

평면 디스플레이를 위한 태스크 기반

저전력 내장형 소프트웨어 구현

하성호^o, 전승훈

삼성전자 디지털미디어 연구소

Implementation of Task-Based Low-Power Embedded SW for Flat Panel Display

Seong-Ho Ha^o, Seung-Hun Jeon

Digital Media R&D Center, Samsung Electronics Co., LTD

knight.ha@samsung.com, shjeon01@samsung.com

요 약

최근들어 내장형 시스템의 설계에 있어서 고도의 집적화와 동작주파수 증가 및 이동형 시스템 수요 증가로 인해 전력 소모에 대한 중요성이 계속적으로 증가하고 있다. 이러한 제약을 해결하기 위해 태스크 형태의 소프트웨어가 나타나기 시작했다. 기존의 일반적인 내장형 시스템에서는 슈퍼루프(Super-loop) 형태의 소프트웨어가 기반을 이루고 있는데 이 구조내에서는 저전력 모드 구현이 어렵다. 저전력 모드 진입 시점과 복귀 시점을 정하기가 어렵고 바꾸더라도 전체 시스템에 영향을 미치기 때문에 전체를 다 수정해주어야 한다. 하지만 태스크 형태의 내장형 소프트웨어에서는 각 태스크들이 독립적으로 돌아가고 태스크 내에서만 저전력 모드를 적용하게 되면 쉽게 저전력 모드 구현이 가능하다. 이는 기능의 다양화와 복잡성에도 쉽게 대응할 수 있다는 일반적인 태스크 형태의 소프트웨어가 갖는 특징을 잘 나타내준다. 일반적으로 태스크 구조의 소프트웨어는 재사용성이 높아지고 실시간 운영체제를 사용함으로써 실시간 성능이 향상된다. 본 논문에서 보여주는 모델은 디바이스에 의존적이면서 빠른 응답시간을 요구하는 평면 디스플레이를 위한 소프트웨어이다. 태스크 기반의 소프트웨어에서 유휴 상태(idle state)를 활용하는 것을 기반으로 구현하였고, 이는 기존 슈퍼루프 형태의 소프트웨어에 비하여 전력소모량이 줄어들음을 보여준다.

1. 서론

개인 PC 나 워크스테이션, 서버로 불리는 컴퓨터 시스템은 복잡한 운영체제 위에 워드 프로세서, 엑셀, 게임, 인터넷 서핑 등의 개인용 응용 프로그램에서부터 인터넷 서비스, 데이터베이스 서비스 등의 방대한 프로그램들을 수행한다. 반면 내장형 시스템은 TV, 핸드폰, 디지털 카메라,

인공위성, 자동차용 전자회로, 의료기기, MP3 플레이어 등 겉으로 보기에 컴퓨터라기보다는 일반적인 스마트 전자기기로 볼 수 있다.

이런 내장형 소프트웨어의 응용범위가 커짐에 따라 시스템의 성능 이외에도 시스템의 전력 소모량이 내장형 시스템 설계 있어서 중요한 요소로 고려되어야 하게 되었다.

이에 따라 수년전부터 전력 소모를 줄이기 위한 많은 연구들이 이루어졌다.

일반적으로 전력소모량 P는

$$P = K \times C_{out} \times V_{dd}^2 \times f$$

K : Activity factor

C_{out} : Total chip capacitance

V_{dd} : Supply voltage

f : Clock frequency

로 나타내지는데 전력 소모를 줄이는 방법에는 크게 2가지 방법이 있다. 첫번째 방법은 CPU가 유휴상태일때 클럭발생기의 클럭수를 줄여 CPU에 공급되는 클럭수를 줄이는 저전력 모드를 이용하는 것이다. 최근 나온 CPU에서는 대부분 DPMS 등의 저전력 모드를 지원한다. 이 방법은 전력 소모를 줄여주는 하나 모드를 바꾸는데 있어서 생기는 지연으로 인해 실시간 시스템에서는 다소 제약적이다.

두번째 방법은 입력 전압을 줄여 CPU의 속도를 동적으로 변화시키는 방법이다. 최근의 CPU들은 입력 전압을 조절하여 속도와 소모 전력 사이의 이해득실을 조절할 수 있는 기능을 갖춘 것들이 많다. 대표적으로 AMD Mobile K6, Intel XScale 등이 있다. 보통의 경우 공급 전압이 낮아짐에 따라 그의 제곱에 비례하여 소모 전력을 줄일 수 있고, 또한 전압을 낮춤과 동시에 회로의 동작 주파수를 낮추게 되어서 프로세서에서 소모되는 전력은 더욱 작아진다.[2][3][4][5][6]

그러나 평면 디스플레이와 같은 시스템에서는 실시간 특성보다는 빠른 응답시간을 요구하고 스케일이나 기타 다른 디바이스의 응답 속도에 의존하기 때문에 클럭을 조절할 수 있는 CPU가 쓰이지는 않는다.

