리눅스 기반 임베디드 시스템의 전력소모 분석을 위한 소프트웨어 프로버 설계

(Designing a Software Prober for Power Consumption Analysis of Linux-based Embedded Systems)

남 영 진 † 신 동 석 ††
(Young Jin Nam) (Dong-Seok Shin)
백 장 운 †† 서 대 화 †††
(Jang Woon Baek) (Dae-Wha Seo)

요 약 본 논문에서는 리눅스 기반 임베디드 시스템 상에서 하드웨어적으로 측정된 소모 전력 데이타를 기반으로 보다 효율적인 분석 기능을 제공하기 위한 소프트웨어 프로버를 설계하고 구현한다. 제안된 소프트웨어 프로버는 타겟 시스템에 모듈형태로 포함되어 전력 데이타를 수집하고, 하드웨어적으로 측정된 시간별 소모전력 데이타와의 동기화를 통하여 유저 어플리케이션 내의 유저 함수 혹은 커널 함수 중에서 전력소모가 많은 부분을 찾는 기능을 제공한다. 또한, 구현된 소프트웨어 프로버의 유용성을 타겟 시스템 상에서 유저 어플레케이션 실행을 통하여 검증하였다.

키워드 : 전력소모분석, 소프트웨어 프로버, 리눅스

· 본 연구는 2008년도 경북대학교 BK21사업과 지식경제부 및 정보통신연구진흥원의 대학 IT연구센터 지원사업의 연구결과로 수행되었음 (IITA-2008-C1090-0801-0045)
· 이 논문은 2007 한국컴퓨터종합학술대회에서 '리눅스 기반 임베디드 시스템의 전력소모 분석을 위한 소프트웨어 프로버 설계'의 제목으로 발표된 논문을 확장한 것임

† 정 회 원 : 대구대학교 컴퓨터·IT공학부 교수
yjnam@daegu.ac.kr
†† 비 회 원 : 경북대학교 전자전기컴퓨터학부
zzang9@ee.knu.ac.kr
kutc@ee.knu.ac.kr
††† 종신회원 : 경북대학교 전자전기컴퓨터학부 교수
dwseo@ee.knu.ac.kr

논문접수 : 2007년 10월 2일
심사완료 : 2008년 5월 28일

정보과학회논문지: 컴퓨팅의 실제 및 레터 제14권 제7호(2008.10)

Abstract This paper designs and implements a software prober to support a more effective power analysis for Linux-based embedded systems by using power consumption data measured in hardware. The proposed software prober, which is included in a target system as a module, collects power data and provides a service to discover major power consumers in user functions of applications or kernel functions through time synchronization between the power data measured in hardware and software. In addition, we verified usefulness of the implemented software prober by running user applications on target systems.

Key words : Power consumption analysis, s/w prober, linux

1. 서 론

리눅스는 차세대 휴대폰의 운영체제로서 널리 각광받고 있으며 그 사용이 점차 확대되고 있다. 리눅스는 오픈소스를 기반으로 하고 있기 때문에, WinCE와 같은 임베디드 시스템용 타 운영체제에 비해 라이선스 및 상용화시에 로열티 비용이 없다는 경제적인 장점 뿐 아니라, 커널을 구성하는 모든 소스코드 수준에서의 최적화가 가능하다는 이점도 함께 존재한다.

임베디드 시스템은 특정 기능 수행만을 담당하고 있으며, 소형화, 이동성 및 저전력성 등의 요구사항을 만족하기 위해서 일반적으로 CPU, 메모리, 배터리 등의 하드웨어 자원이 PC 환경에 비해서 매우 제한적이다. 저전력 임베디드 시스템의 효율적인 설계를 위해서는 전력소모를 최소화하는 하드웨어 컴포넌트 설계 뿐 아니라, CPU 속도, LCD 백라이트 밝기, 입출력 장치 전력상태 등을 소프트웨어적으로 통합 제어하는 저전력 설계가 함께 수행되어야 한다. 또한, 구현된 저전력 임베디드 시스템은 탑재한 유저 어플리케이션이 설계시의 배터리 사양을 만족하면서 동작하는지 최종적으로 검증되어야 하고, 그렇지 않을 경우에는 그 원인을 파악할 수 있어야 한다.

