

# 임베디드 시스템을 위한 동적 전력관리 미들웨어 설계 및 구현

이수일\*, 권기현\*, 변형기\*, 김남용\*, 이규영\*

\*강원대학교 전자정보통신공학부

e-mail:kweon@kangwon.ac.kr

## Design and Implementation of DPM Middleware supporting for Embedded System

Sull Lee\*, KiHyeon Kwon\*, HyungGi Byun\*,

NamYong Kim\*, GyooYoung Lee\*

\*School of Electronic, Information Technology Kangwon Natl. University

### 요 약

다양한 휴대용 정보 기기의 사용이 보편화됨에 따라 배터리 기반의 임베디드 시스템에서는 저전력 요구가 크게 대두되고 있다. 이에 본 논문은 Xscale PXA255 기반 Embedded Linux 환경에서 전력 소비를 줄이기 위해 DPM(Dynamic Power Management) 미들웨어를 설계 및 구현하고 이에 따른 디바이스 드라이버를 제작하여 임베디드 타겟보드의 전력을 감소시키는 방법을 보인다.

### 1. 서론

다양한 휴대용 정보 기기의 사용이 보편화됨에 따라 배터리 기반의 임베디드 시스템에 대한 저전력 요구가 크게 대두되고 있다. 기존의 전력을 고려한 설계 기법은 주로 회로 수준에서 이루어져 왔기 때문에 하드웨어적으로 구현되어 왔으나 마이크로프로세서의 성능이 향상되고 네트워크 대역폭이 커짐에 따라 휴대용 정보 기기 상에서의 동화상, 음악, 게임 등의 다양한 기능 및 고성능 요구로 인해 소프트웨어의 전력 소모가 커지고 있다.

따라서 본 논문에서는 효율적인 전력 소모 관리를 위해 소프트웨어 수준에서의 전력 관리를 위해 임베디드 시스템과 사용자 사이의 미들웨어를 설계하고 구현하여 전력을 감소시키는 방법을 보였다.

논문의 구성은 2장에서는 기존 저전력 시스템에 대한 관련연구, 3장에서는 DPM 미들웨어 설계 및 구현, 4장에서는 실험 및 검토, 5장에서 결론 및 향후 연구방향에 대해서 제시한다.

### 2. 관련 연구

2.1 동적 전압 조절 기법(Dynamic Voltage Scaling :

DVS)

동적 전압 조절 기법은 CMOS 회로로 구성되는 프로세서의 공급 전압을 온라인 상태에서 조절하여 에너지 소모량을 줄이는 기법이다. 일반적으로 CMOS회로의 전력(P) 소모는 공급 전압에 대해 제곱의 관계를 가지므로, 공급 전압의 감소는 에너지 소모를 줄일 수 있는 매우 효과적인 방법이다. 하지만, 공급 전압이 낮아질 때에는 회로의 최대 동작 주파수 또한 선형적으로 느려지는 관계로 인해 프로세서의 처리량도 줄어들게 된다. 이러한 프로세서의 처리속도와 에너지 소모에 대한 관계를 이용하여 주어진 작업이 요구하는 서비스의 질을 훼손하지 않는 범위 내에서 전압 및 클럭 속도를 설정하는 기법을 동적 전압 조절 기법이라 한다[1-3].

2.2 동적 전력 관리 기법(Dynamic Power Management : DPM)

동적 전력 관리 기법은 시스템의 구성 요소별 사용 패턴에 기반을 두어 해당 장치가 비활성화(inactive)상태에 있는 경우 해당 장치가 제공하는 구분된 파워 상태 중 주어진 수행의 요구 조건에 맞

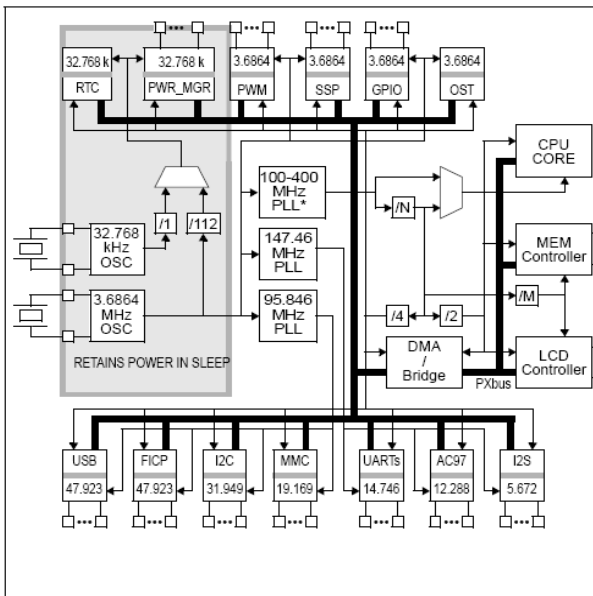
추어 가장 적절한 파워 상태로 전이함으로써 전력의 소모를 줄이는 기법들을 말한다. 시스템을 구성하는 다양한 장치들의 특성에 따라서 구체적인 구현의 내용은 달라지나 기본적으로 해당 장치가 사용되지 않는 사용 패턴에 대한 정확한 예측이 필요하며 저전력 파워 상태로의 전이를 결정할 때에는 상태 전이에 따른 시간 및 에너지 부대비용을 적절히 고려해야 한다[4].

