

嵌入式软件算法级功耗BP网络模型研究

刘啸滨¹, 郭兵¹, 沈艳², 王继禾¹, 伍元胜¹

(1. 四川大学计算机学院 成都 610065; 2. 成都信息工程学院控制工程学院 成都 610225)

【摘要】从算法级分析软件功耗和软件特征的关联关系,对嵌入式软件的时间复杂度、空间复杂度和输入规模3个特征进行度量,提出一种基于算法复杂度的嵌入式软件功耗宏模型。设计、训练一种BP神经网络,用于实现功耗函数逼近。仿真实验表明,该功耗函数的估算结果和真实值误差在10%以内,可用于快速估算软件算法在一定输入规模情况下的功耗值,为下一步开展功耗优化工作打下基础。

关键词 算法级; BP神经网络; 复杂度; 嵌入式软件; 功耗模型

中图分类号 TP302.7

文献标识码 A

doi:10.3969/j.issn.1001-0548.2011.06.022

Research on Embedded Software Power Model of Algorithm Level Using BP Neural Networks

LIU Xiao-bin¹, GUO Bing¹, SHEN Yan², WANG Ji-he¹, and WU Yuan-sheng¹

(1. School of Computer Science & Engineering, Sichuan University Chengdu 610065;

2. School of Control Engineering, Chengdu University of Information Technology Chengdu 610225)

Abstract Power consumption in embedded systems is becoming a hot issue that attracts more attention. Different assemble instruction set, software algorithm, and high-level software architecture can significantly affect the system energy consumption. In this paper, we firstly analyze the relations between software energy consumption and some software characteristics on algorithm level. Through measuring three software characteristics, i.e., average time complexity, space complexity, and input scale, we propose a BP neural network software power model based on algorithm complexity. Then, we design and train a kind of BP neural network to accomplish energy consumption function approximation. Simulation experiment results show that the error between the estimation value of this energy consumption function and the real energy consumption value is below 10%. Therefore, it could quickly estimate the energy consumption of software in some input scale, which is an important fundament to explore the energy consumption optimization in the future.

Key words algorithm level; BP neural network; complexity; embedded software; power model

目前在“低碳经济”的背景下,嵌入式系统的功耗是一个日益引起人们关注的热点问题,业已受到业内软/硬件开发商和政府的高度重视^[1]。

由于嵌入式系统是典型的软件驱动执行的系统,硬件的电路活动“直接”导致系统功耗的产生,软件的指令执行和数据存取等操作驱动底层微处理器、总线、Cache、存储器 and I/O接口等硬件的电路活动,“间接”导致了系统功耗的产生。因此,嵌入式软件是产生系统功耗的“主动”因素和“活跃”因素,这也是嵌入式软件功耗的本质含义。先前的许多研究表明,不同的汇编指令及软件的高层体系结构都将对电路动态功耗翻转率产生影响,从而进一步影响系统功耗。随着嵌入式系统功能的不断增

强,嵌入式软件的规模将不断增长,嵌入式软件频繁运行产生的功耗占了系统功耗的大部分比例,平均为80%左右。借鉴硬件低功耗设计的思路和方法,嵌入式软件的低功耗设计是研究在一定性能(如速度、吞吐量、实时性、可靠性、防危性、可信性、成本等)约束下嵌入式软件的功耗优化(功耗降低或功耗最小化)问题,属于静态功耗优化技术,包括功耗仿真、功耗分析与建模、功耗估计、功耗降低、功耗评价以及相应的支持工具开发等内容,是软件运行时开展动态电源管理(dynamic power management, DPM)和动态电压调节(dynamic voltage scale, DVS)的基础。

软件功耗分析与建模研究具有不同的层面和粒

收稿日期: 2010-03-27; 修回日期: 2010-06-10

基金项目: 国家863计划(2008AA01Z105); 四川省杰出青年科技基金(2010JQ0011)