따라서 본 논문에서는 첫번째 방법을 사용해서 그 전력소모를 줄이는 것을 보여준다. 2장에서는 관련된 연구에 대한 내용을 언급하고, 3장에서는 구현방법에 대해 설명하고, 4장에서는 결과 및 성능에 대해 설명한다. 마지막으로 5장에서는 결론을 기술한다.

2. 관련 연구

2.1 슈퍼 루프 구조

이전부터 주로 내장형 소프트웨어에서는 슈퍼루프, 즉 메인 보드에 전원이 공급된 이후 프로그램 전체가 루프를 돌면서 돌아가면서 기타 인터럽트 입력에 따라 필요한 곳에

서 동작을 하는 구조로 구현되었다. 이 같은 구조는 빠른 시간에 구현이 가능하고 코드 이해가 빨라 쉽게 구현할 수 있는 장점이 있다. 하지만 구조가 서로 시간축에 대해 의존적이기 때문에 기능이 추가되거나 변경될 경우 전체 시스템에 대해 코드 수정 및 타이밍 수정이 필요하다. 또한 최악의 수행시간(WCET)에 대한 편차가 커져 응답속도에 대해 신뢰적이지 못하다. 순차적으로 실행되어야 하는 모듈들이 적절하게 배치되지 않으면 최악 수행시간을 모듈의 수만큼 반복해야 하는 경우가 생길 수도 있다. 모듈 A, B, C, D가 A-B-C-D 순으로 배치되어 있다고 하자. 이 모듈들이 제대로 동작하기 위해서는 D-C-B-A 순으로 동작해야 한다고 하면 이 시스템은 A->B->C->D->A->B->C->D->A->B->C->D->A의 과정을 거쳐 제대로 동작하게 되고 이 같은 경우 더욱더 설계 및 수정이 힘들어진다. 그림 1은 슈퍼루프 형태로 구현된 소프트웨어의 예이다.

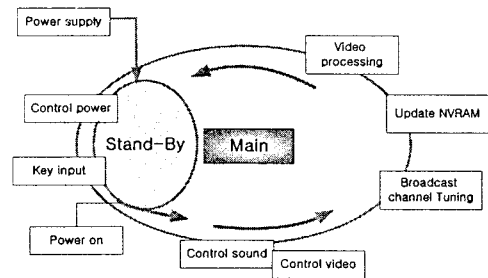


그림 1. 슈퍼루프 형태의 내장형 소프트웨어

2.2 태스크 구조

일반적으로 태스크 구조의 모델은 기능의 추가나 기타 다른 상황에 대해 유연하게 대처하기 위해 생겨났다. 태스크는 서로 비슷한 기능을 하는 부분들을 묶어 다른 것과는 독립적으로 동작할 수 있게 만든 것이다.

태스크 기반의 구조를 구현하기 위해 내장형 시스템에서는 대부분 RTOS(Real-Time OS)를 사용한다. OS를 기반으로 그 위에 여러가지 태스크들이 돌아가게 되고 OS는 각 태스크들이 유기적으로 돌아가게 하기 위해 적절하게 스케줄링 해준다. 각 태스크들은 서로간에 데이터를 주고 받거나 공유자원을 같이 사용하지만 태스크 자체는 각각 독립적으로 돌아간다. 따라서 특정 부분을 수정할 경우에는 그 부분이 포함된 태스크만 수정해주면 되고 기능 추가의 경우

에도 태스크를 추가해 태스크 사이의 동작만 추가해 주면 된다. 이 같은 장점으로 인해 여러가지 기능이 포함된 내장형 시스템에는 태스크 기반의 소프트웨어가 사용된다. 그림 2는 태스크 형태로 구현된 소프트웨어의 예이다.[1][7]