2.3 PXA255의 동적 주파수 관리방법

다음은 PXA255의 동적 주파수 관리방법이다. 동적 주파수 관리(Dynamic Frequency Scaling, DFS)는 처리해야 할 프로세싱에 따라서 동작 주파수를 가변적으로 변동시키는 방법이다.

2.3.1 PXA255 Clock Manager

PXA255의 클럭 입력은 3.6864MHz 크리스탈이 PXTAL과 PEXTAL 핀에 연결되고, 32.768KHz 크리스탈이 TXTAL과 TEXTAL 핀에 연결된다. 이들을 위해 오실레이터들과 PLL들이 PXA255 내부에 집합되어 있어 클럭을 만들어주기 위한 크리스탈 외의 어떤 외부 디바이스도 필요하지 않다.



(그림 2-1) PXA255 Manager Block Diagram

2.3.2 Core Phase Locked Loop(Core PLL)

Core PLL은 CPU코어, 메모리 컨트롤러, LCD 컨트롤러, DMA 컨트롤러로 클럭을 제공한다. 이들은 다음 변수들로 증폭하며, 위 블록 다이어그램의

L, M, N은 다음과 같다.

표 2-1 주파수 배율

L 메모리 주파수 배수	[메모리 클럭 주파수 = 크리스탈 주파수(3.6864MHz) × L]로 메모리 주파수가 결정되며, L은 27, 36, 45의 값을 가질 수 있다.
M RUN 모드 주파수 배수	[RUN 모드 주파수 = 메모리 주파수 × M]으로 RUN 모드 주파수가 결정되며, M은 1 또는 2 를 가질 수 있다.
N TURBO 모드 주파수 배수	[TURBO 모드 주파수 = RUN 모드 주파수 × N] 으로 TURBO 모드 주파수가 결정되며, N은 1.0, 1.5, 2.0, 3.0을 가질 수 있다.

표 2-1에서는 3.6864 MHz에 대한 L, M, N 값들에 따른 Core PLL의 출력을 나타내며, 이 출력 주파수는 TURBO 모드일 경우 최대 400MHz이다.

2.4 Clock 설정관련 레지스터들

2.4.1 Core Clock Configuration Register(CCCR)

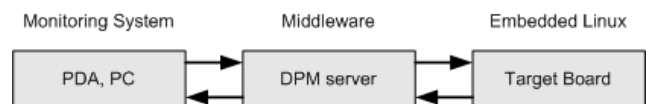
CCCR레지스터를 통하여 Core Clock 주파수에 관련된 메모리 주파수 배수(L), RUN 모드 주파수 배수(M), TURBO 모드 주파수 배수(N)가 설정된다.

2.4.2 Coprocessor 14의 CCLKCFG 레지스터

CCLKCFG 레지스트(CP14 Register 6)는 TURBO 모드와 주파수 변환 시퀀스로 들어가기 위해 사용된다. 주파수 변환 시퀀스로 들어갈 때 Turbo Bit이 변하지 않게 하려면 Read-Modify-Write 방식 즉, 이 레지스터를 읽은 후 해당 비트만을 고치고 다시 쓰는 단계를 사용해야 한다.

3. DPM 미들웨어 설계 및 구현

3.1 시스템 구조 설계



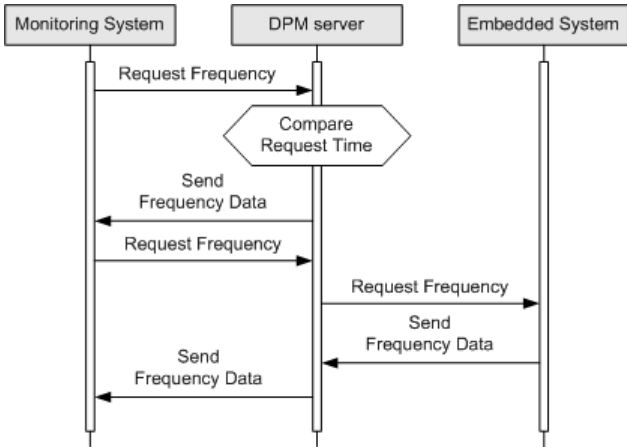
(그림 3-1) 시스템 구조

본 시스템의 구성요소 중 모니터링 시스템은 전력량을 수시로 체크하고 원하는 주파수 조절 파라미터 값을 DPM server에 전송하여 타겟보드의 전력

조절이 가능하도록 한다. 그에 해당하는 장비는 일반적으로 쓰는 PDA, PC, 노트북 등이 된다.

