDK 或者 DDK 开发工具, 并且随着操作系统的升级, 开发工具和驱动程序也需要随之升级, 另外, 商业开发 USB 设备驱动还需要缴纳认证费用, 增加经济负担。

作者简介:王晓东(1976—), 男, 河北衡水人, 高级工程师, 博士, 主要从事光学工程研究。

因此大多数的工业控制器仍然使用串口或者 USB 转串口的方式通讯,正常速度为几十 kbps 到几百 kbps,效率较低。计算机的 USB 接口 HID(Human Interface Device)类协议^[3] 负责人机接口设备如鼠标、键盘和游戏杆等和计算机交互操作。USB HID 接口权限自由度高,特别是鼠标类设备具有操作系统最高权限,其每个操作信号、每组数据都可以在用户许可的情况下发出或者接收,应用开发用户可以调用 HID 接口的 API 函数支持人机接口类别的设备。

STM32 系列硬件是专为要求高性能、低成本、低功耗的嵌入式应用设计的 ARM Cortex 内核处理器,其内部自带 flash 和 RAM,接口资源丰富,并自带 USB 物理收发器,因此使用单芯片即可构建完整的智能物联网仪器仪表的核心模块^[4-5]。STM32 处理器性价比很高,适合于工业和民用测控领域使用,文献[6]即使用 STM32 搭建低成本的测量装置,实现了高精度的电量测量。STM32 处理器的通讯功能强大,在 STM32 处理器内部集成了 USB 收发器,支持 USB HID 通讯协议,使用 USB HID 接口每次最多可发送 1 024 个字节,对于 USB2.0 硬件,最大速度可以达到 24.576 MB/s,足以满足中等性能工业控制器的需求。有学者使用 USB HID 技术开发了专用的鼠标^[7]、键盘^[8]和手势输入设备^[9],但是由于没有主机数据输出,仅有单向数据输入功能,不能用于控制器。

本设计基于 STM32 构建了一个使用 USB HID 协议的多功能控制器,数据传输工作采用中断传输方式,主机和控制器(从机)之间建立相互独立的写入和读出管道,上位机控制软件直接调用 HID 接口的 API 函数实现对下位机硬件的读写控制。

2 系统硬件原理和组成

基于 STM32 的 USB HID 接口控制器的硬件系统结构如图 1 所示。控制器选择 STM32F103 作为系统主处理器,其工作频率为 72 MHz,采用 ARM CortexM3 内核,功耗约为 0.1 W,其内部集成了两个 AD 采样器,可以直接接收 0~3 V 的模拟输入电压,支持 USB

全速设备。主处理器通过定时器中断输出 3.3 V 的 PWM 信号,PWM 信号经过光电隔离后输出给控制驱动电路。本设计使用 LED 驱动电路进行验证,由于驱动电路工作在高电压、大电流状态下,因此 PWM 信号必须隔离输出。在控制器系统集成多路光电隔离的 GPIO 信号,用于外部开关控制,系统中 MAX3232 作为 UART 串口驱动,用于调试接口和功能扩展。

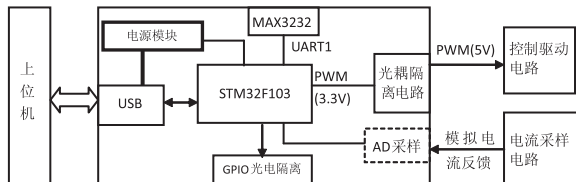


图 1 USB HID 接口控制器的硬件系统结构图

控制器 USB HID 接口部分的实际电路图如图 2 所示。STM32F103 的 PA11 和 PA12 分别连接 USB 总线的 DM 和 DP 端,在 DP 端上拉 1.5 k Ω 电阻用于全速设备检测热插拔事件,如果 DP 端直接上拉到 3.3 V 电压,因从机的上电延迟,会导致枚举失败,因此 USB 总线的上拉电阻接到 STM32F103 处理器的 IO 端口 PB10,当控制器启动,在固件 main 函数中将 PB10 拉高,从而等待 USB 插入事件,开始进行枚举。

