

관점지향 프로그래밍을 적용한 실행시간 누설전력 관리 모델 설계[†]

(Design of a management model for runtime leakage power using
Aspect-Oriented Programming)

김영민^{*} 이찬근[§]
(Youngmin Kim) (Changun Lee)

요약 최근 임베디드 시스템이 발전함에 따라서 저 전력에 대한 요구가 중요한 관심사가 되었다. 하지만 전력 관리 코드가 핵심 관심사 코드와 횡단 결합되어 있기 때문에 가독성과 유지보수성을 저하시키는 원인이 된다. 본 연구에서는 실행시간 동안 발생하는 누설전력을 감소시키기 위한 디바이스 사용에 집중한 전력관리 모델을 제시하고, 이러한 전력관리 모델에 관점지향 프로그래밍을 적용하여 전력관리 코드를 핵심관심사 코드와 분리한다. 마지막으로 제시한 전력관리 모델을 이론적인 방법과 실험적인 방법으로 평가한다.

키워드 누설 전력, 태스크 분할, 관점 지향 프로그래밍

Abstract The importance of the low-power management has increased due to the recent advances of the embedded systems. However, it is noted that low-power concerns are detrimental to the readability and the maintainability of the codes for the core concerns. In this study, in order to reduce occurring leakage power during run-time, we present a power management while considering the run-time leakage power of devices. the power management codes is separated from the core concern codes by applying aspect-oriented programming. Finally, we analyze the theoretical model of our proposed scheme and present experimental results.

Key words leakage power, task division, aspect-oriented programming

1. 서론

임베디드 시스템의 성능이 향상되어 감에 따라 전력소비의 폭도 그만큼 늘어나고 있다. 설계자라면 반드시 성능의 향상을 꾀하면서도 배터리의 수명을 고려해야 한다. 이렇듯 저 전력에 대한

요구는 이미 시스템 설계의 중요한 가치 기준이 되었다. 한 예로써, 모바일 장치는 항상 전원이 켜져 있는 상태로 유지되어 진다. 이러한 경우에 많은 곳에서 누설전류가 발생하게 된다. 이 누설 전력은 소모되는 전력에 많은 영향을 미친다. 따라서 배터리 수명 연장에 있어 누설 전력을 낮추는 것은 주요한 관심사라 할 수 있다. 하지만 이러한 관심사를 핵심으로 하여 시스템을 구현하게 되면 실제 임베디드 시스템의 핵심 관심사 코드와 전력관리 코드가 횡단 결합(cross-cutting)되어 코드의 가독성과 유지보수성이 떨어지게 된다. 임베디드 환경과는 달리 사용자 컴퓨팅 환경에서는 오래 전부터

[†] 본 연구는 한국연구재단 기초연구사업(과제번호 20110013924)의 지원을 받았습니다.

^{*} 학생회원 : 중앙대학교 컴퓨터공학과
remnant1120@gmail.com

[§] 종신회원 : 중앙대학교 컴퓨터공학과 교수, 교신저자
cglee@cau.ac.kr

논문접수 : 2011년 02월 08일

심사완료 : 2011년 03월 11일

프로그램을 유지보수를 위한 여러 가지 연구 [Jan96] [Na00] [Mey87] [Sel05]가 제안되어 왔다.

기존의 누설전력 관리에 대한 연구에서는 태스크의 실행시간을 제외한 나머지 시간에 디바이스에서 소모하는 전력을 감소시키는 것을 목표로 한다[Sim01][Zha09]. 그리고 전력 관리 코드를 핵심관심사와 분리하는 것이 미흡한 실정이다. 본 연구에서는 임베디드 시스템의 실행시간 누설 전력 관리 모델을 제시함과 동시에 관점 지향 프로그래밍을 적용하여 핵심 관심사 코드와 전력 관리 코드를 분리하였다.