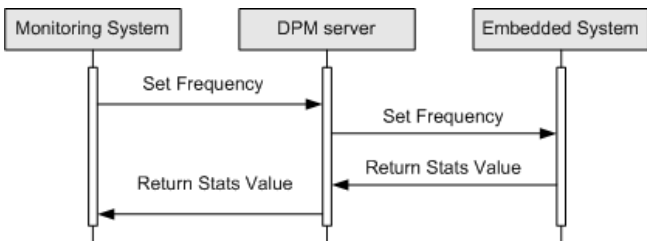
3.2 데이터 상태도

모니터링 시스템에서 DPM server 에게 특정 타겟보드의 주파수를 요청하게 되면 타겟 시스템의 모든 정보를 가지고 있는 DPM server 에서는 타겟보드에게 주파수 데이터를 요청한다. 타겟보드는 현재 자신의 CCCR을 읽어 DPM server에게 전송하며 이것을 받은 DPM server 는 모니터링 시스템으로 주파수 정보를 전송한다.



(그림 3-2) 타겟보드의 주파수 정보 요청 시 데이터 흐름

반대로 주파수 셋팅 시 모니터링 시스템에서는 타겟시스템의 번호와 주파수 값을 전송한다. 이를 전송받은 DPM server 는 타겟시스템에 장비번호와 주파수를 보내어 셋팅을 한다. 다음 그림은 타겟보드의 주파수 셋팅 시 프로토콜 흐름이다.



(그림 3-3) 타겟보드의 주파수 셋팅 시 데이터 흐름

3.3 DPM 디바이스 드라이버

디바이스 드라이버에는 DPM을 하기 위한 모듈이 구현되어 있다.

다음은 주파수 스케일링을 하기위한 디바이스 드라이버 코드 중의 ioctl 함수로 CCCR 레지스터를 제어하는 부분이다.

```

1 __asm__ __volatile__(
2   ldr r0, =0x41300000 @ PXA_REG_CCCR
3   ldr r1, =0x241      @ 1001000001
4   mov r0, #(1 << 1)
5   mcr p14, 0, r0, c6, c0, 0
6   and r1, r1, #(0x7 << 7)
7   cmp r1, #(0x2 << 7)
8   beq 10f
9   mov r0, #(1 << 0)
10  mcr p14, 0, r0, c6, c0, 0
11  10:
12  :
13  :
14  : "r0");
    
```

2, 3번 라인은 CCCR 레지스터를 설정하여 CPU 주파수 변경을 한다. 4, 5번 라인은 동작속도를 변경하기 위해서 코프로세서 레지스터를 변경한다. 6, 7, 8 번 라인은 터보모드가 있는지 확인을 한다. 9, 10 번 라인은 터보모드가 있을 경우 터보모드를 활성화시킨다.

4. 실험 및 검토

4.1 DPM을 위한 실험환경

표 4-1은 DPM 서버 스펙이다.

표 4-1 DPM server 스펙

MODEL	Samsung Magic Station m2761
OS	Linux 2.2.17
Processor	Intel Pentium III 800MHz
Memory	448 MB

표 4-2는 임베디드 시스템 스펙이다.

표 4-2 Embedded System 스펙

MODEL	X-Hyper250B
OS	Embedded Linux 2.4.18
Processor	Intel PXA250 400MHz
Memory	32MB Flash ROM, 64MB SDRAM

4.2 성능 평가

표 4-3는 실험에 사용된 동영상 파일의 스펙이다.

표 4-3 실험용 파일 스펙

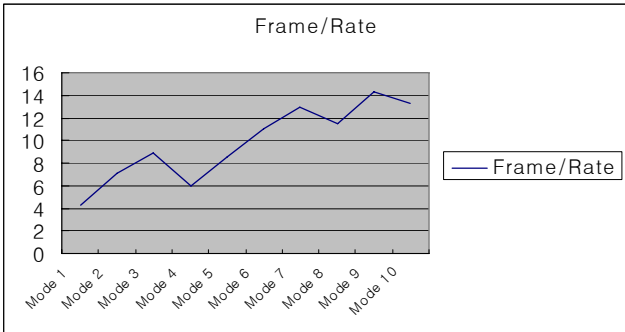
File Size	3.63MB
Average Data Rate	170.19KB per second
Image Size	352 × 240
Frame Rate	29.97 fps

각 모드별 주파수는 표 4-4와 같다.