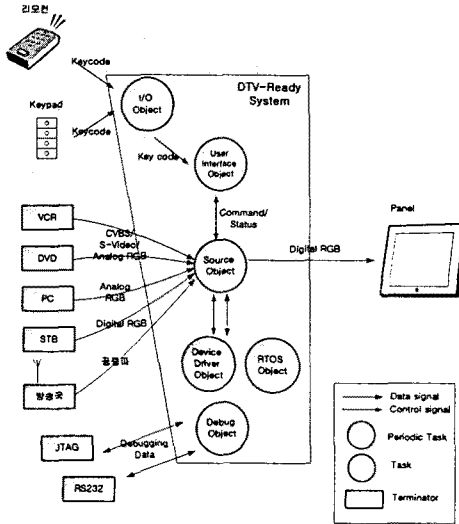


그림 2. 태스크 형태의 내장형 소프트웨어

2.3 평면 디스플레이 시스템

평면 디스플레이 시스템은 데드라인이 정해져 있어 그 시간내 수행을 해야하는 시스템이라기보다는 빠른 응답시간을 요구하는 시스템이다. 단지 시스템을 구성하고 있는 디바이스들이 각기 다른 세팅시간을 요구하고 있고 각 디바이스간의 의존성으로 인해 CPU는 유휴상태에 들어가지만 각종 디바이스에는 계속적으로 클럭공급이 유지되어야 한다. 그래서 제공되는 저전력 모드도 CPU에는 클럭이 공급되지 않더라도 기타 주변 디바이스들에는 계속적으로 클럭이 공급된다.

3. 구현

본 논문에서 사용된 실험 환경은 다음과 같다. CPU는 C166 Core를 사용하였고, 에뮬레이터 장비는 Trace32-ICD를 사용하였다. 태스크 기반의 소프트웨어에서 스케줄링 방법은 고정 우선순위를 사용하였으며 동일 우선순위의 경우 FIRO(First In First Out)를 사용하였다. CPU는 3.3V의 입력 전압을 가지고 노멀모드에서는 33MHz로 동작하고 유휴상태에서는 3MHz로 동작하게 되어 있어 1/10의 스케일을 가진다. 첫번째 방법을 사용, 태스크 기반의 소프트웨

어로 바꾸기 위해 하기 위해서 상용 RTOS(Real-Time OS)인 Nucleus를 사용하였으며 유휴상태를 확인하기 위해 uC/OS처럼 idle 태스크를 두어 idle 태스크가 실행되고 있을 때에는 저전력 모드로 진입하게 하였다.[8]

본 논문에서는 내장형 소프트웨어의 전력을 측정하는 것을 보여주는 것이므로 전력측정에 있어서 디스플레이 패널 자체에서 소모하는 전력은 고려하지 않았다. 그리고 전력량 비교는 시간을 고려해야 하지만 여기서는 시간을 고려하지 않고 편이상 0.1 초 단위로 샘플링한 평균 주파수로 비교하였으며 모드 변경에 발생하는 인터럽트 지연시간은 고려하지 않았다.

4. 실험 결과

슈퍼 루프 형태의 평면 디스플레이를 위한 내장형 시스템은 항상 일정한 주파수를 가지고 동작하였지만 태스크 구조로 바꾸어 저전력 모드를 구현했을 때 결과는 그림 4, 그림 5, 그림 6과 같은 결과를 보였다. 소모 전력은 주파수에 비례하기 때문에 여기서는 단지 주파수로 비교를 하였다. 그림 4에서는 전원을 켜고 후 대기상태에서 전원 버튼을 눌러서 TV를 켜고 후 다시 켜는 때의 평균 주파수를 보여준다. 그림 4에서도 보듯이 전원을 켜고 순간과 끌 순간에는 최고 주파수로 동작하고 기타 입력이 없을 경우에는 저전력 모드로 동작이 가능하였다. 단지 TV가 켜진 상태에서의 무동작 경우일 때와 TV가 꺼진 상태에서의 무동작 경우일 때 각각 평균 주파수가 4.5MHz, 9.4MHz로 나타났다. 화면이 나오는 순간에는 신호를 계속적으로 체크하기 때문에 두 경우의 평균 주파수에 차이가 있었으며 최대 주파수 일때와 비교해보면 화면이 꺼진 상태에서는 약 86.5%의 전력소모를 줄일 수 있고 화면이 나오는 상태에서는 약 71.8%의 전력소모를 줄일 수 있었다. 그림 4에서는 메뉴 버튼을 눌러 화면상에 OSD(On Screen Display)를 띄웠다가 다시 메뉴 버튼을 눌러 사라지게 했을 때의 주파수 변화를 보여준다. 그림 5에서는 앞부분의 피크는 볼륨을 증가시켰을 때, 뒷부분의 피크는 채널을 변경시켰을 때의 평균 주파수 변화를 보여준다. 각각의 경우 그 수행시간은 틀리지만 사용되는 디바이스의 설정이 끝난후에는 같은 주파수를 보였다. 그림 4, 5, 6에서 공통적으로 입력이 있었던 순간에는 최고 주파수로 동작하고 입력이 없는 순간에는 거의 동일한 주파수를 보여 실제 평면 디스플레이 시스템에서 디바이스를 세팅한 이후에는 많은 시스템 내부에서 많은 동작이 일어나지 않아 그 시간동안 전력소모를 줄일 수 있음을 보여준다.