앞으로 2장에서는 횡단 결합된 전력관리 코드를 분리시키는데 사용되는 관점 지향 프로그래밍에 대하여 설명하고, 추가적으로 임베디드 시스템의 주요 가치기준으로 등장한 저 전력 설계를 위한 기법을 언급한다. 3장에서는 본 연구에서 제안하는 관점 지향 프로그래밍을 적용한 실행시간 누설 전력 관리 모델을 제시한다. 4장과 5장에서는 제안하는 전력 관리 모델을 이론적, 실험적으로 평가한다. 마지막으로 본 연구의 결론을 맺고, 향후 연구 방향에 대해 논의하고자 한다.

2. 관련 연구

2.1 관점 지향 프로그래밍

사용자의 요구가 다양해지고 기술의 빠른 변화가 있는 요즘은 프로그램의 개발에서 유지보수에 대한 비용이 많은 부분을 차지하고 있다. 이러한 이유로 프로그램의 가독성과 유지보수성을 위한 연구가 활발하게 진행되고 있다. 가독성과 유지보수성을 위해 제한된 것이 관점 지향 프로그래밍 [Lad03]이다. 관점 지향 프로그래밍은 프로그래머들이 횡단문제를 모듈화 할 때, 사용하는 프로그래밍 기술이다. 객체 지향 프로그래밍(Object-

Oriented Programming)과 같이 모듈화가 뛰어난 기술도 있지만 보안(security), 메모리 관리(memory management), 로깅(logging)과 같은 부가 기능들은 다른 클래스에 긴밀하게 결합되어 있거나 여러 클래스나 오퍼레이션에 횡단하여 결합되어 있어 모듈화가 어렵다는 데에 그 한계가 있기 때문이다[Dou01]. 이러한 관점 지향 프로그래밍의 Aspect 코드와 핵심 관심사 코드의 생성부터 실행 코드의 생성까지의 모든 일련의 과정을 그림 1에서 보여 준다.

관점 지향 프로그래밍에 관련된 연구로는 관점 지향 프로그래밍 지원 도구(Aspect J, Aspect C)를 가지고 프로그램에서 모니터링에 관련된 부분은 따로 기술하는 연구 [Gre05] [Sun09] [Yoo10]가 있다. 추가적으로 최적 전력 흐름 시스템 개발을 위해서 관점지향 프로그래밍을 적용한 연구[Hon07]가 있다. 이는 시스템에 객체 지향 프로그래밍을 사용하기 어려운 경우에 관점 지향 프로그래밍을 사용하여 해결 한다.

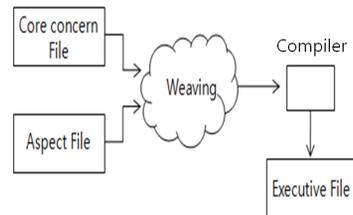


그림 1. 실행 코드의 생성 과정

2.2 임베디드 시스템에서의 전력 관리

휴대용 임베디드 시스템에서 배터리가 한정되어 있기 때문에 전력 소비를 줄이기 위한 연구가 이루어지고 있다. 기반이 되는 연구로는 동적 전력 관리 기법과 정적 관리 기법이 존재한다. 동적 전력 관리 기법은 동적 전력 관리(Dynamic Power Management)기법 관한 연구 [Dev08] [Swa05], 동적 전압 조절(Dynamic Voltage Scaling)기법 관한 연구 [Ayd06] [Jej04], 그리고 동적 전력 관리와

동적 전압 조절을 혼합하여 사용하는 기법에 관한 연구[Sim01][Zha09]가 있다. 동적 전력 관리는 결정 시간 이상으로 디바이스가 가동되지 않는 상태에 머무를 경우 디바이스의 전압 상태를 변화시키는 방법이며, 동적 전압 조절은 전압을 동적으로 조절하여 에너지 소모량을 줄이는 방법이다. 이에 반해 수동 전력 관리 기법은 사용자의 요청에 따라 저 전력 모드로 동작하게 하는 것이다. 정적 누설 관리(Static Leakage Management) [Dro02]를 사용하여 대기 또는 전원 차단(Device-off) 모드로 전환한다.