作者简介: 刘啸滨(1987-),男,博士生,主要从事嵌入式系统、中间件技术、软件体系结构等方面的研究。

度,包括源程序结构级、算法级和软件体系结构级。本文针对单核计算机系统,在算法级对嵌入式软件功耗进行分析与建模,提出一种基于算法复杂度的嵌入式软件功耗宏模型,模型中采用BP神经网络实现功耗函数逼近,以构造准确、实用的功耗模型。

1 软件功耗模型

针对嵌入式软件的功耗分析与建模技术,文献[2]提出了对操作系统采用服务例程级功耗建模的思想,并具体分析了Linux操作系统50种服务例程的功耗-IPC关联模型;文献[3]采用二元线性回归方法对UC/OS II和Embedded Linux两种RTOS的功耗实验数据进行归纳和分析,得出了RTOS服务例程级的宏模型,即功耗与软件的算法复杂度、通信量和路径基本块关联信息等高层度量特征之间的函数关系;文献[4]采用SimBed指令级功耗仿真器对UC/OS、Echidna和NOS的RTOS的功耗和性能数据进行实验、度量、分析与比较,揭示了RTOS的动态运行行为,对RTOS的结构改进提出了一些有价值的建议。由于多数现代微处理器中都嵌入了可编程的事件计数器,文献[5-6]提出通过监控系统发生的事件,测量微处理器的性能,进一步用于软件的功耗估计与优化;文献[7]提出通过对微处理器的每条汇编指令以及指令对(instruction pair)进行功耗测量,实现汇编语言程序级的功耗估计,并通过编译时产生的符号表信息部分实现对高级语言程序的功耗估计与优化;文献[8]通过模拟运行嵌入式操作系统和应用软件,利用微体系结构能耗模型估算单时钟周期指令能耗,提出基于软件功能结构的操作系统内核能耗估算模型;文献[9]使用建模语言CSP建立体系结构级功耗模型,提出了进程的迹的概念,并根据系统一次成功运行的迹引出了系统最大能耗和最小能耗的概念。

嵌入式软件能耗的一般模型可表示为:

$$E = \int_{t_1}^{t_2} P(t) dt = \bar{P} \Delta t = f(C_i) T \quad (1)$$

$$i = 1, 2, \dots, n$$

式中, E 表示软件的能耗(energy consumption); $P(t)$ 表示软件的瞬态功率; \bar{P} 表示软件的平均功率; T 表示软件的运行时间; C_i 表示软件功耗的某些相关度量特征,如算法复杂度或微处理器的周期指令数IPC等; $\bar{P} = f(C_i)$ 称为功耗-特征关联模型(简称功耗模型),表示 \bar{P} 是 C_i 的函数。如何确定功耗函数 f ,采取何种措施影响 C_i ,是目前嵌入式软件功耗分析与优化的主要研究内容。

本文模型在算法层次上考虑 \bar{P} 的软件关联特征,并根据软件功耗产生的定义进一步得到:

$$\bar{P} = f(C_i) = f(O_s, S_s, I_s) \quad (2)$$

式中, Q_s 为算法的时间复杂度度量; S_s 为算法的空间复杂度度量; I_s 为算法输入规模的度量。从目前算法研究成果可知,一个算法的平均时间复杂度和空间复杂度是判定算法消耗计算机资源的直接依据,而算法的具体执行性能也会受到输入规模的影响,在算法层次上,3个特征量将会间接地对计算机功耗产生影响。因此,本文抽取3个特征量作为软件功耗的关联特征量。

为确定功耗函数 f ,建立更精确的算法级功耗模型,主要过程如下:

1) 准确地分析和度量算法级的软件特征,获得相应度量值。

2) 通过功耗仿真实验平台HMSim测得 E 和 T 的值,根据式(1),得出:

$$\bar{P} = E/T \quad (3)$$

从而可获得软件的平均功率 \bar{P} 值。

3) 从理论上讲,以前的一些研究成果假设软件功耗与软件高层度量特征之间的关联关系是线性回归函数关系,缺乏充分的依据;从实践上看,通过许多实验发现,功耗是由时间复杂度、空间复杂度和输入规模3个算法层特征量相互作用的结果,一般不是线性相关关系。线性函数拟合功耗函数关系,虽然模型简单,估计速度较快,但功耗估计值误差也较大,许多情况下不能满足实际工程需要。两方面综合分析,嵌入式软件功耗关联函数更可能是一个非线性函数。由于数学上BP神经网络能够逼近任意非线性函数,具有较高的拟合度,在许多领域中得到了广泛的应用,并取得了令人满意的效果。因此,在获得了函数 f 的输入/输出值后,本文使用BP神经网络拟合该关联函数 f 。

2 3个算法特征量的度量

算法复杂性不但反映出一个具体算法的复杂程度,并可直接将该复杂性表现在算法的软件功耗上。算法复杂性分析的研究范围是分析具体算法所需的执行时间、空间开销和它们随某些参数增长的规律^[10]。

2.1 时间复杂度 O_s 度量

时间复杂度是衡量算法优劣的一个重要特征,其对软件功耗也存在较大影响。因此,本文选择其作为功耗度量的第一个特征量。目前,对于算法复

杂度度量的研究, 仍是一个需要进一步解决的问题, 很难采用一个精确的、通用的计算方法进行度量, 一般针对具体算法, 采用算法中基本操作的最大频度量级进行计算。

目前, 关于算法时间复杂度的一些具体度量方法主要包括:

1) 对于一些常用的算法, 有比较公认的平均时间复杂度, 如训练网络使用的冒泡排序BubbleSort() 和选择排序SelectionSort() 算法, 其平均时间复杂度为 $O(n^2)$ 。

2) 对于一些较简单的算法, 可将函数中基本语句使用频度最大的量级作为其时间复杂度, 如使用两层for循环进行运算的算法, 其时间复杂度一般为 $O(n^2)$ 。

3) 某些算法的时间复杂度可能取决于其特定输入的细 节, 如在 n 个元素的列表中查找一个给定项(如查找键 K), 一般可顺序查找列表中的每一个元素, 直到发现了匹配查找键 K 的元素或者到达了列表的终点。在最坏的情况下, 表中没有匹配元素或者第一个匹配元素恰巧是列表的尾元素, 该算法的键值比较次数最多为 n ; 对于顺序查找, 最理想的查找应该是第一个元素等于查找键的列表, 该算法的键值比较次数为1^[11]。

然而, 无论是 最坏情况还是最理想情况, 都不能确切地表达算法试图提供的信息。为此本文选择算法的平均时间复杂度作为软件功耗的一个高层度量特征。

2.2 空间复杂度 S_s 度量

时间复杂度和空间复杂度是算法复杂度的两个重要特征。因此, 算法空间复杂度也是衡量软件功耗的一个重要特征。

目前, 关于算法空间复杂度的度量主要包括3个方面。

1) 问题本身所用空间: 所解决问题的规模, 主要取决于问题本身。由于与算法无关, 因此本文不对其进行讨论。

2) 程序本身所占空间: 程序代码所占的空间对算法空间复杂度影响很小, 不同算法之间的区别很大。

3) 动态变量所占空间: 程序运行时, 动态变量所申请的空间与算法有关, 不同算法之间的区别很大。

程序代码空间由编译器决定, 一个编译器可提供各种优化选项, 包括代码优化、执行时间优化等, 相同的一段高级语言代码, 不同编译器或相同编译

器采用不同编译、优化选项后, 二进制代码所占空间大小一般不相同。并且, 由于程序代码所占空间对算法空间复杂度影响很小, 本文也不对程序代码空间进行讨论。

因此, 本文着重度量算法辅助变量所占空间, 汇总计算算法在其运行期间所申请的空间总量。对于一个结构变量, 将其每个成员所占用的空间累积起来, 即可得到该变量所需要的内存。类似地, 可以得到一个数组变量所需要的空间, 用数组的大小乘以单个数组元素所需要的空间, 如一个int A[rows][cols]的数组, 如果使用Microsoft Visual C++编译器, int数据类型占用字节数为4, 则数组A占用了 $4 \times \text{rows} \times \text{cols}$ 字节, 对于算法中辅助变量未使用列表的情况, 其空间度量为1个int单位。

