

# 프로세스 상태, 배터리 특성, 응용프로그램 종류를 고려한

## 동적 전력관리 시스템

김광중<sup>○</sup> 박창현  
 영남대학교 컴퓨터공학과  
 kim7463@ynu.ac.kr park@yu.ac.kr

### Dynamic Power Management System Considering Process Status, Battery Characteristics and Application Program Type

Kwang-jung Kim<sup>○</sup> Chang-hyeon Park  
 YeungNam Univ. Department of Computer Engineering

#### 요 약

최근들어 휴대용 미디어 플레이어(Portable Media Player)와, 노트북 컴퓨터, PDA(Personal Data Assistant)의 사용이 늘어나면서 얼마나 오랜 시간동안 휴대용 장치를 사용하는가에 대한 문제가 큰 이슈로 떠오르고 있다. 그에 대해 많은 연구가 이루어져 있으며, 현재도 활발히 연구가 진행되고 있다. 본 논문에서는 프로세스 상태, 배터리 상태 그리고 응용프로그램 종류로 이루어진 상황을 고려한 동적 전력관리 시스템에 대해서 제안하고 그에 따른 세부적인 모듈에 관한 설명과 실험 결과를 보여준다. 실험결과 제안한 시스템을 적용하지 않았을 경우와 비교하여 약 14%의 전력 손실 감소 효과를 볼 수 있었다.

#### 1. 서 론

최근 휴대폰, 스마트폰 등의 이동통신 단말기 외에도 PDA 등 모바일 정보 기기의 사용이 급격히 증가하여 노트북 컴퓨터와 공존하며 많은 사용 현황을 보이고 있다 또한 사용자의 이용 요구 증가에 따른 휴대용 게임기 PMP 등의 사용이 점점 늘어나고 있는 추세이다 이렇듯 다양한 휴대용 정보 기기의 사용이 보편화됨에 따라 배터리 기반의 임베디드 시스템에 대한 저전력 요구가 크게 증가되고 있다. 기존의 전력을 고려한 설계 기법은 주로 회로 수준에서 이루어져 왔기 때문에 하드웨어적으로 구현되어 왔으나 마이크로프로세서의 성능이 향상되고 네트워크 대역폭이 커짐에 따라 휴대용 정보 기기상에서의 동화상, 음악, 게임 등의 다양한 기능 및 고성능 요구로 인해 소프트웨어의 전력 소모가 커지고 있다 따라서, 효율적인 전력 소모 관리를 위해서는 하드웨어의 전력 소모 내용을 기반으로 한 시스템 또는 소프트웨어 수준에서의 전력 관리 기법이 요구된다 이에 대해 본 논문에서는 세 가지 환경을 고려한 하드웨어 수준의 전력 관리 기법에 대해 제안한다

첫 번째는 프로세스 상태이다 어떠한 프로세스의 CPU 사용율에 따라 프로세스 동작에 영향을 주지 않는 범위 내에서 CPU에 인가되는 전압을 조절하여 절전하는 방법이다 일반적으로 회로의 전력(P) 소모는 공급 전압 VDD에 대해  $P \propto VDD^2$  인 관계를 가지므로, 공급 전압의 감소는 에너지 소모량 측면에 있어서 매우 효과적이다 [1][11]

두 번째는 배터리 상태에 따른 전력관리이다 배터리 는 화학적 특징 때문에 충전에서 방전되기 까지 다양한

요인들에 의해서 소비 전력량이 바뀐다 그 중에서 소비 비율유지 효과와 회복 충전효과를 바탕으로 배터리 특성 구간을 설정해 각 특성 구간별 적절한DPM 정책을 적용하는 방법이다.[12] 세 번째는 응용프로그램의 종류에 따른 DPM 정책이다. 응용 프로그램에 따라 사용하는 장치가 다른데, 컴파일 시점에서 그 응용프로그램이 사용하는 장치를 미리 지정함으로써, 실행 시점에서 해당 응용 프로그램이 사용하지 않는 장치에 전원을 차단해 저전력을 구현하는 방법이다

본 논문의 2장에서는 기존의 저전력 관련 연구에 대한 소개와 문제점을 지적하고 3장에서 시스템 구현 부분 4장에서 실험 및 결과를 보여준 뒤에 마지막에서 결론을 내린다.