3. 관점지향 프로그래밍을 적용한 실행시간 누설전력 관리 모델

3.1 디바이스 모델

디바이스는 두 가지 상태를 지원한다. 실행을 위한 전력이 공급되고 있는 active 상태와 시스템 전력만 공급되는 sleep 상태이다. 각 상태마다 소비되는 전력은 차이를 보이게 된다. 본 연구에서는 각 디바이스를 다음과 같이 정의한다.

정의 1: 디바이스 $D_i = (P_a^i, P_s^i, E_{sd}^i, E_{wu}^i, T_{sd}^i, T_{wu}^i)$ 의 각 매개변수는 다음과 같다.

- P_a^i : active 상태에서의 소비전력.
- P_s^i : sleep 상태에서의 소비전력.
- E_{sd}^i : 디바이스의 sleep 상태로 sleep down할 때 발생하는 에너지 소모량.
- E_{wu}^i : 디바이스의 active 상태로 wake up할 때 발생하는 에너지 소모량.
- T_{sd}^i 와 T_{wu}^i : 디바이스의 상태 전환 시 소모되는 시간.

break-even time은 디바이스의 상태 변경 시 발생하는 에너지 오버헤드를 보상받기 위해서 디바이스가 sleep 상태로 유지해야하는 최소 시간이다.

정리 1: 디바이스 i의 break-even time(BET_i)은 다음과 같다[Zha09].

$$BET_i = \frac{E_{sd}^i + E_{wu}^i - T_{sw}^i \cdot P_s^i}{P_a^i - P_s^i}$$

디바이스의 sleep down과 wake up시 소모되는 시간의 합은 $T_{sw}^i = T_{sd}^i + T_{wu}^i$ 이다.

3.2 디바이스 사용에 집중한 전력 관리 모델

본 연구에서는 하나의 프로세서를 가진 임베디드 시스템으로 가정한다. 시스템의 요구 사항에 따라서 핵심 관심 사항을 나눈 후 태스크로 할당한다. 그리고 핵심 관심사를 디바이스의 사용에 따라 여러 개의 Job으로 분할한다. Job을 분할 시 Job의 최소 실행시간이 태스크에서 사용하는 디바이스의 가장 긴 BET보다 길어야 한다. 또한 임베디드 시스템의 특성을 감안하여 전력 소모를 줄이기 위해서 실행 중에 있는 Job의 실행이 종료 될 때까지는 다른 Job은 실행되지 않는다. 이것은 디바이스의 실행을 보장함으로써 지연으로 인한 누설 전력을 감소시킨다. 제안하는 시스템 모델은 그림 2에 나타나 있다. Task1은 디바이스 D1과 D2,D3의 사용에 의해서 2개의 Job으로 분할한다. 그리고 D2,D3는 분할하지 않는다. 이것은 최소 실행 조건을 만족하지 않기 때문이다. 마지막으로 task3은 분할되지 않은 것을 볼 수 있다. 이것은 분할 시 Job의 실행시간이 최소 실행 시간을 만족하지 않기 때문이다.

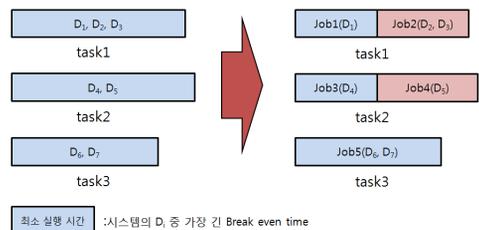


그림 2. 디바이스 관점의 태스크 분할

3.3 디바이스 사용에 집중한 전력 관리 모델

전력 관리 모델을 지원하기 위해서 전력 관리 코드들은 도출된 핵심 관심사 내부에 횡단 결합되어있다. 이러한 코드의 가독성과 유지 보수성 향상을 위해서 본 연구에서는 관점 지향 프로그래밍을 적용하여 전력관리 모델을 지원하도록 한다. 관점으로 고려되어야 할 부분으로는 디바이스의 상태전환을 포함한다. 그림 3은 분할 된 태스크에 삽입되어야 할 전력 관리 코드들의 위치를 보여주고 있다.