2.3 输入规模IS度量

算法的运行情况还受到具体的输入规模影响, 因此, 算法的输入规模也是软件功耗的一个重要度量特征。

输入规模的度量方法一般受算法的操作细节影响, 不同的具体算法有不同的度量方法, 如:

1) 对于一个英文字符拼写检查算法, 如果算法对输入的每一个独立字符都要进行相应检查, 可使用字符的数量作为输入规模的度量; 如果算法的操作是以单词为单位, 可统计输入单词的数量作为输入规模的度量。

2) 对于排序、查找列表的最小元素算法等与列表有关的算法, 输入规模通常是列表的长度, 如使用冒泡排序算法对一个整数数组A[MAX_SIZE]进行排序, 排序产生的功耗将根据MAX_SIZE大小而有所不同。因此, MAX_SIZE是该算法的输入规模。

3) 对于一个对 n 次多项式 $p(x)=a_nx_n+\dots+a_0$ 求值的问题, 输入规模是多项式的次数或多项式系数的个数。虽然多项式系数的个数比次数大1, 但如此细小的差别对于功耗分析来说一般是无关紧要的。

4) 对于与数字特性相关的算法, 在度量其输入规模时, 应特别注意检查给定的整数 n 是否为质数。对于该算法, 一般度量数字 n 的二进制表示中的比特数作为算法的输入规模。

3 基于BP神经网络的功耗函数逼近方法

3.1 BP神经网络的结构

BP网络也称为反向传播网络(back-propagation network), 是将W-H学习规则一般化, 对非线性可微分函数进行权值训练的多层网络。BP网络可用于函

数逼近,使用输入矢量和相应的输出矢量训练一个网络逼近一个函数。本文采用具有 n 个隐层节点的单隐层3层BP网络,其结构如图1所示。该结构主要包括:

1) 隐层数的确定

Robert Hecht-Nielson证明了对任何在闭区间内的连续函数,都可用单隐层的BP网络逼近,因而一个3层的BP网络可完成任意的 n 维到 m 维的映射。

2) 节点数的确定

在本文的BP网络中,输入层为OS、SS和IS,输出层为平均功率PS,因此,输入层节点数有3个,输出节点为1个。训练一个单隐层的3层BP网络,一般根据如下经验公式选择隐层节点数:

$$n_1 = \sqrt{n + m} + a \quad (4)$$

式中, $n=3$ 为输入节点个数; $m=1$ 为输出节点个数; a 为1~10之间的常数。因此, n_1 取值范围为3~12。

3) 各层函数的确定

在单隐层BP网络训练中,隐层传递函数一般选用tansig函数,输出层传递函数选用purelin函数。

确定了网络层数、每层节点数、传递函数等内容,即确定了一个具体的、完整的BP网络,然后可开始进行网络训练。

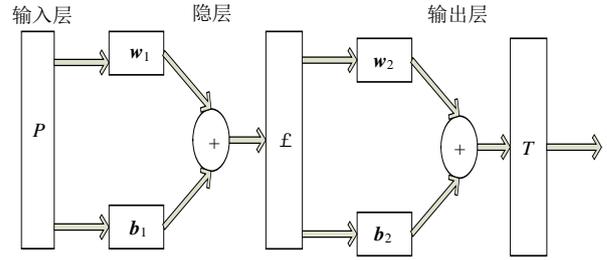


图1 BP神经网络结构图

3.2 BP神经网络的训练

通过功耗仿真实验平台HMSim获得实验数据,用于训练、构造BP神经网络,调整各层的权值和阈值,使网络的均方差MSE达到预期的目标,从而逼近功耗函数 f ,使该神经网络能够在一定输入规模下产生期望的相应输出,即该函数的 \bar{P} 。