#### 2. 관련 연구

모바일 장치의 사용이 증가하면서 전력 소비와 절전에 대한 연구가 활발하게 진행되고 있다 전력 관리 기법에 대한 연구는 크게 두 가지로 분류 할 수 있는데 하드웨어 레벨에서 수행하는 방법과 소프트웨어 레벨에서 수행하는 방법으로 나뉜다.

##### 2.1 하드웨어 레벨

하드웨어 레벨의 전력 관리는 직접적인 전력 소모를 줄인다는 장점이 있기 때문에 가장 많이 연구 구현되고 있는 부분이다. 이 방법은 사용되지 않는 외부장치의 전원을 차단한다거나, CPU 의 속도를 늦추고, 모바일 장치 전체의 상태를 idle 로 진입시켜 절전하는 방법이다 대표적인 예로 디바이스 특성을 고려하는 저전력 스케줄링 공유자원을

고려한 저전력 스케줄링이 있다 이 방법은 어떤 응용프로그램이 어떠한 자원을 사용하는지에 대한 정보와 디바이스에 대한 특성을 파악하기가 어렵다는 단점이 있다 그리고 디바이스 제어는 WCET(Worst Case Execution Time)와 AET(Actual Execution Time)를 바탕으로 하는 것이기 때문에 예측 자체가 어렵고 불분명하다는 문제점이 존재한다. [4]

### 2.2 소프트웨어 레벨

소프트웨어에서 이뤄지는 연구는 대부분 스케줄링 알고리즘과 슬랙 분석, 컴파일 시점에서 응용프로그램의 BottleNeck 구간을 분석하여 효율적인 전력 관리 정책을 세우는데 요점을 두고 있다[2][3] 이들 소프트웨어 레벨에서 수행된 연구는 구현하기 어렵다는 단점이 존재한다 이 외에도 멀티미디어 데이터를 해독하는 디코더를 개선하여 CPU의 부하를 줄이는 방법이나 FrameBuffer의 갱신을 비율을 QoS 감소없이 떨어뜨리는 방법이 연구되고 있다.[6]

본 논문에서는 하드웨어 레벨에서의 전력관리 방법을 제안한다.

## 3. 제안 시스템

제안하는 시스템은 크게 다섯 가지 큰 모듈로 구성되어 있다.

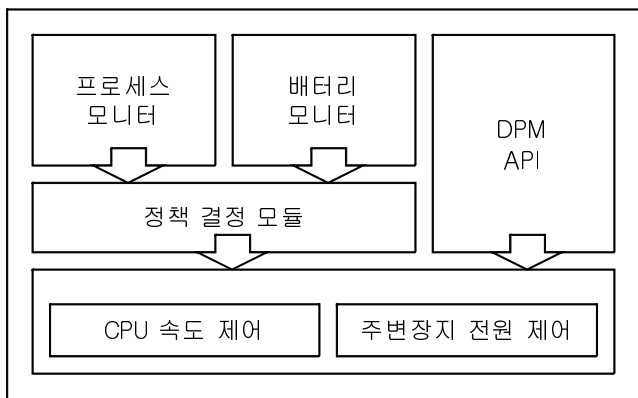


그림 1. 전력관리 시스템의 전체적 구성

정책 결정기는 배터리 모니터, 프로세스 모니터, DPM(Dynamic Power Manager) API 라이브러리를 통해 전달 받은 각종 인자들을 이용해 CPU의 속도, 그리고 주변장치 전원의 상태를 결정한다 그 다음 장치제어기를 통해 실제적인 하드웨어 조절에 들어가게 된다

### 3.1 프로세스 상태를 고려한 DPM

가장 많이 사용되고 있는 임베디드 프로세서인 PXA255/PXA27x 시리즈는 각각 450Mhz, 624Mhz 라는 CPU 클럭 속도를 가지고 있다[5] 이러한 속도를 만들어 내기 위해서는 많은양의 전력소모가 필요하며 휴대용 기기의 런타임동안 이러한 속도가 필요한 구간은 그렇게 많지

않다. 이에 따라 낮은 CPU에서 동작하더라도 QoS가 떨어지지 않는 프로세스가 동작할 때는 CPU의 클럭을 떨어뜨리고, CPU에 많은 부하가 걸리는 프로세스는 CPU의 클럭을 높여서 QoS를 보장하는 방법으로 시스템을 구성한다. 이 시스템은 경험적 데이터로 기능을 수행하며 수행 과정은 다음과 같다.