임베디드 시스템에서 소모되는 전력을 측정하기 위한 다양한 기법들이 제안되었다. 간단하게는 하드웨어적인 프로버(hardware prober)를 이용하여 임베디드 시스템의 배터리 단에서 소비되는 전력(전류)을 측정할 수 있다. 만일, 각 하드웨어 컴포넌트 회로에서 전력을 측정할 수 있는 단자가 제공될 경우에, 다수의 하드웨어 프로버를 이용하여 각 하드웨어 컴포넌트 별로 소비하는 전력을 동시에 측정할 수 있다[1]. 이때, 각 하드웨어 프로버를 통하여 측정되는 데이타들 간의 시간 동기화가 보장되어야 하며, 이들 데이타는 잘 정의된 형태로 차후 분석을 위해서 저장되어야 한다[1].

하드웨어 프로버에만 의존하는 전력측정의 경우에 전

력소비가 많은 하드웨어 컴포넌트를 찾아낼 수 있지만, 임베디드 시스템에서 동작하는 유저 어플리케이션의 어떤 부분에 의해 많은 전력 소모가 발생하는지에 대한 정보를 찾을 수 없다. 시스템 설계자 입장에서는 유저 어플리케이션 내의 어떤 함수가 어떤 하드웨어 컴포넌트를 사용함으로써 전력소비를 많이 하는가에 대한 정보가 필요하다.

임베디드 시스템 상에 탑재한 유저 어플리케이션이 어떤 함수를 수행하는지를 실시간으로 측정하고 그 측정값을 저장할 수 있는 소프트웨어적인 모듈을 본 논문에서는 소프트웨어 프로버(software prober)로 정의한다. 소프트웨어 프로버는 하드웨어 수준에서의 전력 소모를 실시간으로 측정하는 하드웨어 프로버와 함께 동작하며, 소프트웨어 프로버에 의해 측정된 데이타는 하드웨어 프로버에 의해서 측정된 데이타와 시간적인 동기화를 통하여 높은 전력소비가 발생한 함수를 파악하는 중요한 기초 자료로 이용된다. 본 논문에서는 리눅스 기반의 임베디드 시스템에서 동작하는 유저 어플리케이션을 함수단위로 분석하는 도구인 소프트웨어 프로버를 설계하고 구현하여 시험한 결과를 제시한다.