借助MATLAB 6.5神经网络工具箱,对网络进行训练和仿真。模型选用的训练样本函数如表1所示。

表1 训练样本函数列表

函数名称	功能描述	时间复杂度	输入规模	空间复杂度
gen_arr	生成一个随机数组	$O(n)$	MAX_SIZE	MAX_SIZE
nn_time1	两层循环算术运算	$O(n^2)$	MAX_SIZE	1
bubbleSort	冒泡排序算法	$O(n^2)$	MAX_SIZE	MAX_SIZE
selectionSort	选择排序算法	$O(n^2)$	MAX_SIZE	MAX_SIZE
nn_time2	两层循环算术运算	$O(n^2)$	MAX_SIZE	1
gen_arr2	生产一个随机的二维数组	$O(n^2)$	NUMBER	2*NUMBER*NUMBER
nn_time	两层循环算术运算	$O(n^2)$	MAX_SIZE	1
mult	两个矩阵相乘	$O(n^3)$	NUMBER	3*NUMBER*NUMBER
arr_insert	在有序数组中插入整数	$O(n)$	MAX_SIZE	MAX_SIZE
checksum	计算整数数组校验和	$O(n^2)$	MAX_SIZE	MAX_SIZE
prime	计算求出素数	$O(n^2)$	MAX_SIZE	MAX_SIZE

以表1中样本函数在功耗仿真实验平台HMSim不同输入规模下测得的功耗数据作为样本集,对BP

网络反复训练,网络训练误差MSE达到 10^{-7} ,得到的BP神经网络权值和阈值如下:

$$w_1 = \begin{bmatrix} 0.6157 & 0.5905 & -8.0500 \\ 10.6114 & 39.1033 & -10.2720 \\ -0.5483 & 16.6181 & 14.5978 \\ -0.9468 & 16.0240 & 7.5257 \\ -5.4141 & 5.7042 & 1.3921 \\ -22.3434 & 21.6025 & 7.8046 \\ 4.4207 & 6.4670 & 0.7924 \\ -28.2331 & 5.8474 & 1.9735 \\ 4.6436 & -0.2500 & 0.7884 \\ -22.2469 & 5.9132 & 2.9127 \\ 0.2610 & -4.6990 & -15.1784 \end{bmatrix}, b_1 = \begin{bmatrix} 2.7250 \\ -11.2109 \\ 7.1159 \\ 11.6679 \\ 2.7347 \\ -5.6363 \\ 11.4869 \\ -23.5519 \\ 4.9896 \\ -19.2797 \\ 6.1130 \end{bmatrix}, w_2 = \begin{bmatrix} 0.8107 \\ 0.0242 \\ 39.5886 \\ -1.3185 \\ -0.0200 \\ -42.7424 \\ 60.2471 \\ 13.9229 \\ -60.2610 \\ -7.4243 \\ 3.7802 \end{bmatrix}, b_2 = [-1.5508]$$

3.3 仿真实验结果及分析

用训练好的网络对测试样本函数进行仿真, 得到网络输出值, 即该函数的 \bar{P} , 然后由实际运行时

间 T , 根据式(1), 可计算出软件功耗 E , 并与实际测试的功耗值比较。测试样本函数如表2所示。

表2 测试样本函数列表

函数名称	函数功能描述	时间复杂度	输入规模	空间复杂度
Max	求二维整数数组中的最大值	$O(n^2)$	NUMBER	NUMBER*NUMBER
Invector	矩阵的逆矩阵	$O(n^2)$	NUMBER	NUMBER*NUMBER
GetBig	得到数组中的最大值	$O(n)$	NUMBER	NUMBER
Sum_diagnoal	矩阵对角线元素之和	$O(n)$	NUMBER	NUMBER