- ① 아무런 경험적 데이터가 없는 상태에서 CPU 속도를 최고로 높여서 프로세스들을 감시한다
- ② CPU 속도를 단계적으로 낮추면서 프로세스들의 CPU 점유율을 감시한다
- ③ 특정 프로세스의 CPU 점유율이 80~90%가 될 때까지 위 과정을 수행하며 80~90% 되었을 경우 특정 프로세스를 현재 CPU 속도와 함께, 데이터베이스에 저장한다.
- ④ 저장된 프로세스가 수행 중이지 않을 때 위 과정을 반복하여 다른 프로세스의 데이터도 저장한다

위의 과정을 계속 진행하게 되면 QoS를 보장할 수 있는 프로세스와 그 프로세스가 필요한 CPU의 속도를 저장하고 있는 데이터베이스가 만들어지게 된다 이 시스템은 이렇게 저장된 데이터베이스를 바탕으로 CPU Frequency Scaling을 수행하게 된다.

### 3.2 배터리 특성을 고려한 DPM

배터리는 화학적 특징 때문에 충전에서 방전되기까지 다양한 요인들에 의해서 소비 전력량이 바뀐다 그 중에서 가장 크게 영향을 미칠 수 있는 특성은 전력 소비비율 유지 효과와 회복 충전효과이다 전력소비 비율 유지 효과는 배터리의 방전량에 따라 배터리 사용시간이 변하는 것을 의미하며, 회복 충전 효과는 배터리가 방전되는 동안에 유휴시간을 갖게 되면 소량의 전력이 재충전 되는 특징을 말한다 배터리의 수명동안 전류상태를 측정하면 다음과 같은 결과를 볼 수 있다.[8]

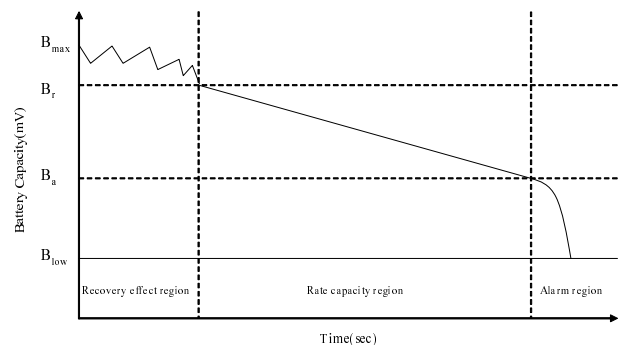


그림 2. 배터리 생명시간동안의 전압 변화

배터리 방전상태는 크게 세 구간으로 나눌 수 있는데 회복충전구간, 소비 비율 유지 구간 경고 구간이다 최근의 모바일 장치의 배터리는 스마트 배터리로 배터리의 잔량이나, 출력 전압등의 측정이 가능하다 그 정보를 바탕으로 구간을 설정한 후, 각 구간에 적합한 DPM 알고리즘을 적용한다.

첫 번째, 회복충전 구간에서는 유휴시간이 갖아 질수록

충전회복 효과가 증가하기 때문에 빠른 시간안에 응용 프로그램을 수행하고 유휴시간을 오래 갖는 것이 중요하다 그래서 회복 충전 구간에서는 프로세서의 속도를 최고로 올려서 응용프로그램을 빠르게 수행한다

두 번째, 소비 비율 유지 구간에서는 기존의 알고리즘으로 전력 소비를 최소화 시킨다 [11][12]

마지막 경고 구간에서는 응용프로그램의 속도나 QoS 를 떨어뜨리더라도 응용프로그램 수행시간을 늘리는 방향으로 알고리즘을 적용한다 이 구간에서는 프로세서의 속도를 응용프로그램을 수행할 수 있는 최저의 속도로 떨어뜨려 디바이스 수명을 늘린다[3]

3.3 사용하는 장치를 고려한 DPM

모바일 장치가 동작 하는 동안에 많은 외부 장치들이 함께 동작한다. 모든 응용프로그램이 이 외부 장치들을 사용하는 것이 아니므로, 꼭 필요한 장치만 동작시키는 것은 전력소비 감소에 매우 효과적인 방법이라고 할 수 있다 이 장에서는 제안된 API 를 이용하여 프로그램이 사용하는 외부장치를 정의하고 그것을 이용하여 외부 장치를 제어하는 방법에 대해서 기술한다