2. 관련 연구

JouleTrack[2]는 명령어 수준의 전력 분석 값을 명령어 및 사이클 단위로 전력 소모에 대한 가중치를 두고, 사용자가 웹을 통해 실행파일을 업로드하고 실행시간에 이를 분석하여 소모되는 전력분석 결과를 출력해주는 서비스를 제공한다. PowerScope[3]는 외부 하드웨어 멀티미터를 이용하여 주기적으로 시스템의 전력 소모값을 측정하고, 측정값을 프로그램 실행시간에 수집한 프로그램 카운터(PC) 값과 연계 분석하여 전력소모가 많은 프로그램의 영역을 추출한다. 하지만, 이 도구는 이식성이 높은 반면 프로그램의 실행정보를 수집하는 과정에서의 함수 경로가 길어지고 동기화가 일정하지 않기 때문에 정확한 분석이 어렵다. EMSIM[4]은 IntelStrongARM에서 동작하는 임베디드 리눅스를 위한 시뮬레이션 프레임워크를 제공한다. 프로세서 코어, 메모리, 주변기기의 사이클당 소모 전력을 미리 가정하고, 실행시간에 명령어를 분석하여 소모 전력을 예측한다. 그러나 실제 측정된 전력 소모 값과는 정확성이 부족하다는 단점이 있다. SES[5]는 명령어 사이클 수준의 실측을 통해 전력 소모를 수집하고, 수집된 값을 리눅스에서 수행되는 프로그램 소스코드와 연계하여 전력 소모에 대한 정보를 제공한다. 분석 결과를 그래픽 사용자 인터페이스를 통하여 출력한다. 하지만 다른 플랫폼에 대한 전력 측정 시에는 새 하드웨어 및 프로그램 개발이 요구된다. ePRO[6]는 규칙적인 인터벌로 타이머 인터럽트를 이용해 전류값을 샘플링하고 현재 PID(Process ID)와 PC 및 성능 지수를 수집한다. 하지만, 인터럽트 처리 루틴을 이용해 실행 시간에 자료를 수집하고 성능분석을 하기 때문에 리눅스 시스템의 전체적인 성능을 느리게 할 수 있다. 기존연구[1]에서는 동기화된 4개의 하드웨어 프로버를 제공하는 멀티채널 하드웨어 전력측정 시스템을 제시하였다. 하드웨어 프로버는 초당 100~1.8M의 전류 샘플링이 가능하며 1u~3A 측정범위를 갖는다. 4개의 하드웨어 프로버를 제공하는 멀티채널 동기화된 4채널 전력측정이 가능하므로 임베디드 시스템에 장착된 CPU, RAM, LCD, Flash Memory, HDD 등의 모듈별 소모 전력을 측정할 수 있다. 측정 시스템은 동작하는 동안 LCD 화면을 통해 전류 측정 그래프를 출력한다. 측정된 자료는 하드디스크에 기록되며 100GB HDD를 장착 시약 7,000시간 동안의 측정 자료를 저장할 수 있다. 측정 시스템은 모듈별 전력측정, 자료저장, 전류그래프 출력이 가능하며 분석 시스템과의 통합 작업이 필요하다. 하지만, 멀티채널 하드웨어 전력측정 시스템의 경우, 앞서 언급한 바와 같이 하드웨어적으로 측정된 소모 전력이 실행중인 유저 어플리케이션의 어떤 부분에 의해서 발생하는지를 알려주지 못하는 한계가 있다. 본 연구에서는 이를 보완하기 위한 소프트웨어 프로버를 설계한다.

3. 전력소모분석 소프트웨어 프로버 설계

멀티레벨 하드웨어 전력측정 시스템과 함께 동작하는 소프트웨어 프로버 설계에 앞서 소프트웨어 프로버가 가져야 하는 특징을 살펴본다. 첫째, 소프트웨어 프로버는 하드웨어 프로버와 마찬가지로 측정되는 데이타의 높은 정확성과 신뢰성이 요구된다. 또한, 하드웨어 및 소프트웨어 프로버를 통하여 수집된 두 소모전력 관련 데이타의 동기화가 이루어져야한다. 둘째, 소프트웨어 프로버는 임베디드 시스템에 탑재되어 동작하는 소프트웨어 내에 포함되어 동작하기 때문에 높은 이식성이 요구된다. 셋째, 소프트웨어 프로버를 통하여 측정된 소모 전력과 소프트웨어에 대한 수집 정보에 대한 정밀한 분석이 요구되며, 성능이 제약적인 임베디드 시스템 대신 워크스테이션을 통한 분석의 자동화가 필요하다.

3.1 소프트웨어 프로버 구조 및 동작과정

소프트웨어 프로버 구조는 그림 1과 같이 타겟 시스템 리눅스 커널에 포함되어 동작하는 전력 데이타 수집모듈과 수집된 데이타를 분석하는 전력 데이타 분석 모듈로 구성된다. 데이타 수집을 위한 주요자료구조는 원형 큐 형태로 관리되는 DAR(Data Acquisition Record) 버퍼이며, 이 큐에는 주기적으로 샘플링 되는 PC값을 포함하는 DAR 뿐 아니라, 동기화 신호를 저장한다. 저장된 동기화 신호를 이용하여 하드웨어 프로버에 의해 수집된 정보와 동기화를 수행한 후, 최종 소모 전력을 분석한다. 소프트웨어 프로버에 의해 수집된 정보는 타겟 임베디드 시스템