各测试函数 E 预测值和真实值比较如图2~图5所示。由图2~图5可以看到, BP网络实现了很好的预测效果, 预测值和真实值的误差在10%以内, 达到了预定目标, 和其他软件功耗线性宏模型的20%估算误差相比更理想^[3]。本文对于嵌入式功耗的分析和优化研究的意义在于:

1) 确定了用于算法层次软件特征和软件功耗转换的BP网络结构, 根据测试样例得出了网络的权值和阈值, 从而可以定量地估算具体算法的实际软件功耗值, 不再主观性地模糊描述。

2) 验证了算法的时间复杂度、空间复杂度和输入规模是在算法级上分析软件功耗的关键指标。

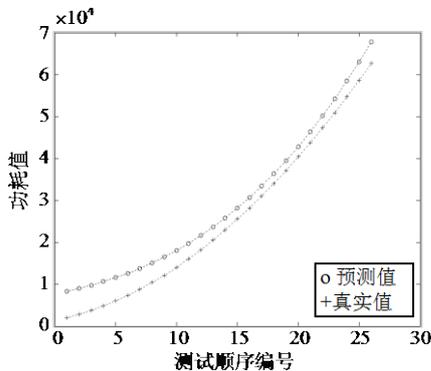


图2 Max函数预测值与真实值比较

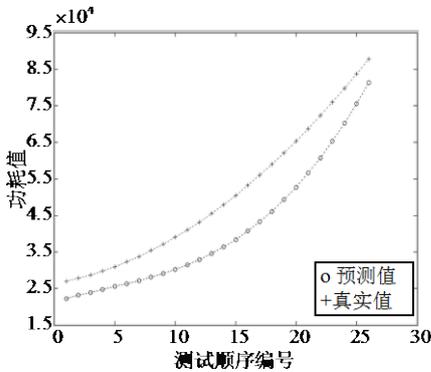


图3 Invector函数功耗预测值与真实值比较

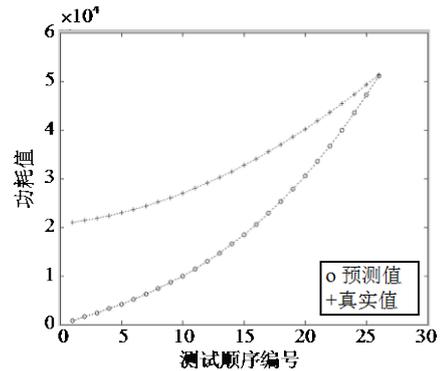


图4 GetBig函数功耗预测值与真实值比较

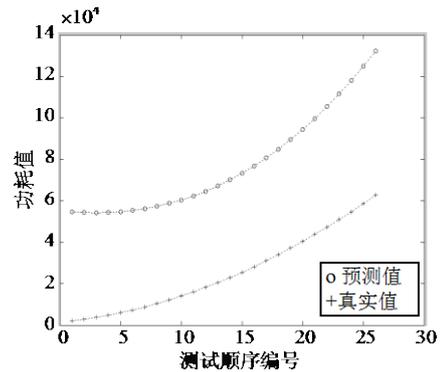


图5 Sum_diagnoal函数功耗预测值与真实值比较

BP神经网络模型能有效地拟合软件功耗和软件高层度量特征之间的非线性关系, 并能为估测算法的软件功耗建立软件功耗的宏模型提供理论支持。

样本的类型和数量对BP网络的训练和仿真影响很大。因此, 本文实验对样本函数和测试函数的类型及数量的选取可根据具体应用需要进行调整。

HMSim是一种高精度的指令级嵌入式软件功耗仿真实验平台, 可模拟单核处理器的ARM7指令集, 支持Embedded Linux和UC/OS II操作系统应用程序的运行。功耗数据由功耗仿真实验平台HMSim测得, 但尚未在其他嵌入式系统和多核芯片架构下进行仿真实验工作, 有待进一步探索和研究。