3 USB HID 接口多功能控制器设备固件软件设计

参考 USB HID 协议,USB HID 接口控制器和主机之间需建立两种传输管道:控制传输管道和中断传输管道。控制传输管道用于设备枚举,从设备上传设备描述符、配置描述符以及 HID 描述符等信息,下发 Get 请求和 Set 请求,在设备资源方面占用默认的端点 0。因为本控制器属于自定义的 HID 设备,首先需要重新定义 HID 报表描述符,在报表描述中增加和修改数据输入和输出报告,修改通讯数据包的字长、字数目、传输方向、最大值和最小值。HID 报表描述符完成后,在 HID 描述符填充 HID 报表描述符信息(长度、类别)。控制器使用中断传输进行实际的负载数据传输,其内部分配两个中断类型的端点,分别用于数据接收和发送。

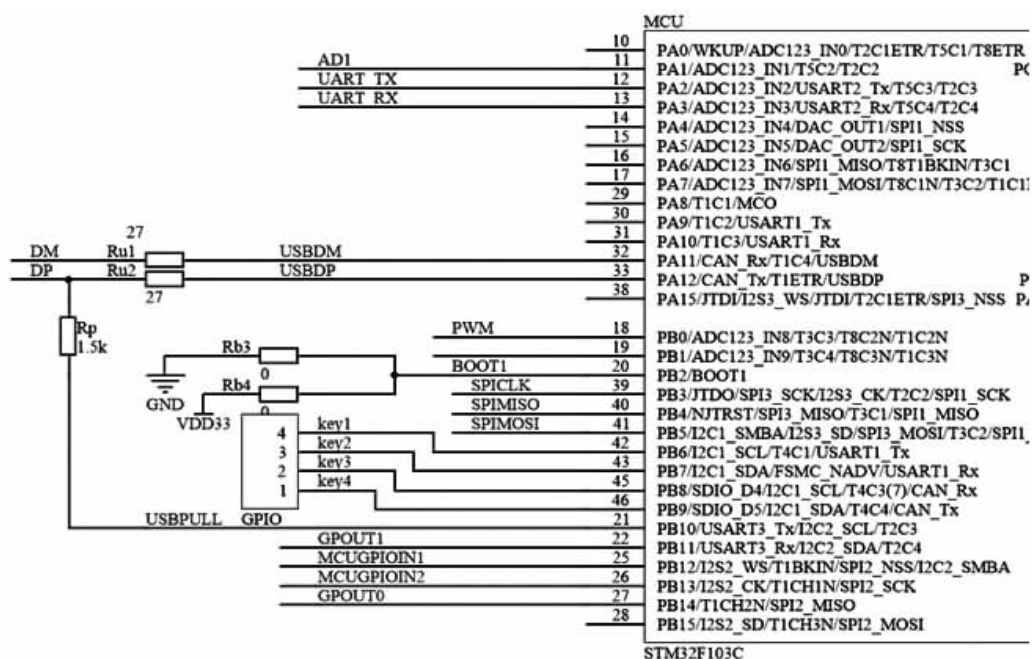


图 2 USB HID 接口电路图

3.1 控制器 USB HID 接口枚举和工作过程

控制器 USB HID 接口主从机枚举工作和交互过程如图 3 所示。主机检测到从机接入后,开始标准设备枚举过程,包括请求和接收设备描述符、设置从机地址、请求和接收配置描述符三个步骤。通过标准设备枚举过程,主机获得控制器的设备描述符、配置描述符、接口描述符和端点描述符。标准设备枚举之后,进行 HID 类设备枚举,主机向从机发送 Get HID 类描述符请求,然后收到 HID 类描述符和报表描述符。在控制器全部枚举后,系统进入正常的工作状态,在 Windows 环境下主机通过调用 ReadFile 和 WriteFile 实现和从机之间收发数据。在 Linux 环境下可以调用 libusb 库实现 USB HID 通讯。HID 设备定义了 6 个 HID 类请求,在 STM32 固件程序的 Setup 阶段中断函数中响应这些请求命令,根据 HID 请求发送相应的 HID 类描述符。

