

嵌入式 $\mu\text{C}/\text{OS-II}$ 系统的 SNMP Agent 研究与实现

王 涛,李宥谋,刘永斌,赵成青

(西安邮电大学 计算机学院,陕西 西安 710000)

摘 要:现如今对于仪器仪表的要求越来越高,不仅要求测量精度高、操作简单,而且对它的实时性要求也越来越高。为了满足这一需求,选择一款合适的 RTOS(嵌入式实时操作系统,embedded real-time operation system)对于代理系统稳定高效的运行具有十分重要的意义。 $\mu\text{C}/\text{OS-II}$ 系统是一款开源的抢占式的多任务实时内核,其结构紧凑、占用空间小、执行效率高,具有可扩展性好、程序代码免费开源、资料丰富、技术成熟、简单易用等特点,适合用于教学科研和一些实时系统的应用。基于嵌入式 $\mu\text{C}/\text{OS-II}$ 系统,按照 SNMP 协议要求,文中对 SNMP 代理进行精简和优化,并在仪器管理系统中采用 C 语言实现 SNMP 仪器代理功能,提高了 SNMP 管理站和 SNMP 代理之间信息交互的实时性。通过对 SNMP 仪器代理功能测试,结果证明在应用中是可行的。

关键词:SNMP 代理; $\mu\text{C}/\text{OS-II}$ 系统;实时性;信息交互;功能测试

中图分类号:TP216

文献标识码:A

文章编号:1673-629X(2018)08-0135-04

doi:10.3969/j.issn.1673-629X.2018.08.028

Research and Implementation of SNMP Agent Based on Embedded $\mu\text{C}/\text{OS-II}$ System

WANG Tao, LI You-mou, LIU Yong-bin, ZHAO Cheng-qing

(School of Computer Science, Xi'an University of Posts and Telecommunications, Xi'an 710000, China)

Abstract: Now the requirement of the instruments is increasingly high. We need not only its high accuracy and simple operation, but also better real-time. Therefore, it is significant to choose a suitable RTOS (embedded real-time operation system) for the operation of the agent system stably and efficiently. The $\mu\text{C}/\text{OS-II}$ system is an open source preemptive multi-task real-time kernel which is usually applied on teaching and research and RTOS because of its compact structure, less space occupied, higher efficiency, with better scalability, free open source program code, abundant data, mature technology and easy use. Based on embedded $\mu\text{C}/\text{OS-II}$ system, according to SNMP, we streamline and optimize SNMP agents, and use C language to realize SNMP agent instrument proxy function in the instrument management system to enhance the real-time of information exchange between SNMP station and SNMP agent. After the test of the function of SNMP instrument, it is proved to be feasible.

Key words: SNMP agent; $\mu\text{C}/\text{OS-II}$ system; real time; information interaction; functional testing

0 引言

随着互联网的飞速发展,网络设备往往由许多相对独立的可管理模块组成,并要求实现灵活扩展,以满足特定的应用^[1]。由于这些可管理模块可能包含其相应的标准或私有的 MIB 库,故对于网络管理,尤其是对代理的可扩展性提出了更高的要求。然而作为网络管理协议业界标准的简单网络管理协议(simple network management protocol, SNMP),却没有提供能将众多模块纳入统一的 SNMP 管理之下的解决方案^[2]。

网络管理标准共有 SNMP(v1/v2c/3)三个版本,应用最多的是 v1 和 v2c 版本。由于 HFC(混合光纤同轴电缆网)接入网还没有定义标准的节点,因此,要用 SNMP 对设备进行管理,必须要扩展原有的 MIB。由于不同设备的操作系统有很大的区别,这就要求代理程序能够方便地进行移植,不应带有专有系统的痕迹^[3]。

为了满足这一需求,选择一款合适的 RTOS(嵌入式实时操作系统,embedded real-time operation sys-

收稿日期:2017-08-08

修回日期:2017-12-27

网络出版时间:2018-03-07

基金项目:陕西省重大科技创新专项资助项目(2010ZKC02-08);嵌入式 LXI 网络仪器开放及产业化(2015KTCQ01-04)