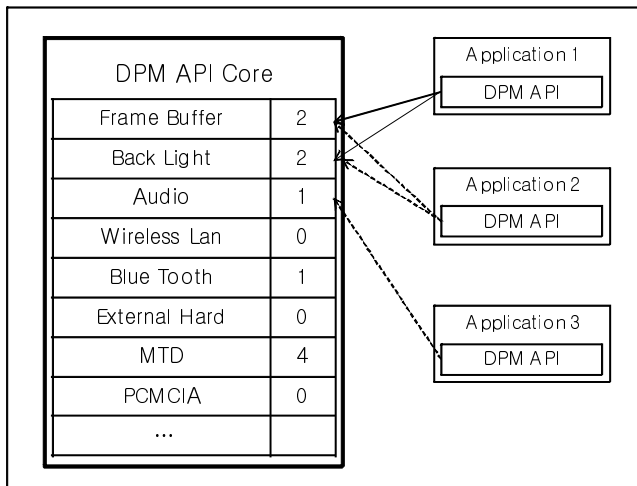


그림 3. API Core 구성도

API 코어부분에는 특정 장치와 그 장치를 사용하고 있는 응용 프로그램의 수를 저장하고 있는 배열이 존재한다 응용 프로그램 실행 시점에서 특정 장치와 연결되어 있는 카운터를 1 증가시키고, 응용 프로그램이 종료될 때 카운터를 1 감소시킨다. API 코어는 배열을 참조하여 장치의 on/off 를 결정하게 된다.

3.4 정책 결정 모듈

정책 결정 모듈에서는 프로세스 모니터 배터리 모니터가 가지고 있는 정보를 바탕으로 적용해야할 전력 관리 정책을 결정하는 역할을 수행한다 정책이 결정되면 이 모듈은 장치 제어가 장치를 제어 할 수 있도록 신호를 보내며 장치 제어기는 Linux Kernel 에 포함되어 있는 PM(Power Management) 을 이용하여 주변 장치를 제어한다.

4. 실험 및 결과

본 실험은 PXA255/PXA270 실습보드에 서 이루어졌으며, 배터리 상태에 따른 실험은 Zaurus, IPAQ에서 수행되었다.

4.1 단위시간 동안의 전력 소비량

이 실험에서는 프로세스 상태에 따른 DPM 과 응용프로그램 종류에 따른 DPM 에 대한 성능 측정을 수행한다 30분 동안 MPEG4 동영상과 MP3 파일을 재생시켰다 그리고 중간에 약 1분씩 유휴 상태를 두고, 무선랜을 일정 주기로 on/off 시키면서 전력 소비를 측정하였다 그림은 DPM 을 적용시켰을 때와 그러지 않았을 경우를 비교한 그림이다.

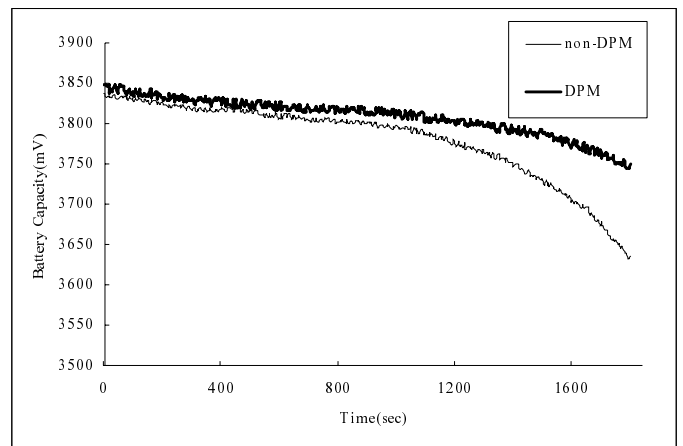


그림 4. 단위시간 동안 전력 소비량

실험결과 DPM 을 적용했을 경우에 그러지 않았을 경우와 대비하여 약 20% 의 전력 소비 감소 효과를 보였다 여기서 대부분의 효과는 장치제어에서 가져오는 효과가 대부분 일 것으로 판단되어, DPM API 가 정책결정에 영향을 미치지 않도록 설정한 후 실험을 재수행 하였다