표 4-4 모드별 주파수

Mode	1	2	3	4	5	6	7	8	9	10
Freq (Mhz)	99.5	199.1	298.6	132.7	199.1	298.6	398.1	265.4	331.8	398.1

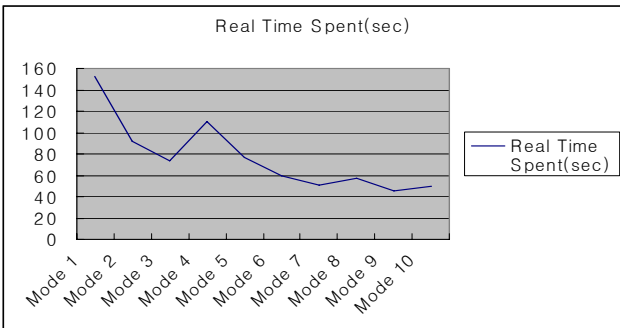
위 파일을 각 CPU 주파수 스케일링 별로 실행했을 때의 Frame rate는 그림 4-2와 같다.



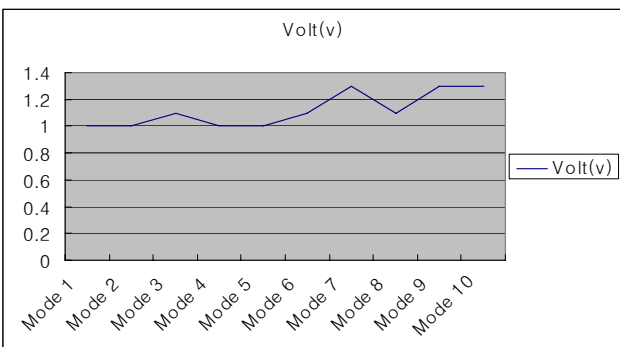
(그림 4-1) 주파수 스케일링 별 프레임 레이트

그림 4-1을 보면 동영상 파일을 모드1일 인 클럭 주파수 99.5MHz 로 실행시킨 프레임 레이트가 모드 10의 398.1MHz에 비해서 프레임 레이트가 낮음을 알 수 있다.

각 주파수 소비시간은 그림 4-3과 같다.



(그림 4-2) 주파수 스케일링 별 소비시간



(그림 4-3) 주파수 모드 별 소비전압

그림 4-2에서 동영상 파일이 모드 1일 경우에 프로세싱 시간은 152초인 반면 모드 10일 경우의 프로세싱 시간은 49초 밖에 소요되지 않음을 알 수 있다.

그림 4-3은 각 주파수 모드 별 전압을 나타내고 있다. 모드1일 경우에 1V의 전압이 걸리는 반면 모드10일 경우는 1.3V의 전압이 걸린다. 로우 클럭 상태에서의 동영상 처리보다는 하이 클럭 상태에서 처리가 더 빠르며 프레임 레이트도 더 높게 나타났지만 소모되는 전력의 양도 그에 비례한다는 것을 알 수 있다.

### 5. 결론 및 향후 연구방향

본 논문에서는 임베디드 시스템의 주변 기기들은 배제하고 마이크로프로세서에 초점을 두고 전력소모를 줄이기 위해서 DPM 미들웨어를 이용하여 동적 전력관리 시스템을 설계 및 구현에 대해서 연구하였다.

제안된 저전력 시스템 모델은 임베디드 리눅스 상에서 미들웨어 기반으로 마이크로프로세서를 제어함으로써 프로세서 스펙에 나와 있는 주파수로 변경을 하여 소비전력을 조금이나 줄이고자 감소 방안을 제안하였는데, 향후 과제로는 Win CE.net 과 uCOS 같은 RTOS 기반에서 DPM 미들웨어를 구현하고 보다 더 전력감소에 대해서 연구되어야 할 것이다.

### 참고문헌

[1] D. Grunwald, P. Levis, C. B. M. III, M. Neufeld, and K. I. Farkas, "Policies for Dynamic Clock Scheduling," in Proc. of the Fourth USENIX Symposium on Operating Systems Design and Implementation, Oct. 2000, pp. 73-86.

[2] A.Varma, B. Ganesh, M. Sen, S. R. Choudhury, L. Srinivasan, and J. Bruce, "A Control-Theoretic Approach to Dynamic Voltage Scheduling," in Proc. of International Conference on Compilers, Architectures and Synthesis for Embedded Systems, July 2001, pp. 255-266.

[3] K. Flautner, S. Reinhardt, and T. Mudge, "Automatic Performance-Setting for Dynamic Voltage Scaling," in Proc. of 7th Conference on Mobile Computing and Networking, July 2001.

[4] 성경모 외 "임베디드S/W: 국제경쟁력 분석을 통한 산업육성방안", KISTI, 기술산업분석, 2004.