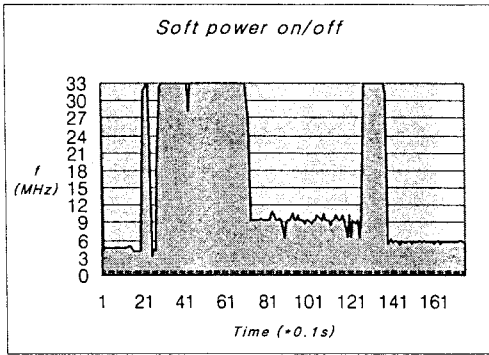


그림 3. Soft power on/off 시 평균 주파수

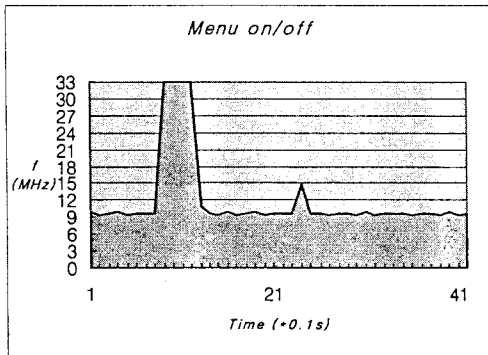


그림 4. Menu on/off 시 평균 주파수

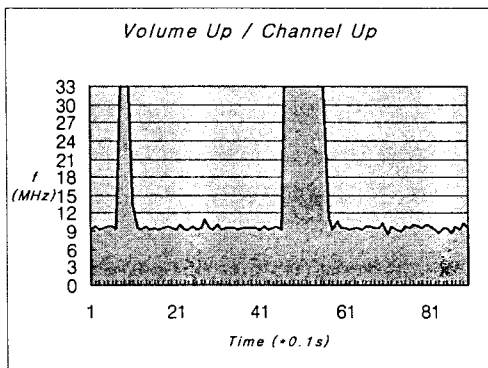


그림 5. Volume up / Channel up 시 평균 주파수

5. 결론

본 논문에서는 슈퍼루프 형태의 소프트웨어를 고정 우선 순위를 갖는 태스크 기반의 소프트웨어로 바꾸어 저전력

모드를 구현, 그 결과를 보여주었다. 이론적으로 태스크 형태의 소프트웨어가 속도나 재사용성에 있어서 향상된 성능을 나타내는데 전력소모면에서도 기본적으로면서도 평이한 방법으로 평면 디스플레이의 특성을 고려, 전력소모를 줄일 수 있음을 보여주었다. 추후 빈번한 주파수 변경과 이로 인한 지연시간 사이에 대한 적절한 연구가 이루어진다면 좀 더 개선된 저전력 모드 구현이 가능할 것이다.

6. 참고 문헌

- [1] Hassan Gomma, Software design methods for concurrent and real-time systems, Addison-Wesley, 1993.
- [2] Y. Shin and K. Choi, "Power Conscious Fixed Priority Scheduling for Hard Real-Time Systems," Proceedings of Design Automation Conf., pp.134-139, 1999.
- [3] William Fornaciari, Paolo Gubian, Donatella Sciuto, Cristina Silvano, "Power Estimation of Embedded Systems : A Hardware/Software Codesign Approach," IEEE Transactions on VLSI Systems, Vol.6, No.2, Jun. 1998.
- [4] Higami, Y., Kobayashi, S.Y., Takamatsu, Y., "A Method to Reduce Power Dissipation during Test for Sequential Circuits," Test Symposium, 2002. (ATS '02). Proceedings of the 11th Asian, p326-331, Nov. 2002
- [5] Robert P.Dick, Ganesh Lakshminarayana, Anand Raghunathan, Niraj K.Jha. "Analysis of Power Dissipation in Embedded Systems Using Real-Time Operating Systems," IEEE Transactions on Computer-Aided Design of Integration Circuits and Systems, Vol.22, No.5, May 2003
- [6] Kumar, P., Srivastava, M., "Predictive Strategies for Low-Power RTOS Scheduling," Computer Design, 2000. Proceedings. 2000 International Conference, p343-348, Sep. 2000
- [7] 전승훈, 이종인, "디지털 티브이-레디를 위한 태스크 기반의 내장형 소프트웨어", 한국전자공학회, 제 26 권, 제 5 호, pp. 36-47, 2003 년 8 월.
- [8] -, uC/OS-II, Lawrence, KS : R&D Books, 1998