作者简介:王 涛(1991-),男,硕士研究生,研究方向为嵌入式系统开发与设计;李宥谋,教授,研究方向为集成电路设计、嵌入式系统开发与设计。

网络出版地址: <http://cnki.net/kcms/detail/61.1450.TP.20180307.1427.046.html>

tem) 具有十分重要的意义。当下比较流行的嵌入式操作系统有: 美国 WindRiver 公司的 VxWorks、Windows CE、 μ Clinux、 μ C/OS-II, 另外还有 eCos 等^[4]。 μ C/OS-II 是一款优秀的嵌入式实时操作系统, 移植时十分的简单, 只需要对 CPU 中相关的三个文件进行修改即可, 而且占用空间小、执行效率高, 可以满足文中的需求^[5]。

1 SNMP 的主要组成

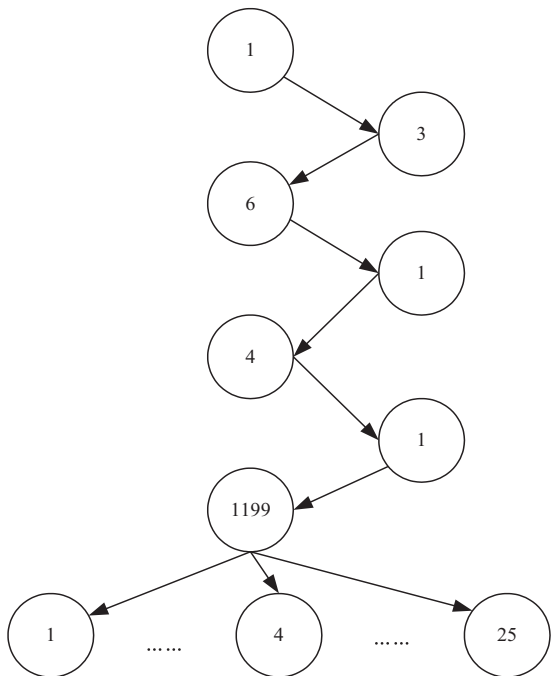
1.1 管理信息结构(SMI)

SMI 规范了被管对象的表示和命名方法, 以及 MIB 中各类信息的数据类型。为了保证管理信息具有唯一识别性, 可对被管对象命名和标识。标识由一个简单的文本描述和一个整数组成, 在消息传递中比名字更容易编码, 更节省空间^[6]。因此, 使用标识定义被管对象。对象标识采用树形结构来管理信息目录。

1.2 管理信息库(MIB)

为了实现被管设备的网络管理, 每一个被管理元素被看作是一个对象, 而 MIB 就是这些对象的结构化集合。MIB 使用 SMI 和 ASN.1 来描述管理信息, 管理信息建立在 SMI 定义的信息树下(iso.org.dod.internet), 每个管理信息都是管理信息树的一个节点, 拥有唯一的对象标识(object identifier, OID)。MIB 库定义了管理信息的 OID、类型、取值、权限, 管理者通过管理协议对 MIB 库的信息进行存取和访问, 就可以实现基本的网络管理, 因此对 MIB 库的访问存取是实现网络管理的关键^[7]。

建立的私有 MIB 如图 1 所示。



万方数据 1 MIB 库的建立

由于树中的每个节点都有唯一的一个双亲节点, 因此, 采用“双亲表示法”存储 MIB 树型结构。双亲表示法和二叉树的双亲链表相类似, 用一组连续的存储空间(一维数组)存储树中的各个节点。数组中的一个元素采用树中的一个节点, 每个节点含两个域, 数据域存放节点本身信息, 双亲域指示本节点的双亲节点在数组中的位置。

这种表示方法有利于更快速地查找父节点和根节点^[8]。

1.3 SNMP 规程

SNMP 以 UDP 协议为基础, 规定了管理者和 Agent 之间 SNMP 报文信息交互的一些语法、语义和交互基本规则。为了方便实现管理者与 Agent 之间信息的交互, SNMP 提供了一套命令(get、set、bulkget、get-next、Trap)来实现对 MIB 信息的存取访问。