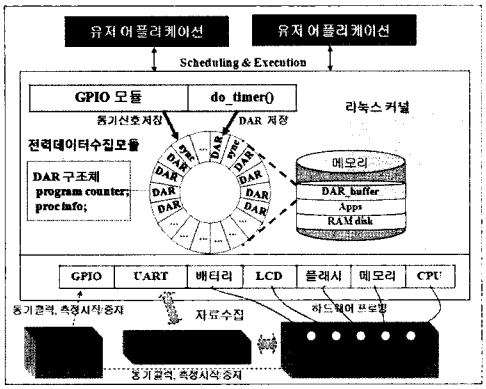

그림 1 소프트웨어 프로버 구조

(이하, 타겟 시스템)에 장착된 네트워크 및 시리얼(UART) 통신을 통하여 외부 시스템으로 받을 수 있다.

소프트웨어 프로버 동작 과정은 다음과 같다.. 첫째, 시리얼(UART) 통신을 이용하여 타겟 시스템과 전력 분석 시스템의 네트워크 파일 시스템 설정, 프로그램 실행, 모듈 적재 등의 초기화 작업을 실시한다. 둘째, 클럭 동기 신호 제어기의 첫 번째 신호가 GPIO 디바이스를 통해 들어오면, GPIO 모듈이 이를 인식하고 타겟 시스템의 전력 데이타 수집 작업이 시작되도록 한다. 동기화 신호 정보는 신호 제어기 설정을 통해 그 주기를 1에서 1.8MHz 범위 내에서 설정할 수 있다. 클럭 동기 신호 제어기의 첫 번째 신호에 의해서 하드웨어 전력 측정 시스템도 동시에 동작하게 되며 지속적으로 전력 데이타 정보(하드웨어의 전류)를 샘플링하고, 동기화 신호를 저장한다. 셋째, 타겟 시스템의 전력 데이타 수집 모듈은 동기화 신호 정보를 GPIO 모듈 인터럽트 서비스 루틴을 통해서, 수행중인 프로그램의 PC값과 프로세스 관련 정보를 포함하는 DAR 정보를 리눅스 커널 내의 클럭 인터럽트 서비스 루틴을 통하여 주기적으로 버퍼에 저장한다. 샘플링 주기는 리눅스 커널의 타이머 인터럽트(jiffy) 시간과 동일하게 10ms로 설정하였다. 넷째, 전력 측정을 종료하기 위해서 클럭 동기 신호 제어기는 클럭 입력을 통해 전력 데이타 수집 모듈과 하드웨어 전력 측정 시스템의 동작을 종료 시킨다. 종료 후, 전력 데이타 수집 모듈은 버퍼에 있는 수집된 자료를 파일로 저장하고 네트워크 파일 시스템을 통해 전력분석 모듈로 전송한다. 분석 모듈로 수집된 자료 전송하기 위해 이더넷을 사용한다. 전력 측정 시스템은 USB와 같은 고속 통신을 이용해 데이타를 분석 모듈로 전송한다. 전력 분석 모듈은 수집도구와 전력 측정 시스템에서 보내온 자료를 통합 및 분석하여 이를 출력한다.