그림 3. 전력 관리 코드의 삽입 위치

이러한 전력 관리를 위한 관점 지향 프로그래밍 코드는 Job의 시작점과 Job의 종료점을 가리키는 교차점 선언하고, 각 교차점에서 수행될 코드를 포함한다. Job의 시작점에서는 디바이스를 active 상태로 전환하고, Job의 종료점에서는 디바이스를 sleep 상태로 전환한다. 그림 4는 이러한 aspect코드 형태를 보여준다.

```

aspect power management
1  pointcut declare(Job의 시작점)
2  advice 생성
3  {
4      device를 active state로 전환
5  }
6  pointcut declare(Job의 종료점)
7  advice 생성
8  {
9      device를 sleep state로 전환
10 }
    
```

그림 4. aspect 코드 형태

4. 이론적 평가

본 절에서는 제시한 전력관리 모델에 따라 예제 태스크를 수행시켰을 경우 소모하는 전력을 계산한다. 그리고 그 결과 값을 분할하지 않고 태스크를 수행하였을 경우에 얻어지는 전력 소모량과 비교한다.

수학적 평가를 위해서는 각 Job에서 소모하는 전력을 계산하기 위한 수식이 필요하다. 현재 수행되고 있는 Job에서 소비하는 전력은 3가지 요소의 합으로 나타내어 질 수 있다. 첫 번째 요소는 프로세서와 Job에서 사용하는 디바이스의 전력 소모이다. 두 번째는 Job이 수행을 시작하고 마칠 때 상태 전환을 위한 오버헤드 에너지이다. 마지막으로 Job이 수행되는 동안 sleep state를 유지하는 디바이스에서 소모하는 전력이다. 이를 정리 2를 통해서 확인 할 수 있다.

정리 2: 각 Job에서 소모한 전력은 다음과 같다.

$$E^m(c_{cur}) = (af^3 + \sum_{i|D_i \in D^n} P_a^i) \frac{c_{cur}}{f} + \sum_{i|D_i \in D^n} (E_{sd}^i + E_{wu}^i) + \sum_{i|D_i \in D-D^n} (\frac{c_{cur}}{f} - BET_i) P_s^i$$

Job이 수행하는 동안 시스템 내의 전력 소모는 ccur은 현재 수행 중인 Job의 실행시간이다. a는 프로세서의 switching capacitance이다.

평가를 진행 할 때 많은 태스크를 생성해서 평가하는 것은 무의미하다. 왜냐하면 태스크를 분할하여 실행 시간 중에 디바이스의 누설전력을 줄이는 것을 목표로 하기 때문이다. 그러므로 본 연구에서는 하나의 태스크만을 가지고 평가를 진행한다. 수행 될 태스크 조건은 다음과 같다.

- 디바이스 D1(0.5, 0.1, 5, 5, 10, 10), D2(0.8, 0.2, 5, 5, 10, 10)를 사용한다.
- 최악 경우 실행시간, WCET는 43ms이다.
- switching capacitance, a는 1이다.
- 프로세서의 구동 주파수, f는 fmax=1이다.

수식과 조건을 가지고 계산한 전력 소모량과 분할하지 않고 태스크를 수행 하였을 경우 계산된 전력 소모량의 차이를 분할된 2개의 Job으로 나누어서 그림 5의 그래프에서 보여준다. 여기서는 분할하지 않은 태스크 수행 시 소모된 전력을 각 Job에 따라 나누어 나타내기 위해서 Job의 수행 시간에 비례하게 나누도록 한다. 그래프를 통해서 각 Job의 계산 결과는 분할하지 않은 태스크 수행 시 보다 제안된 알고리즘에 의해 태스크를 분할하였을 경우 전력 소모가 적다는 것을 확인할 수 있다.