2 代理的设计平台

2.1 软件开发工具的选择

针对 ARM 处理器常用的开发工具有 ADS(ARM developer suite)、Keil μ Vision、ARM-Linux-GCC/GDB、Embedded VC 等, 其中 GCC/GDB 主要用于嵌入式 Linux 系统的开发, Embedded VC 主要用于 WinCE 系统下的嵌入式开发^[9]。

由于文中是基于嵌入式实时操作系统 μ C/OS-II, 同时 Keil μ Vision 相比 ADS 在窗口界面上为软件开发者提供了一个整洁、方便、高效的环境, 新版本支持更多最新的 ARM 芯片, 比如文中使用的 LPC1768 微处理器, 所以综合考虑选择 Keil μ Vision4 作为软件开发环境^[10]。

2.2 移植 μ C/OS-II 的步骤

第一步: 在 Keil 中新建一个工程, 并添加启动代码 start.s。

第二步: 通过 <http://micrium.com/> 下载 μ C/OS-II 的工程文件, 将下载的源码包按要求分类加入到新建的工程中。

第三步: 按需要对源代码进行修改, 由于修改的源代码主要与系统内核有关, 而 μ C/OS-II 里 Ports 文件夹的文件 os_cpu.h、os_cpu_a.asm、os_cpu.c 都是与系统内核有关的, 因此只要对这三个文件进行修改, 就能把它移植到嵌入式处理器 LPC1768 上, 如图 2 所示。

3 仪器代理的设计方案

3.1 仪器代理模块硬件设计

代理模块硬件由开发板和功能模块组成, 如图 3 所示。

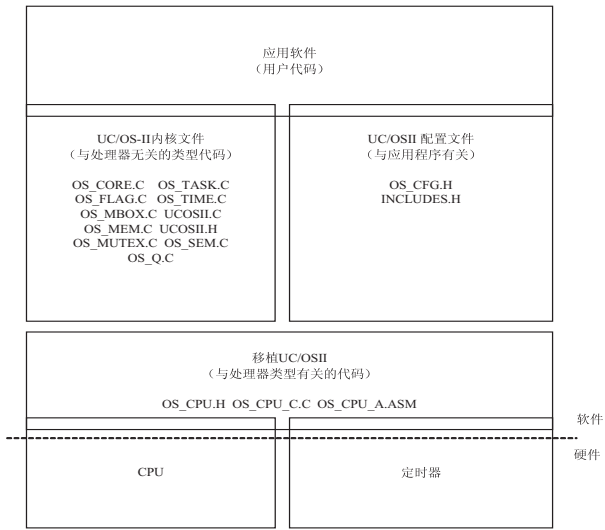


图 2 μ C/OS-II 的文件结构

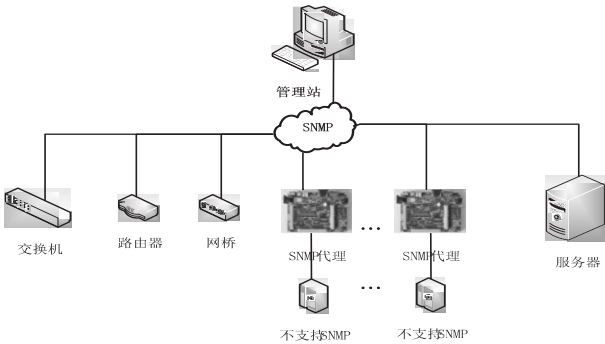


图 3 代理硬件组成

硬件主电路如下:

- (1) 恩智浦 LPC1700, 支持可移植 μ C/os-II 操作系统。
- (2) 有多外设扩展接口供用户应用, 如 GPIO 接口、CAN 总线接口、SPI 接口、以太网接口、USB 接口、可供下载调试的 JTAG 接口等。
- (3) 开发板支持多种调试软件、调试工具进行操作, 如支持 KEIL、IAR 等仿真软件; 支持 J-LINK、AK100 等仿真工具^[11]。
- (4) 仪器模块: 温度传感器模块、二氧化碳模块、示波器等。