3.2 전력 데이타 수집 모듈

소프트웨어 프로버의 전력 데이타 수집 모듈은 그림 1에서와 같이 소프트웨어 프로버의 핵심 부분으로서 타겟 시스템 상의 리눅스 커널 내에 포함되어 동작한다. 전력 데이타 수집 모듈은 리눅스 커널 내의 PC 값 저장 루틴, DAR 데이타 버퍼 및 GPIO 모듈(ISR)로 구성된다. 앞서 동작과정에서 설명한대로 리눅스 커널에서 제공되는 타이머 인터럽트는 10ms 주기로 발생하며, 이때 스케줄러에 의한 프로세스 문맥교환이 일어난다. 타이머 인터럽트 처리루틴(do_ timer())은 스케줄링을 위해 현재 수행중인 프로세스의 레지스터 구조체에 대한 포인터를 함수 파라미터로 포함하고, 이 값을 참조하여 PC 값을 알 수 있다. 데이타 수집 처리 루틴은 PC 값과 프로세스를 식별할 수 있는 정보를 DAR 버퍼의 구조체 배열에 저장한다. 버퍼는 메모리 사용량을 최소화하기 위해 원형 큐 자료구조를 이용하였다. 원형 큐 자료구조에서는 덮어 쓰기로 인해 기존 저장 데이타 손실이 발생할 수 있으므로, 버퍼의 상한선을 정하고 초과하면 현재 버퍼의 내용을 파일로 자동 저장하도록 설계하였다. 데이타 수집 주기는 커널 설정을 통해 수정할 수 있고, 분석하고자 하는 소프트웨어가 실행되는 리눅스의 시스템 타이머에 맞춰 수정할 수 있다.

전력 데이타 수집 모듈은 리눅스 커널 내에서 빈번히 호출되어 수행되기 때문에 전체적인 성능에 최대한 영향을 미치지 않도록 코드의 크기를 최소화 하였다. 본 모듈에서 수집된 자료는 타겟 시스템에 이더넷 장치가 존재할 경우, 네트워크 파일 시스템을 통하여 수집된 자료를 전송하도록 하였다. 또한, 전력 데이타 수집 모듈은 타이머 인터럽트 수정 및 DAR 버퍼를 위한 커널 패치와 GPIO 모듈 적재를 통하여 기존 리눅스 커널에 쉽게 탑재 가능하므로 높은 이식성을 제공할 수 있다.

3.3 전력 데이타 분석 모듈

소프트웨어 프로버의 전력 데이타 분석 모듈은 그림 2과 같이 데이타베이스(DB) 생성기와 전력 데이타 분석기로 구성되어 있다. DB 생성기는 심볼테이블 DB 생성기와 코드 세그먼트 DB 생성기를 포함한다. DB 생성기 내의 심볼테이블 DB 생성기는 유저 어플리케이션 이미지와 리눅스 커널 이미지로부터 심벌 테이블을 추출하며, 이를 이용하여 해당 주소값과 관련 함수 쌍의 집합으로 이루어진 심볼테이블 DB를 만든다. 코드 세그먼트 DB 생성기는 각 이미지로부터 세그먼트 텍스트를 추출하여 해당 주소값, 관련함수, 관련 어셈블리 코드 쌍으로 구성된 코드 세그먼트 DB를 구축한다.

전력 데이타 분석기는 함수레벨 분석기와 코드레벨 분석기로 구분된다. 함수레벨 분석기는 심볼테이블 DB, 전력 데이타 수집 모듈로부터 가져온 소프트웨어 프로빙 된 전력 데이타 정보, 그리고 하드웨어 프로빙된 전력 데이타 정보를 이용하여 유저-함수레벨 분석 데이타와 유저/커널 통합-함수레벨 분석 데이타를 생성한다. 유저-함수레벨 분석은 전력 데이타 수집 모듈을 통하여 측정한 주