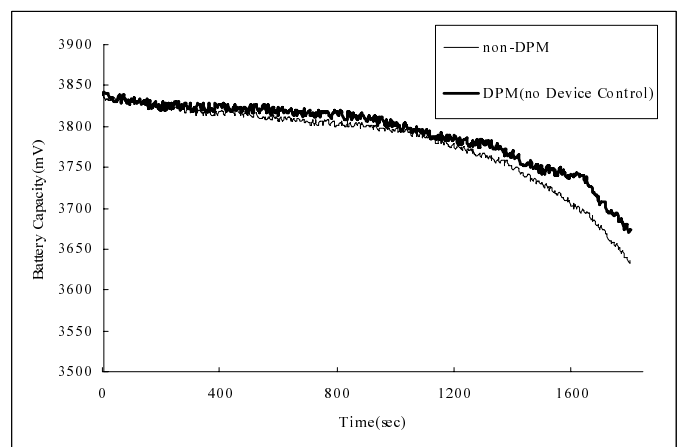


그림 5. 단위시간동안 전력 소비량(장치 제어 제거)

실험결과 DPM 이 적용되지 않았을 경우와 대비하여 약 6% 가량의 전력 소비 감소효과를 볼 수 있었다

4.2 배터리 수명

이 실험은 두 번째 제안했던 배터리 상태에 따른 DPM의 성능을 측정하기 위해 수행되었다 수행 방법은 배터리 수명이 끝날 때까지 MP3와 MPEG4 동영상을 재생시키고 배터리가 방전되는 시간을 측정하였다

실험 결과 DPM 을 적용하지 않았을 경우와 대비 하여 약 11% 의 배터리 수명의 증가 효과를 가져왔다

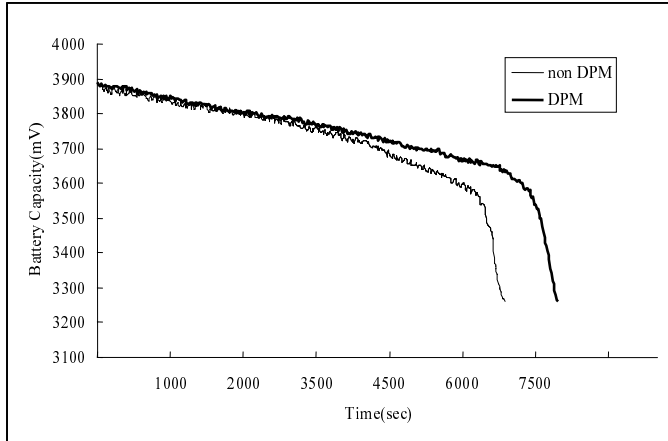


그림 6. 배터리 수명 비교

4.3 전체적 성능

DPM 의 종합적인 성능 평가를 하기 위하여 배터리 수명 시간 동안 여러 가지 응용프로그램을 실행시키고 배터리가 방전될 때까지의 시간을 측정하였다

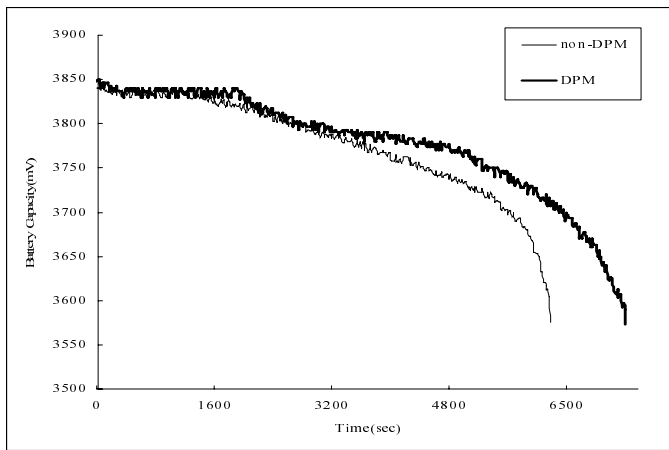


그림 7. 전체적 성능 비교

실험 결과 약 14.3% 의 배터리 수명 증가 효과를 볼 수 있었다. 이 실험 결과는 어떤 프로그램을 실행시켰느냐에 따라 달라 질 수 있는 수치이지만 DPM 을 적용했을 때와 그렇지 않았을 경우 전력소비 감소의 효과를 볼 수 있었다는 점에 의미를 둘 수 있다

5. 정리 및 결론

본 논문에서는 세 가지의 상태에 따라서 소프트웨어 수준에서의 전력 관리 방법에 대한 방법을 제시 하였다 첫 번째는 응용 프로그램이 모니터링을 통하여 응용 프로그램