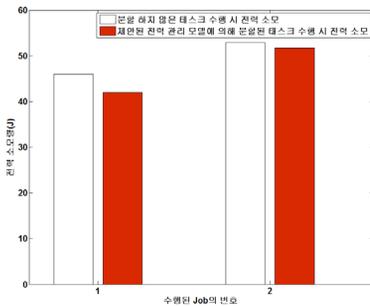


그림 5. 계산 결과 및 전력 비교

5. 실험적 평가

실시간성이 배제된 임베디드 시스템을 고려한 간단한 예제를 통해서 제시한 알고리즘을 평가한다. 성능 평가를 위해서 사용한 임베디드 시스템은 ARM7 32bit 마이크로프로세서를 탑재한 레고 마인드스톱이다. 그리고 인텔 코어 2 Duo 3.0GHz를 탑재한 PC에서 모든 프로그램이 작동하고 블루투스2.0을 사용하여 레고 마인드스톱을 제어한다. 마지막으로 PC에서 작동하는 프로그램에 관점지향 프로그래밍을 적용하기 위해서 AspectJ를 사용하였다. 생성된 코드는 임베디드 시스템에 포팅 가능하지 않다.

본 연구에서 사용하는 예제는 모터, 빛 센서, 초음파 센서를 사용하는 1개의 태스크이다. 태스크는 2개의 Job으로 분할된다. Job1은 빛 센서와 초음파센서를 사용하고, Job2는 모터를 사용한다.

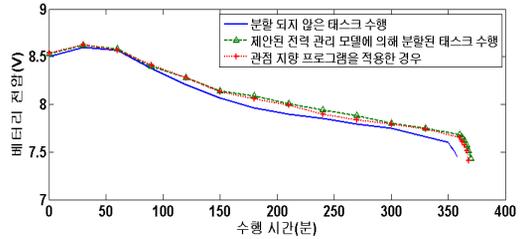


그림 6. 실험 결과

그림 6은 예제 태스크를 분할하지 않고 수행시켰을 경우와 제시한 전력 관리 모델에서 관점지향 프로그래밍을 적용하지 않고 태스크를 수행시켰을 경우 시간에 따른 배터리 전압 강하를 보여준다. 또한 제안된 전력 관리 모델에 관점지향 프로그래밍을 적용하였을 경우 시간에 따른 전압 강하도 보여주고 있다. 그래프를 통해서 제안하는 전력 관리 모델을 사용하였을 경우에 전압이 서서히 감소하는 것과 12분의 수행시간 차이가 있었다는 것을 확인할 수 있다. 이것은 제안한 전력 관리 모델을 사용한 경우가 전력 소모가 적다는 것을 의미한다. 또한 관점 지향 프로그래밍을 적용한 경우에도 많은 오버헤드를 발생시키지 않음을 확인할 수 있다. 이러한 수행 결과들을 표 1에서 보여주고 있다.

표 1. 수행결과 비교

	기존의 전력 관리 방법을 적용한 경우	제안한 전력 관리 방법을 적용한 경우
수행 시간(분)	362	374
평균 전압 강하 (mV/분)	3.525	3.395

7. 결론 및 향후연구

임베디드 시스템의 성능 향상과 대규모 임베디드 시스템의 등장으로 인하여 저 전력과 코드 모듈화의 필요성이 증가하게 되었다. 본 연구에서는 임베디드 시스템에서 사용하는 디바이스의 누설전력을 감소시키기 위한 전력 관리 모델을 제안하였다. 그리고 관점 지향 프로그래밍을 적용하여 핵심 관심사 코드로부터 전력관리 코드를 분리하였다. 이로써 임베디드 시스템의 디바이스에서 발생하는 누설전력을 감소시키고, 전력관리 코드를 모듈화 하였다. 하지만 본 연구에서 제안하는 전력관리 모델은 실시간 임베디드 시스템에 적용하기에는 부족하다.

앞으로의 연구에서는 실시간 임베디드 시스템에 적용할 것이다. 또한, 태스크 분할 자동화를 지원하기 위한 기술과 관점 지향 프로그래밍을 적용하였을 때 발생하는 오버헤드에 대한 연구를 추가적으로 진행할 계획이다.