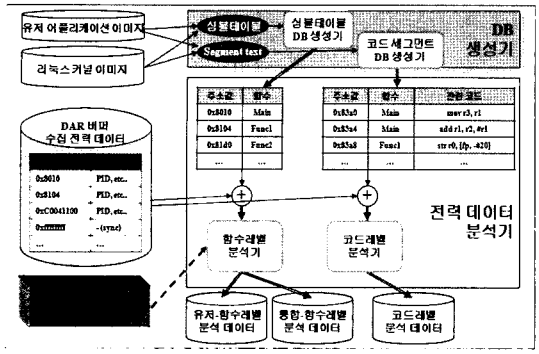

그림 2 전력 데이타 분석 모듈 구조

소 값들 중에서 유저 영역 주소 값과 심볼테이블 DB를 사용한다. 우선, 측정된 유저 영역 주소 값을 이용하여 심볼테이블 DB 내의 해당 유저 함수를 맵핑한다. 맵핑된 각 유저 함수에 대해서 그 측정 데이타에 포함되어 있는 비율에 따른 값을 출력한다. 이 분석 결과를 이용해 하드웨어 프로버를 이용하여 측정한 특정 전력소모 구간에 수행된 유저 어플리케이션 함수 실행 비율을 알 수 있다. 결과적으로 유저-함수레벨 분석을 통하여, 주어진 유저 어플리케이션 실행시에 높은 전력소모를 일으키는 유저 함수를 찾아 낼 수 있다. 분석시에 하드웨어 프로빙 된 데이타와의 동기화는 소프트웨어 프로빙시에 DAR 버퍼에 함께 주기적으로 저장된 동기화 정보를 이용하여 이루어진다.

유저/커널 통합-함수레벨 분석은 전력 데이타 수집 모듈을 통하여 측정된 모든 주소 값들을 대상으로 한다. 즉, 수집된 유저 영역 주소 값과 커널 영역 주소 값에 대해 심볼테이블 DB를 이용하여 해당 유저 또는 커널 함수를 결과에 추가하고, 커널 영역을 호출한 유저 함수에 대해 실행 비율 가중치를 적용한다. 이러한 통합 분석 결과를 이용해 시스템 콜과 같은 커널 함수를 호출하는 함수에 대한 분석이 가능하다. 또한, 특정 어플리케이션이 주로 사용하는 커널 함수 등과 같이 어플리케이션과 커널을 연계한 분석기능을 제공할 수 있다.

전력 데이타 분석기의 코드레벨 분석기는 코드 세그먼트 DB, 전력 데이타 수집 모듈로부터 가져온 소프트웨어 프로빙된 전력 데이타 정보, 그리고 하드웨어 프로빙된 전력 데이타 정보를 이용하여 코드레벨 분석 데이타를 생성한다. 코드레벨 분석기는 전력 데이타 수집 모듈로부터 소프트웨어 프로빙된 PC 주소값 중 사용자가 선택한 특정 주소에 대한 정보를 분석한다. 즉, 코드 세그먼트 DB를 이용해 열람하고자 하는 주소값을 검색하고, 코드가 위치하는 함수, 해당 어셈블리 코드, 그리고 사용 추출 빈도를 제공한다. 본 분석방법은 함수레벨의 분석보다는 보다 세밀한 수준의 정보를 제공한다는 장점은 있지만, 함수레벨의 분석에 비해서 샘플링에 따른 오차가 발생하는 확률이 높아진다는 단점이 있다.

전력 데이타 분석 모듈은 유저 어플리케이션 이미지 및 커널 이미지 자체를 이용하기 때문에 직접 전력 측정 관련 코드를 유저 어플리케이션 소스 코드에 넣는 방법에 비해 이식성 및 사용의 편리성 측면에서 이점을 갖는다.

4. 소프트웨어 프로버 기능 및 성능 검증

4.1 구현결과 및 시험환경

소프트웨어 프로버 구현 환경은 호스트 PC(전력 데이타 분석 모듈)와 타겟 시스템(전력 데이타 수집 모듈)을 서버-클라이언트 형태로 구성하였고, 유저 어플리케이션으로 MP3 Player 프로그램과 카메라 구동 프로그램을 사용하여 소프트웨어 프로버를 시험하였다. 타켓 시스템과 호스트 PC의 사양은 표 1과 같다.