4 总 结

本文在算法级上对嵌入式软件进行特征量的提取, 使用BP神经网络逼近功耗估算函数, 通过MATLAB实验仿真对各算法的实现函数 E 进行比较准确的估算, 估算误差在10%以内, 验证所提出建模方法的可行性, 对嵌入式功耗研究的后序工作嵌入式软件功耗的优化有很好的指导作用。然而嵌入式软件的实际功耗还可能与其软件体系结构、指令CPI等有关系。因此还将对其进行深入详细的研究, 融合其相关的特征量, 使功耗模型能更加精确地体现函数特征量与其功耗之间的关系。

参 考 文 献

- [1] 郭兵, 沈艳, 邵子立. 绿色计算的重定义与若干探讨[J]. 计算机学报, 2009, 32(12): 2311-2319.
GUO Bing, SHEN Yan, SHAO Zi-li. The redefinition and some discussion of green computing[J]. Chinese Journal of Computers, 2009, 32(12): 2311-2319.
- [2] LI Tao, JOHN L K. Run-time modeling and estimation of operating system power consumption[C]//Proceeding of SIGMETRICS'03. San Diego: [s.n.], 2003.
- [3] TAN T K, RAGHUNATHAN A, JHA N K. Energy macromodeling of embedded operating systems[J]. ACM Transactions on Embedded Computing Systems, 2005, 4(1): 231-254.
- [4] BAYNES K, COLLINS C, FITERMAN E. The performance and energy consumption of embedded real-time operating systems[J]. IEEE Transactions on Computers, 2003, 52(11): 1454-1469.
- [5] JOSEPH R, MARTONOSI M. Run-time power estimation in high performance microprocessors[C]//Proceeding of the International Symposium on Low Power Electronic Device. Huntington Beach: [s.n.], 2001.
- [6] SHERWOOD T, PERELMAN E, HAMERLY G, et al. Automatically characterizing large scale program behavior [C]//Proceedings of the International Conference on Architectural Support for Programming Languages and Operating Systems. California: San Jose, 2002.
- [7] LI Y T S, MALIK S, WOLFE A. Performance estimation of embedded software with instruction cache modeling[J]. ACM Transactions on Design Automation Electronic Systems, 1999, 4(3): 257-279.
- [8] 赵霞, 郭耀, 雷志勇, 等. 基于模拟器的嵌入式操作系统能耗估算与分析[J]. 电子学报, 2008, 36(2): 209-215.
ZHAO Xia, GUO Yao, LEI Zhi-yong, et al. Estimation and analysis of embedded operating system energy consumption [J]. Acta Electronica Sinica, 2008, 36(2): 209-215.
- [9] 张滕滕, 吴晓. 基于CSP的构件化嵌入式软件能耗分析与评估方法研究[J]. 计算机学报, 2009, 32(9): 1876-1883.
ZHANG Teng-teng, WU Xiao. On energy-consumption analysis and evaluation for component-based embedded system with CSP[J]. Chinese Journal of Computers, 2009, 32(9): 1876-1883.
- [10] 王振宇. 程序复杂性度量[J]. 计算机与数字工程, 1998, 26(5): 1-5.
WANG Zhen-Yu. Program complexity metrics[J]. Computer and Digital Engineering, 1998, 26(5):1-5.
- [11] LEVITIN A. 算法设计与分析基础[M]. 2版. 潘彦, 译. 北京: 清华大学出版社, 2007.
LEVITIN A. Introduction to the design and analysis of algorithms[M]. 2nd ed. Translated by PAN Yan. Beijing: Tsinghua University Press, 2007.
- [12] 贾惠芹, 刘君华. 基于BP神经网络的分布式传感器网络的可靠性分析[J]. 铁道学报, 2002, 24(5): 80-83.
JIA Hui-qin, LIU Jun-hua. Reliability analysis of distributed sensor network based on back propagation neural network[J]. Journal of the China Railway Society, 2002, 24(5): 80-83.

编辑 税红