3.2 仪器代理模块软件总体设计

代理软件在总体结构上是按照硬件适配层、TCP/IP 协议栈层、应用软件层来划分的, 其中应用层又分为 SNMP 服务器模块、扫描仪器模块、Trap 模块等, 硬件适配层包括串口接口驱动、IIC 驱动等。各层次间的函数调用并不一定都是从上往下逐层展开的, 例如应用层软件有可能直接调用硬件适配层的函数^[12]。软件总体架构如图 4 所示。

4 SNMP 代理模块的测试与分析

为了对开发软件进行测试, 搭建如下的测试系统:

硬件包括一台远端控制计算机、一个路由器和一个 LPC1768 开发板, Windows 操作系统, NMS 管理软件 net-snmp-5.4.1-3.exe^[13]。

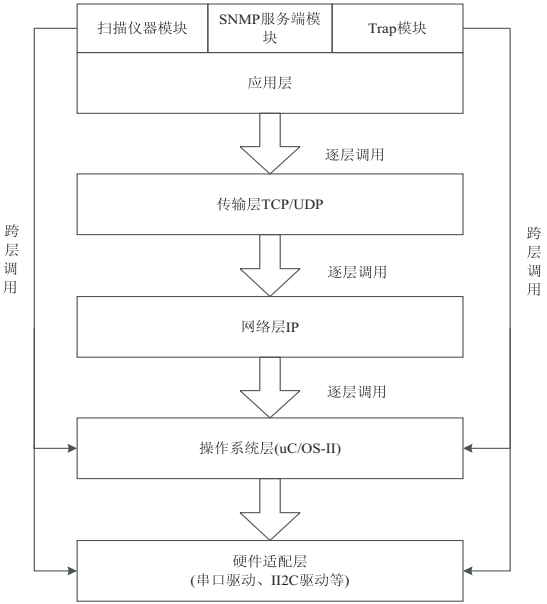


图 4 软件总体架构

在代理 LPC1768 开发板上运行 SNMP 代理, 等待远端管理主机上 SNMP 管理站的消息请求, 测试过程为: 首先通过 get 查看代理的信息, 然后通过对代理上的第四个仪器上的四个节点的访问, 可以查看到仪器的四个属性包括基本的仪器属性, 例如: 仪器名称、仪器型号等。文中代理 1 上第四个仪器连接的是一个高精度的温度传感器, 可以通过 get 命令实时地查看温度的变化情况, 可以实时地把温度传给 SNMP 的管理站, 管理站就可以得到温度的变化情况。这里以代理 1 的第四个仪器为例来进行测试。

Get 命令格式为: snmpget -v 2c -c 团体名 代理的 IP 地址 对象 OID

snmpget -v 2c -c public 192.168.0.201 1.3.6.1.4.1.1199.1.5.0

响应信息为:

SNMPv2-SMI::enterprise.1199.1.5.0=STRING: "Agent1_lpc1768"

Get 命令格式为: snmpget -v 2c -c 团体名 代理的 IP 地址 对象 OID

snmpget -v 2c -c public 192.168.0.201 1.3.6.1.4.1.1199.1.4.1.0

响应信息为:

SNMPv2-SMI::enterprise.1199.1.4.1.0=STRING: "IIC"

Get 命令格式为: snmpget -v 2c -c 团体名 代理的 IP 地址 对象 OID

snmpget -v 2c -c public 192.168.0.201 1.3.6.

1.4.1.1199.1.4.2.0

响应信息为:

SNMPv2-SMI::enterprise.1199.1.4.2.0 =
STRING: "p_wendu"

Get 命令格式为: snmpget -v 2c -c 团体名 代理的 IP 地址 对象 OID

snmpget -v 2c -c public 192.168.0.201 1.3.6.1.4.1.1199.1.4.3.0

1.4.1.1199.1.4.3.0

响应信息为:

SNMPv2-SMI::enterprise.1199.1.4.3.0 =
STRING: "GY-80"