참 고 문 헌

- [Ayd06] H. Aydin, V. Devadas, D. Zhu, "System-level energy management for periodic real-time tasks," In Proc. IEEE Real-Time Systems Symposium (RTSS), 2006.
- [Dou01] R. Douence, O. Motelet, M. Südholt, "A Formal Definition of Crosscuts," Proc of the Third International Conference on Metalevel Architectures and Separation of Crosscutting Concerns, 2001.
- [Dro02] S. Dropsho, V. Kursun, D.H. Albonesi, S. Dwarkadas, E.G. Friedman, "Managing Static Leakage Energy in Microprocessor Functional Units," Proc. of the IEEE International Symposium on Microarchitecture, MICRO, pp.321, 2002.
- [Dev08] V. Devadas, H. Aydin, "Real-time dynamic power management through device forbidden regions," In Proc. IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS), 2008.
- [Gre05] M. Grechanik, D. E. Perry, D. Batory, "Using aop to monitor and administer software for grid computing environments," In Proc. of COMPSAC'05, Vol. 1, pages 241-248. IEEE, 2005.
- [Hon07] L. M. Honorio, D. A. Barbosa, A.C.Z. Souza, C.V. Lopes, "Intelligent optimal power flow system development using aspect-oriented modeling," IEEE Trans. on Power Systems 22(4), 1826-1834, 2007.
- [Jan96] G. S Jang, Y. S. Moon, C. J. Yoo, O. B Chang, "A Study on Documentation of Object-Oriented Program for Software Maintenance and Reuse," Proc. of the 23th KIISE Spring Conference, vol.35, no.1(A), pp.669-672, 1996.(in Korean)
- [Jej04] R. Jejurikar, R. Gupta, "Dynamic voltage scaling for systemwide energy minimization in real-time embedded systems," In Proc. International Symposium on Low Power Electronics and Design (ISLPED), 2004.
- [Lad03] R. Laddad, "AspectJ in Action," 2nd Ed., p.3, Manning, 2003.
- [Mey87] B. Meyer, "Reusability: The Case for Object-Oriented Design," IEEE Software, vol. 4, no. 2, pp. 50-64, Mar. 1987.

[Na00] H. Y Na, Y. L Choi, S. R Rhew, “A Case Study of Software Maintenance by Database Reverse,” Proc. of the 27th KIISE Spring Conference. vol.27, no.1(A), pp.588-590, 2000.(in Korean)

[Sel05] R. W. Selby. “Enabling reuse-based software development of large-scale systems,” IEEE Trans. Softw. Eng., 31(6):495-510, 2005.

[Sim01] T. Simuinic, L. Benini, A. Acquaviva, P. Glynn, G. D. Micheli, “Dynamic voltage scaling and power management for portable systems,” In Proc. Design Automation Conference (DAC), 2001.

[Swa05] V. Swaminathan, K. Chakrabarty, “Pruning-based, energy-optimal, deterministic i/o device scheduling for hard real-time systems,” ACM Trans. on Embedded Computation Systems, 4(1), 2005.

[Yoo10] G. J. Yoo, E. S. Lee, “Monitoring Methodology using Aspect Oriented Programming in Functional based System,” Proc. of the IEEE International Conference on Advanced Communication Technology, ICACT, Fed 2010.

[Zha09] B. Zhao, H. Aydin, “Minimizing expected energy consumption through optimal integration of dvs and dpm,” in ICCAD, 2009.

저 자 소 개



김 영 민

2010년 중앙대학교 전자전기 공학부 학사
2010년~현재 중앙대학교 컴퓨터 공학과 석박통합과정.

<관심분야> 실시간 소프트웨어, 전력 관리 알고리즘



이 찬 근

1996년 중앙대학교 전자계산학과 학사
1998년 KAIST 전산학과 석사.
2005년 Univ. of Texas at Austin 전산학과 박사.
2005년~2007년 미국 인텔 소프트웨어 엔지니어.
2007~현재 중앙대학교 컴퓨터공학부 조교수.

<관심분야> 실시간 소프트웨어, 수행시간 모니터링, 소프트웨어 테스트