표 1 소프트웨어 프로버 구현 환경

구 분	사 양
타겟시스템 (전력 데이타 수집 모듈)	Xcale255 CPU(500MHz), 32MB RAM, LCD, 64MB NOR Flash Memory, 10Mbps Ethernet, Linux Kernel 2.4.18
호스트 PC (전력 데이타 분석 모듈)	펜티엄 2.4GHz, 1GB RAM, 160GB HDD, Linux Kernel 2.6.18 (Fedora Core 6), MySQL
유저 어플리케이션	MP3 Player, 카메라 구동 프로그램

소프트웨어 프로버 사용자 인터페이스는 테스트 프로그램 리스트, 측정 전류 그래프, 실행구간지정 입력박스, 프로세스명과 프로그램 카운터, 커널(K)/유저(U)영역의 해당함수, 해당함수를 호출한 함수로 구성된다.

4.2 유저 어플리케이션 시험 결과

소프트웨어 프로버의 성능을 테스트하기 위해 타겟 임베디드 시스템에서 동작하는 유저 어플리케이션 중에서 MP3 player 프로그램(madplay)과 카메라 구동 프로그램(QT Camera)의 전력을 측정하고 분석하였다.

본 논문에서 소프트웨어 프로버는 10ms 마다 프로그램의 전류값을 샘플링한다. 전압이 일정하므로, 각 함수에 의해 소모된 에너지는 전류 샘플링 값과 샘플링 된 횟수에 비례한다. 본 논문에서는 전체 프로그램 에너지 소모량 대비 각 함수의 상대적인 에너지 소모량을 이용하여, 프로그램에서 각 함수의 소모 에너지를 비교하였다. 함수 a의 상대적인 에너지 소모량(E_a)은 아래의 식과 같이 정의되며, I_t는 전체 전류 샘플링 값의 합을 나타내고, n_a는 함수 a의 총 샘플링 수를 나타낸다:

$$E_a = \frac{1}{I_t} \times \sum_{k=1}^{n_a} i_k \times 100(\%)$$

표 2는 MP3 player 프로그램에서 샘플링 빈도가 큰 함수의 에너지 소모량을 비교한 것이다. 타겟 시스템 상에 MP3 player 프로그램을 20초 동안(2,000번 샘플링) 실행하였고, 모든 함수에 대한 전체 전류 샘플링 값의 합은 536mA 이었다. 음악파일을 재생하기 위한 코덱 라이브러리에서 전체의 에너지 소모의 50%정도를 소모함을 알 수 있다. 사용자함수 중에서는 음원을 출력하기 위한 pcm 재생 관련 함수인 audio_pcm_s16le에서 22%의 에너지를 소모하였고, 사운드 출력관련 함수(output)과 프로그램 실행 중 사용자 설정 및 커맨드 입력관련 함수(audio_oss)에서 각각 6%, 3%의 에너지를 소모하였다. 오디오 관련 시스템 콜에서 9.7%의 에너지를 소모하였다.

표 2 MP3 Player 프로그램의 함수 에너지 소모량 비교

	함수명	함수 설명	소모 에너지비
사용자함수	audio_pcm_s16le	pcm 재생 관련 함수	22%
	output	사운드 출력 관련	6%
	audio_oss	설정 및 커맨드 입력	2.8%
라이브러리	libmad.so	MP3 음악파일 코덱	50.7%
	libid3tag.so	MP3 파일 정보 출력	2.7%
시스템콜	audio_write	오디오 관련 시스템콜	9.7%

표 3은 타겟 시스템에 장착된 CMOS 카메라를 구동시키는 프로그램을 실행하고 분석한 것이다. 프로그램은 카메라 모듈로부터 입력되는 이미지 데이타를 주기적으로 LCD에 출력하고, 사진 촬영과 사용자 요청이 있을 때 화면상의 이미지를 JPEG 파일로 저장한다. 본 실험에서 타겟 시스템 상에 카메라 구동 프로그램을 13초 동안(1,300번 샘플링) 실행하였고, 모든 함수에 대한 전체 전류 샘플링 값의 합은 369mA이었다. 프로그램은 대부분 라이브러리에서 실행되었고 전체 소모 에너지의 70% 이상을 차지하였다. 특히, 커널 시스템 콜에서는 *.c2u_0cpy8lp*라는 사용자 페이지폴트 처리 루틴 자주 실행되었고 전체 에너지 소모량의 16%나 차지하였다. 이를 통해, 유저 어플리케이션이나 커널의 메모리관리에 문제가 있음을 알 수 있었다. 이 *.cu_0cpy8lp* 시스템 콜을 호출하거나 폴트로 인해 호출되기 전에 실행되었던 사용자영역의 주소 중 특정하나를 추출한 결과 사용자 라이브러리 영역임을 알 수 있었다. 따라서 해당 주소에 대한 라이브러리 소스코드를 수정함으로써 문제를 해결할 수 있다.