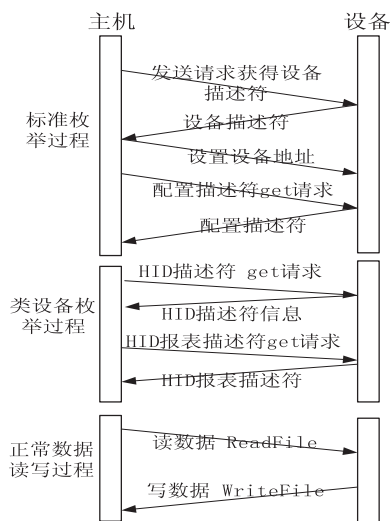


图 3 控制器 USB HID 接口
主从机枚举工作和交互过程

3.2 控制器固件的 HID 类描述符定义及专用的数据结构和对象变量

对于自定义的 USB 总线 HID 设备,必须重新构建 Report 报表描述符^[3]。图 4 是控制

器 HID 协议报表描述符的基本结构,包括输入报告和输出报告,有效数据报文包含 8 个字单元,每个字单元由 8 个二进制位(1 个字节)组成。报表描述符定义好后在 HID 设备的接口描述符和端点描述符进行传输方式定义,接口描述符的 bNumEndpoints, bInterfaceClass 和 bInterfaceProtocol 字段分别定义控制器端点数目、HID 设备类别以及 HID 交互协议。

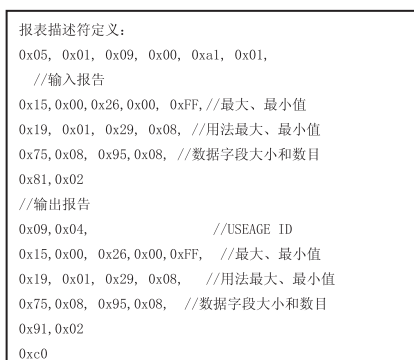


图 4 控制器 USB HID 接口协议
报表描述符的基本结构

控制器 STM32 硬件的固件程序使用三个数据结构处理 USB 通讯事务,分别是:① DEVICE_INFO,保存 USB 请求命令和通讯状态信息;② DEVICE_PROP,USB 通讯过程的回调函数,包括控制传输 Setup 阶段处理,Stage IN 和 OUT 事务处理,获得设备、配置和字符描述符处理;③ USER_STANDARD_REQUESTS,一组标准请求回调函数指针,用于定制上位机的标准请求,可编写相应的函数实例化这些指针,也可以设置成 NOP_Process 系统函数。本控制器系统函数均设置成 NOP_Process。控制器固件程序使用 STM32 开发库定义以上三个数据结构的实例化对象指针变量 pInformation, pProperty, pUser_Standard_Requests 处理 USB 通讯。在 STM32 的 USB 总线的中断处理 USB_LP_CAN1_RX0_IRQHandler 中调用 CTR_LP 中断服务函数,中断服务函数内部这三个指针指向结构体对象完成控制传输过程。

3.3 控制器固件软件主控程序设计

多功能控制器固件主控程序流程如图 5 所示。控制器分别使用中断模式的端点 1(管道 EP1)、端点 2 接收和发送数据。在端点 1 的中断服务中如果接收到一串长度和格式符合要求的数据,即作为一条控制消息,将其放入一个消息队列。在主程序中使用一个 while 循环任务作为后台主线程程序,主线程程序首先检测消息队列是否为空,如果不为空,则从队列中读取一条控制消息,对控制消息进行 CRC 校验,如

果 CRC 校验成功,则将控制消息里命令参数赋值给控制寄存器,然后在后续的过程中根据控制寄存器的值运行 PWM 控制程序、AD 采样程序、GPIO 读写程序等子模块。