이 필요로 하는 최소한의 CPU 속도를 할당해 전력소비를 감소시키는 방법이었고 두 번째는 배터리 상태를 여러 구간으로 나누어 구간에 따라 다른 정책을 주어 저전력을 구현하는 방법이었다 마지막으로, 응용프로그램에 사용되지 않는 외부 장치의 전원을 차단하여 불필요한 전력손실을 막는 방법이었다. 실험 결과 각 방법에 따라 6%, 11%, 20% 의 효과를 볼 수 있었고 세 가지 방법을 동시에 적용했을 경우 14.3% 의 효과를 확인 할 수 있었다 배터리를 고려한 DPM 은 배터리 수명을 늘릴 수 있다는 효과를 가지고 있지만, QoS 가 떨어진다는 점에서 RTOS 등, 상호작용이 중요하게 적용되는 OS나 응용프로그램에 적용하기엔 많은 문제점을 안고 있다 그리고 응용프로그램 종류에 따른 DPM 에서는 사용되는 응용프로그램을 API 를 적용하여 재 컴파일 해야 한다는 치명적인 단점을 가지고 있다. 차후에는 QoS 를 보장하면서 배터리 수명을 늘릴 수 있는 방법과, 시스템 I/O 감시를 통한 외부 장치 사용 여부를 판단하는 방법에 대한 연구가 필요할 것이다 더불어 전력 관리 정책을 사용자가 직접 지정할 수 있는 기법과 사용자의 패턴을 수집해서 예상 정책을 적용시키는 기법에 대한 연구가 필요하다

6. 참고 문헌

1. T.Sakurai and A.R. Newton, "Alpha-Power Law MOSFET Model and Its Application to CMOS Inverter Delay and Other Formulas." IEEE journal of Solid-State Circuits, vol SC-25, No. 2
2. I. Ripoll, A. Crespo and A. G Fornes, "An Optimal Algorithm for Scheduling Soft Aperiodic Task in Dynamic-Priority Preemptive Systems." IEEE Transactions on Software Engineering, vol 23
3. J. P. Lehoczky and S. Ramos-Thuel, "An Optimal Algorithm for Scheduling Soft Aperiodic Task Fixed-Priority Preemptive Systems." in Proceedings of the IEEE Real-Time Systems Symposium
4. H. Yang, R. H, "Low Power Scheduling Based On Device Characteristics"
5. <http://www.intel.com/design/intelxscale>
6. I. Choi, H. Shim, and N. Chang, "Low-power color TFT LCD display for hand-held embedded systems", Proc. ACM ISLPED
7. J. Lorch, Operating systems techniques for reducing processor energy consumption, Ph.D. thesis, Computer Science, University of California at Berkeley, December 2001.
8. S. Choi, "Application and Battery Aware DVS System", Yonsei University M.S thesis
9. J. Luo, J. K. Niraj, "Battery-aware static scheduling for distributed real-time embedded systems Design," Proceedings of ACM/IEEE conference on Design automation, Las vegas, Nevada, United States, pp. 444 - 449, December

- 2001.
10. T. L. Martin, Balancing Batteries, Power and Performance: System Issues in CPU Speed-Setting for Mobile Computing, PhD thesis, Department of Electrical and Computer
  11. D. Rakhmatov, S. Vrudhula, "Energy management for battery-powered embedded systems," ACM Transactions on Embedded Computing Systems(TECS), Vol. 2, Issue 3, pp. 277 - 324, August 2003.
  12. M. Doyle, T. Fuller, J. Newman, "Modeling of Galvanostatic Charge and Discharge of a Lithium/Polymer/Insertion Cell," Journal of the Electrochemical Society, Vol. 140, No. 6, pp. 1526-1533, June 1993
  13. L. Benini, G. Castelli, A. Macii, E. Macii, M. Poncino, R. Scarsi, "Extending lifetime of portable systems by battery scheduling," Proceedings of the conference on Design, automation and test, Munich, Germany, pp. 197-20, March 2001.
  14. Y. Shin, K. Choi, "Power conscious fixed priority scheduling for hard real-time systems," Proceedings of ACM/IEEE conference on Design automation, New Orleans, Louisiana, United States, pp. 134-139, November 1999.
  15. <http://www.handhelds.org/geeklog/index.php>