Get 命令格式为: snmpget -v 2c -c 团体名 代理的 IP 地址 对象 OID

snmpget -v 2c -c public 192.168.0.201 1.3.6.1.4.1.1199.1.4.4.0

1.4.1.1199.1.4.4.0

因为 1.3.6.1.4.1.1199.1.4.4.0 对应的 OID 是高精度温度传感器采集温度传上来的十六进制数,是一个实时变化的数据,所以此时得到的数据只能代表此时的温度。举例如下,采集到的温度是: SNMPv2-SMI::enterprise.1199.1.4.4.0 = STRING: 01 02,然后将传上来的十六进制的 01 02,通过传感器资料转换成十进制的温度值: $(0 * 16^3 + 1 * 16^2 + 0 * 16^1 + 2 * 16^0) / 10 = 25.8$,所以此时的空气温度是 25.8 度。

4.1 测试步骤

(1) 采用 keil 编译工具对 SNMP_Agent 代理源代码进行编译、调试、运行;

(2) 采取不同的测试方法,完成对管理对象的操作。

4.2 NMS 管理站测试结果

在测试过程中,SNMP 代理进程运行在 LPC1768 开发板上,其 IP 地址为: 192.168.0.201。该方法使用标准的 net-snmp 软件包,通过命令行方式完成系统测试。有关 net-snmp 安装请参考相关资料,这里不再赘述^[14]。对于系统中所有的管理对象的测试方法类似,在此以管理对象对代理 1 上的第四个仪器测试为例进行说明,其测试结果如图 5 所示。

```
C:\usr\bin>
C:\usr\bin>snmpget -v 2c -c public 192.168.0.201 1.3.6.1.4.1.1199.1.5.0
SNMPv2-SMI::enterprises.1199.1.5.0 = STRING: "Agent1_lpc1768"

C:\usr\bin>snmpget -v 2c -c public 192.168.0.201 1.3.6.1.4.1.1199.1.4.1.0
SNMPv2-SMI::enterprises.1199.1.4.1.0 = STRING: "IIC"

C:\usr\bin>snmpget -v 2c -c public 192.168.0.201 1.3.6.1.4.1.1199.1.4.2.0
SNMPv2-SMI::enterprises.1199.1.4.2.0 = STRING: "p_wendu"

C:\usr\bin>snmpget -v 2c -c public 192.168.0.201 1.3.6.1.4.1.1199.1.4.3.0
SNMPv2-SMI::enterprises.1199.1.4.3.0 = STRING: "GY-80"

C:\usr\bin>snmpget -v 2c -c public 192.168.0.201 1.3.6.1.4.1.1199.1.4.4.0
SNMPv2-SMI::enterprises.1199.1.4.4.0 = Hex-STRING: 01 0C

C:\usr\bin>snmpbulkget -v 2c -c public 192.168.0.201 1.3.6.1.4.1.1199.1.3.4.0
SNMPv2-SMI::enterprises.1199.1.4.1.0 = STRING: "IIC"
SNMPv2-SMI::enterprises.1199.1.4.2.0 = STRING: "p_wendu"
SNMPv2-SMI::enterprises.1199.1.4.3.0 = STRING: "GY-80"
SNMPv2-SMI::enterprises.1199.1.4.4.0 = Hex-STRING: 01 0C
```

万方数据 NMS 测试结果(1)

从图 5 可以看出,SNMP 管理站对代理 1 的管理是成功的。首先查看的是代理的信息,可以查看出是第几个代理和代理的型号;其次通过 get 查看代理 1 上仪器 4 的每一个节点,对每一个节点进行 get 就可以得到不同的属性。可见 SNMP 管理站可以通过与代理之间的交互,直接获得仪器的数据和属性信息。对代理发送 snmpbulkget 命令,通过该命令就可以一次获得仪器上的多个属性,文中为了方便对仪器的操作,对 snmpgetbulk 进行了限制,使其一次获得最多的仪器属性为 4 个,就是在命令行输入 OID 节点,就可以连续获得其节点下面的四个连续节点的属性。