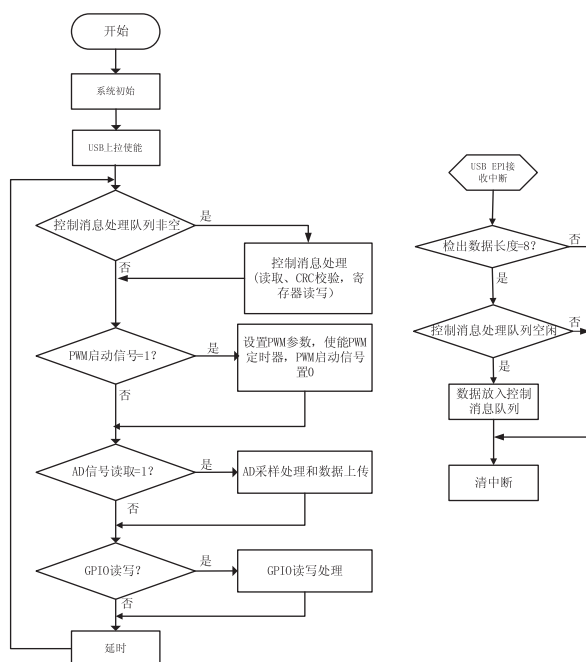


图 5 多功能控制器固件主控程序流程图

4 上位机对 USB HID 控制器读写程序设计

针对本 USB HID 接口多功能控制器,上位机软件使用 API 函数实现 USB HID 功能,在 Windows 环境下可以调用 hid.lib, hidclass.lib, hidparse.lib 和 setupapi.lib 四个接口库完成,而在 Linux 环境下可使用 libusb 调用 libusb-1.0.so.0.1.0 库完成。限于篇幅,以下介绍 Windows 环境下上位机实现 USB HID 功能编程的方法。

在 Windows 环境下编程时程序中需要 hidsdi.h 和 setupapi.h 两个头文件。调用这两个头文件必须按照 C 语言的调用规则。Windows 环境下上位机对 USB HID 接口控制器读写流程如图 6 所示。首先找到 HID 设备的 GUID,调用后获得 HID 设备的指针,然后找到单个 HID 设备的信息,调用后使用 CreateFile 函数按照设备路径打开,设备打开后读出设备属性,然后根据设备属性找到设备属性中符合

控制器 ID(VID 和 PID)的设备,针对满足条件的设备重新使用 CreateFile 函数调用两次,建立读写句柄 hReader 和 hWriter。以上过程是通过调用 hidsdi.h 和 setupapi.h 头文件里面的 HidD_GetHidGuid, SetupDiGetClassDevs, SetupDiEnumDeviceInterfaces, SetupDiEnumDeviceInterfaceDetail 等一系列 API 函数实现的。设备检测过程中,因为 USB 从机设备最多有 127 个,因此设备遍历检测计数设置为 127,即可保证全部 HID 设备属性被检测到。在主机的读写进程中直接调用 ReadFile 和 WriteFile 函数实现对控制器的读写,hReader 和 hWriter 句柄是关键参数。

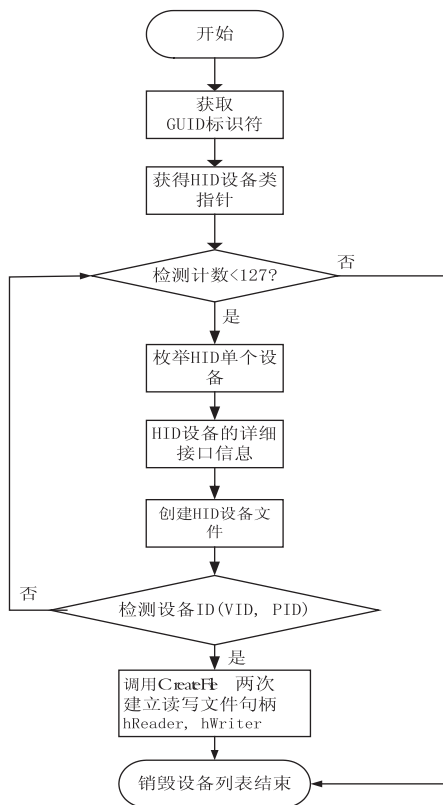


图 6 上位机对 USB HID 接口控制器读写流程图

5 实验验证

在验证实验中,USB HID 接口控制器和主机之间的数据流管道配置如图 7 所示。主机通过 EP0 端点的控制传输方式对控制器进行枚举和初始化;设备的端点 1(管道 EP1)用于接收主机实际控制信息,端点 2(管道 EP2)用于

设备数据读出,EP1,EP2 工作采用中断传输方式。控制器数据包的大小采用 8 个字节,控制器和主机之间采用短数据包的形式保证控制器响应的敏捷性,每个控制输出事务和状态读取的输入事务均包含 8 个字节的有效数据。