소프트웨어 프로버를 통해 소비되는 전류 그래프를 보고 프로그램 실행의 전체 또는 특정 시점에 대한 함수 실행 정보를 분석할 수 있으며, 사용자 프로그램의 함수 수행 패턴, 사용자 프로그램에서 사용하는 커널 함수의 사용 패턴 및 전력소모가 많은 함수를 분석할 수 있었다.

표 3 MP3 Player 프로그램의 함수 에너지 소모량 비교

	함수명	함수 설명	소모에너지비
라이브러리	-	- (비공개 라이브러리)	72%
시스템 콜	.c2u_0cpy8lp	사용자 페이지 폴트처리 루틴	16%
	EmulateAll	실수연산 명령 emulate	3%
	get_irq_list	인터럽트 리스트 구함	2.6%

5. 결론 및 향후 연구계획

본 논문에서는 타겟 시스템에 모듈형태로 포함되어 전력 데이타를 수집하고, 하드웨어 프로버에 의해서 측정된 시간별 소모전력 데이타와의 동기화를 통하여 유저 어플리케이션 내의 유저 함수 혹은 커널 함수 중에서 전력소모가 많은 부분을 찾는 기능을 제공하는 소프트웨어 프로버를 설계하고 구현하였다. 소프트웨어 프로버의 타겟 시스템 의존적인 전력 데이타 수집 모듈은 타겟 시스템에 사용되는 리눅스 커널 소스에 간단한 패치 및 모듈추가를 통하여 이식이 용이하도록 하였다. 또한, 타겟 시스템 독립적인 전력 데이타 분석 모듈은 하드웨어 프로버를 통하여 수집된 전력 데이타, 전력 데이타 수집 모듈을 통하여 받은 데이타, 리눅스 커널 및 유저 어플리케이션 이미지로부터 구축된 데이타베이스를 이용하여 다양한 레벨의 전력 분석이 자동적으로 실행되도록 설계 및 구현하였다. 또한, 구현된 소프트웨어 프로버의 유용성을 타겟 시스템 상에서 유저 어플리케이션 실행을 통하여 검증하였다.

참 고 문 헌

[1] Y. Nam, E. Yang, J. Lee, S. Kim, D. Seo, "Measurement-based Analysis of Power Consumption Patterns for Portable Multimedia Players," Proc. of Embedded Systems and Applications, Jun. 2006.

[2] A. Sinha, A. Chandrakasan, "JouleTrack-A Web Based Tool for Software Energy Profiling," Proc. of Design Automation Conference, Jun. 2001.

[3] J. Flinn, M. Satyanarayanan, "PowerScope: A Tool for Profiling the Energy Usage of Mobile Applications," Proc. of Workshop on Mobile Computing Systems and Applications, Feb. 1999.

[4] T. Tan, A. Raghunathan, N. Jha, "EMSIM: An Energy Simulation Framework for an Embedded Operating System," Proc. of IEEE International Symposium on Circuits and Systems, May 2002.

[5] 김영진, 백웅기, 김지홍, "저전력 소프트웨어 개발을 위한 전력 분석 도구", 정보처리학회지, 2004.

[6] W. Baek, Y. Kim, J. Kim, "ePRO: A Tool for Energy and Performance Profiling for Embedded Applications," Proc. of International SoC Design Conference, Oct. 2004.