从图 6 可以看出,远程管理主机对代理进行 snmpgetnext、set 管理是成功的,通过 snmpgetnext 命令可以得到下一个节点的值,也就是通过上一个节点可以查看下一个节点的值,而通过远程管理主机向代理首先发送 get 命令查询该管理节点当前的属性,然后通过 set 命令修改该节点的属性,最后再通过 get 命令查询该节点的属性是否修改成功。通过图 6 可以看出, set 成功修改了仪器原来的名称,使仪器拥有了一个新的名称。

```
C:\usr\bin>snmpgetnext -v 2c -c public 192.168.0.201 1.3.6.1.4.1.1199.1.4.1.0
SNMPv2-SMI::enterprises.1199.1.4.2.0 = STRING: "p_wendu"

C:\usr\bin>snmpget -v 2c -c public 192.168.0.201 1.3.6.1.4.1.1199.1.4.2.0
SNMPv2-SMI::enterprises.1199.1.4.2.0 = STRING: "p_wendu"

C:\usr\bin>snmpset -v 2c -c public 192.168.0.201 1.3.6.1.4.1.1199.1.4.2.0 s p_temperature
SNMPv2-SMI::enterprises.1199.1.4.2.0 = STRING: "p_temperature"

C:\usr\bin>snmpget -v 2c -c public 192.168.0.201 1.3.6.1.4.1.1199.1.4.2.0
SNMPv2-SMI::enterprises.1199.1.4.2.0 = STRING: "p_temperature"

C:\usr\bin>
```

图 6 NMS 测试结果(2)

5 结束语

首先对 $\mu C/OS-II$ 系统的优势进行了简单的介绍,对 SNMP 协议的 SMI、MIB 库以及 SNMP 规程进行分析,然后建立私有 MIB 库和 $\mu C/OS-II$ 系统的移植,通过测试环境对所需软硬件的要求,给出设计方案,最后使用标准的 net-snmp 管理软件对代理开发板上的 SNMP 代理进行请求 (get、bulkget、getnext、set),以检测该代理是否可以实现远程管理站与仪器之间的通信。结果表明,测试基本满足了实现对仪器远程控制的实时性要求,可能由于以太网的实际应用环境,会在网络层及传输层产生延迟,会对仪器的实时控制产生影响。

参考文献:

- [1] 葛茂,杨清娜,赵进.网络仪器技术与物联网的发展[J].物联网技术,2011(6):77-79.
- [2] LIU Kang, YANG Lishen, TANG Yongli. The research and implementation of safety isolation system SNMP agent[J].

(下转第 143 页)

三分钟以内推送到对端接收前置服务器中,较短的时间是自动站和国家级站,分别是资料到达后 128 s 内推送到接收前置机的数据库表中。传统的传输方式使得数据获取时效比较慢,原因可能是气象数据源分布在不同的观测系统中,数据分别分散地存储在不同服务器中,获取数据前没有对数据进行有效的整合和预处理困难等等。

推送系统实现后,通过预先整合各类气象数据源到一个统一的平台,对数据进行推送预处理并按需定制策略推送,这样优化了数据推送环节,有效保障了数据的推送时效。同时也可根据业务扩展实现推送资料的种类定制推送,对每一类数据的推送方式和推送时效的触发控制,达到对每一类数据推送指令的配置目的,实现对每一类数据推送策略的配置与管理,提高了传输时效。

5 结束语

从气象数据共享的方式,设计并实现了气象数据的快速推送系统,并在海南省气象局的内网中实现了对省三防办所需的气象数据的实时推送,为海南当地的气象服务提供了便捷高效的气象资料推送服务。该系统上线以来,在强台风、暴雨等重大天气过程中,发挥了积极的作用。下一步,将开发多策略预订等推送服务,不断完善系统功能,提供更高时效性的气象数据推送服务,为海南的经济发展和海南气象服务贡献一份绵薄之力。

参考文献:

[1] 李 强,朱立谷,曾赛峰,等. 分布式目录同步的冲突处理与副本管理研究[J]. 计算机研究与发展,2012,49:257-262.

[2] 李 强,朱立谷,曾赛峰. 弱连接网络环境中的多主机文件同步[J]. 小型微型计算机系统,2012,33(10):2182-2187.

[3] 胡晓勤,卢正添,刘晓洁,等. 远程文件快速同步方法[J].

电子科技大学学报,2008,37(4):594-597.