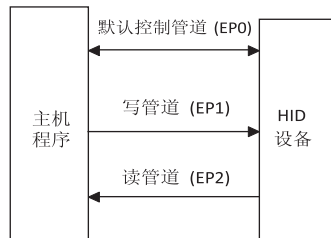
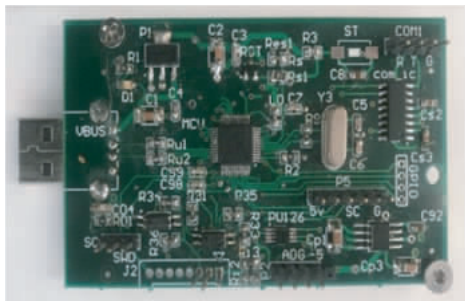


图 7 控制器和主机之间数据流管道配置

实验验证用的处理器型号是 STM32F103C8T6。图 8 是 USB HID 接口控制器实际硬件电路和系统测试实验图。外部测试硬件使用一个 LED 光源,通过 PWM 接口控制 LED 恒流源驱动器输出。



(a) 实际硬件电路

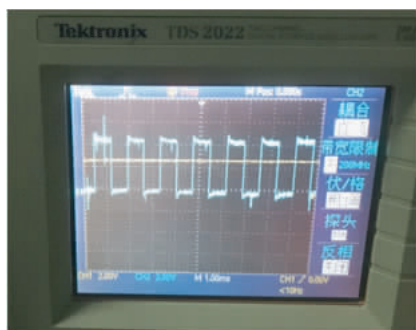


(b) 系统测试实验图

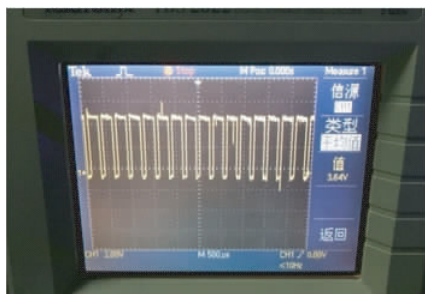
图 8 USB HID 接口控制器实际硬件电路和系统测试实验图

图 9 是示波器显示的 USB HID 接口控制器 PWM 信号输出的实际波形,其中(a)为 50% 占空比,(b)为 80% 占空比,硬件根据控制要求改变 PWM 信号的占空比。图 10 是控制

器通过 PWM 信号控制 LED 光源获得的实验数据,横坐标为 PWM 信号的占空比(ratio),其范围从 0.05 到 1.0(较小数据没有显示),纵坐标为 LED 工作电流值。LED 的驱动电路使用 LM3409 恒流源驱动芯片,在实验过程中上位机向控制器下发占空比(ratio)和频率参数,然后控制器根据这两个参数生成 PWM 信号, PWM 信号控制 LED 光源驱动芯片输出的开关,从而调节 LED 光源的亮度。选择 2 kHz 和 8 kHz 的 PWM 信号频率进行测试,由图 10 可知,当选择较高频率(8 kHz)PWM 信号时,LED 调节的范围较大,较小的占空比也有电流输出,LED 电流输出最小值可以小于 1 mA,可见控制器可以很方便地实现一个宽范围的控制过程。控制器硬件平均工作电流为 135 mA,功耗低、性能稳定,数据测试过程中上位机输出 100% 成功。系统经过 24 h 长时间测试,能够稳定运行。



(a)通道 1(50%的占空比)



(b)通道 2(80%的占空比)

图 9 USB HID 接口控制器 PWM 信号的输出波形(泰克示波器检测)

6 结论和展望

本文提出了一种基于 STM32 硬件的 USB HID 接口多功能控制器设计方案,主机和 USB

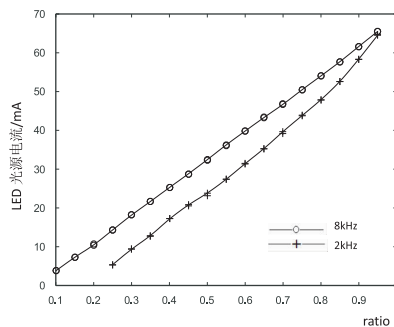


图 10 控制器通过 PWM 信号控制 LED

光源图示(改变占空比获得电流输出)

设备之间采用控制管道、读管道和写管道等相互独立的多个传输管道实现控制数据的读写。通过重新定义 USB HID 报表描述符和设备与主机之间的双向通讯控制流程,实现了可编程变频 PWM 信号控制、GPIO 读入和输出、模拟量 AD 采集多种功能。实验验证的结果表明,系统使用灵活,反应快速,可以稳定运行。

此控制器可以在跨平台操作系统的主机下使用,安装和部署方便,且免驱安装,可用于机器视觉系统、生产测试、工业测控自动化、仪器仪表、边缘计算、可穿戴设备等领域,具备运算处理能力强、功耗小、使用和维护成本低等优点,而且为应对智能传感器大数据控制的需求,此控制器可以通过增加 USB 接口设置、修改包长、升级 USB 协议版本来满足数据流传输的要求,因此具有良好的应用价值。

参考文献:

- [1] COMPAD, HEWLETT_PACKARD, INTEL, et al. Universal serial bus specification revision 2.0[S/OL]. (2000-04-27). <https://www.usb.org/2000-4-27>.
- [2] 匡洋. 球一杆系统自校正控制研究及其测控设备的 USB 驱动程序开发[D]. 长沙:中南大学, 2010.
- [3] USB-IF. Universal Serial Bus (USB) device class definition for Human Interface Devices (HID) Version 1.11[S/OL]. (2001-06-27). <https://www.usb.org/2001-6-27>.

(下转第 67 页)

车方向。机动车通行高峰时段的仿真实验结果显示,与动态连续车道管理方法和静态车道管理方法相比,复合动态车道管理方法能够改善道路交叉路口的整体性能,能够更加均衡地利用干道双向的道路空间,这种性能优势随着交叉路口机动车交通负荷的提高而愈发显著,而且结合现有道路上的直行待行区,可以增加车辆的通行效率,缓解交通拥堵。

参考文献:

[1] 张卓然. BP神经网络和自适应模糊推理系

(上接第26页)

- [4] 薛雅丽. 基于STM32W108的粮食储备系统粮情监测[J]. 唐山学院学报, 2016, 29(3): 35-37.
- [5] 任丽棉. 基于STM32的电力谐波测试仪的设计[J]. 唐山学院学报, 2014, 27(6): 67-68.
- [6] 范大勇. 基于STM32的低成本高精度电能测量装置设计[J]. 传感器与微系统, 2019(10): 85-89.

(上接第41页)

- [8] XU G, CHENG Y, LIU F et al. A water level prediction model based on ARIMA-RNN[C]. 2019 IEEE Fifth International Conference on Big Data Computing Service and Applications (BigDataService), Newark, CA, USA, 2019: 221-226.
- [9] 韩敏, 王亚楠. 基于储备池主成分分析的多

元在多传感器信息融合系统中的研究[D]. 武汉: 武汉工业学院, 2012.

- [2] 韩鹏. 基于短时交通流预测交叉口优化配时研究[D]. 合肥: 中国科学技术大学, 2017.
- [3] 唐凡. 基于神经网络的城市交通流预测研究[J]. 四川水泥, 2019(4): 335.
- [4] 胡枫. 基于马尔科夫模型的短时交通流预测研究[D]. 南京: 南京邮电大学, 2013.
- [5] 杨慧慧. 城市交通流短时预测模型研究[D]. 焦作: 河南理工大学, 2015.

(责任编辑: 夏玉玲)

- [7] 戴捷, 胡晓吉. 基于光传输的USB键盘鼠标一体化设计[J]. 计算机工程与设计, 2012, 33(7): 2620-2627.
- [8] 刘春林. 基于USB协议键盘信息快捷输入技术研究[D]. 哈尔滨: 哈尔滨工业大学, 2017.
- [9] 滕鹏, 姜昌华, 王春慧, 等. 基于HID类的USB手势输入系统设计与实现[J]. 数字技术与应用, 2017(2): 180-181.

(责任编辑: 李秀荣)

元时间序列预测研究[J]. 控制与决策, 2009, 24(10): 1526-1530.

- [10] JAEGER H. The "echo state" approach to analysing and training recurrent neural networks with an Erratum note [R]. Bonn: German National Research Center for Information Technology, 2001.

(责任编辑: 李秀荣)