[4] 徐 旦,生拥宏,鞠大鹏,等. 高效的两轮远程文件快速同步算法[J]. 计算机科学与探索,2011,5(1):38-49.

[5] 李 雪,咸 迪. 跨平台文件共享系统设计与实现[J]. 计算机技术与发展,2012,22(6):191-194.

[6] 敖 莉,舒继武,李明强. 重复数据删除技术[J]. 软件学报,2010,21(5):916-929.

[7] 杨小龙,李 涛,刘晓洁,等. 基于差异的文件同步系统的设计 and 实现[J]. 微计算机信息,2009,25(3-3):67-69.

[8] 曾 珊,周 薇,韩冀中. 面向多应用的文件同步方法[J]. 计算机应用与软件,2016,33(10):297-302.

[9] 何 骞,卓碧华. 一种远程文件同步方法[J]. 计算机应用,2012,32(2):566-568.

[10] 汤晓迪,马晓旭,宁 静,等. 远程文件差异同步系统的设计与实现[J]. 计算机工程与设计,2010,31(20):4389-4392.

[11] 李 贞. 基于 Rsync 算法的远程文件同步系统的设计与实现[D]. 北京:北京邮电大学,2010.

[12] SUEL T, NOEL P, TRENDAFILOV D. Improved file synchronization techniques for maintaining large replicated collections over slow networks[C]//Proceedings of the 20th international conference on data engineering. Boston, MA, USA:IEEE,2004:153-164.

[13] IRMK U, MIHAYLOV S, SUEL T. Improved single-round protocols for remote file synchronization[C]//24th annual joint conference of the IEEE computer and communications societies. Miami, FL, USA:IEEE,2005:1665-1676.

[14] ARMBRUST M, FOX A, PATTERSON D A, et al. SCADS: scale-independent storage for social computing applications[C]//Proceedings of 4th Biennial conference on innovative data systems research. Asilomar, CA, USA:IEEE,2009:86-93.

[15] YANG Fan, SHANMUGASUNDARAM J, YERNENI R. A scalable data platform for a large number of small applications[C]//Proceedings of 4th Biennial conference on innovative data systems research. Asilomar, CA, USA:IEEE,2009:17-27.

(上接第 138 页)

Advanced Materials Research,2014,971-973:1998-2003.

[3] 胡方炜. 基于嵌入式 Linux 系统的 SNMP 代理开发研究[D]. 杭州:浙江工业大学,2004.

[4] 张雪峰. 基于实时内核 $\mu\text{C}/\text{OS-II}$ 的应用研究[D]. 成都:西南交通大学,2004.

[5] LABROSSE J J. 嵌入式实时操作系统 $\mu\text{C}/\text{OS-II}$ [M]. 北京:北京航空航天大学出版社,2003.

[6] XUE Sujing. Discuss the design method of topology discovery system based on SNMP[J]. Advanced Materials Research,2011,267:594-598.

[7] TIAN Jinghua, WANG Jun. Application software based on SNMP in the teaching of networks management[J]. Advanced Materials Research,2011,271-273:1816-1818.

[8] 严蔚敏. 数据结构[M]. 北京:清华大学出版社,1996:305-379.

[9] 马海红. 基于 ARM 和 RTOS 的嵌入式 GUI 应用平台设计[D]. 武汉:武汉理工大学,2005.

[10] ARM 发布 Keil Vision4 集成开发环境为嵌入式软件开发提供简化的开发环境[J]. 电子与电脑,2009(4):70.

[11] 陈承慧. 基于 FS8610 的嵌入式 SNMP 代理的分析与实现[D]. 苏州:苏州大学,2009.

[12] 姜晓琳,王厚军,马 敏. LXI 标准下的网络 VISA 设计与实现[J]. 计算机测量与控制,2010,18(8):1849-1851.

[13] 刘小兵. 基于 LXI 网络化仪器通信模块的设计与实现[D]. 西安:西安邮电大学,2016.

[14] 葛化敏,杨利青,郑 静. 基于 NET-SNMP 的嵌入式网络管理代理的开发[J]. 通信技术,2009,42(11):138-140.