

## cnu\_RTOS를 위한 저전력 기능의 구현

오승택<sup>○</sup>, 고영관<sup>\*</sup>, 이철훈<sup>\*</sup>

<sup>○\*</sup>충남대학교 컴퓨터공학과

e-mail: ost2011@cnu.ac.kr, Yg1985@cnu.ac.kr, clee@cnu.ac.kr

## Implementation of Low Power Function for cnu\_RTOS

Seung-Take Oh<sup>○</sup>, Young-Kwan Ko<sup>\*</sup>, Cheol-Hun Lee<sup>\*</sup>

<sup>○\*</sup>Chung-Nam National University

### ● 요약 ●

최근 Web 서비스, MP3, 동영상 재생, 무선통신 등 다양한 기능을 제공하는 스마트폰, 태블릿 PC와 같이 배터리로 동작하는 디지털 컨버전스 기기들의 사용량이 증가되었고, 이러한 이동형 임베디드 기기들은 배터리 용량에 따라 사용시간이 제한된다. 그렇기 때문에 기기들의 평가항목 중 전력소모라는 성능지표가 대두되고 있으며, 소비전력을 낮추기 위한 저전력 기법이 전 세계적으로 연구되고 있다. 본 논문에서는 이동형 임베디드 기기에 사용 가능한 cnu\_RTOS에 DPM(Dynamic Power Management)과 S3C2450에서 제공하는 DPM(Device Power Management)을 이용한 저전력 기능을 구현하여, 소비전력 감소율을 측정하였다.

**키워드:** 저전력, 실시간 운영체제, 내장형 시스템

## I. 서론

실시간 운영체제는 범용 운영체제와 달리 하드웨어 자원에 대한 공평한 분배를 목적으로 하기보다는, 시스템의 내·외부에서 발생하는 이벤트에 대하여 미리 정해진 마감시간(Dead Line)내에 처리하는 것을 목적으로 하는 운영체제이다. 이러한 실시간 운영체제는 원자력 발전 제어 시스템, 인공위성, 미사일과 같이 규모가 큰 내장형 시스템에 주로 사용 되어져 왔다. 하지만 웹 서비스, MP3, 동영상 재생, 무선통신 등 다양한 기능을 제공하는 스마트폰, 태블릿 PC와 같은 디지털 컨버전스 기기들이 등장함에 따라 실시간 운영체제가 이동형 임베디드 시스템에도 많이 사용되기 시작하였다. 이러한 이동형 임베디드 시스템들은 쾌적한 서비스 환경을 제공하기 위해 CPU의 클럭 증가, 메모리 사이즈 증가, LCD 크기의 증가와 같은 하드웨어적 성능의 향상과 기기의 구성품이 증가되는 추세이다. 이로 인해 단말기에서 소비하는 전력이 증가되고 있다. 그러나 소비전력의 증가에 따른 배터리 용량 증가는 매우 더딘 상황이며, 이를 해결하기 위해 최근 들어 디지털 시스템 전 분야에 걸쳐 소비전력을 줄이기 위한 방법이 활발히 연구되고 있다. 이러한 저전력 기술은 적은 전력 소모를 통해 실제 제품의 사용시간을 늘리고, 발열 문제를 완화 시킴으로 시스템 동작의 안정성을 향상시키게 된다. 따라서 이동형 임베디드 기기를 관리하는 시스템에서는 저전력 기능의 탑재가 필수화 되고 있으며, 이동형 내장형 시스템이 아니더라도 전자 산업 전반에 걸쳐 저전력 기

술에 대한 중요성이 높아지고 있다. 본 논문에서는 저전력 기능을 구현하기 위해 동적 전력 관리 기법, 장치 전력 관리 기법, 동적 전압 조절 기법에 대해 조사 및 분석하여 cnu\_RTOS에 적용하고, 소비 전류 감소비율을 측정하였다.

본 논문은 2장에서 관련연구로서 실시간 시스템, 실시간 운영체제 cnu\_RTOS와, 저전력 기법에 대해 소개하고, 3장에서는 동적 전력 관리 기법과, 장치 전력 관리 기법을 이용한 저전력 기능의 구현을, 4장에서는 테스트 환경과 결과를 보이며, 마지막으로 5장에서는 결론 및 향후 연구과제를 기술한다.

## II. 관련 연구

### 1. 실시간 시스템

실시간 시스템은 시스템 내·외부에서 발생하는 이벤트에 대하여, 이벤트 발생 시간과 그 이벤트 처리가 끝날 때까지의 지연 시간이 미리 제시된 마감시간을 넘지않는 시스템을 의미하며, 이벤트 발생과 처리가 실시간으로 이루어지는 특징을 가진다. 또한 기존의 일반 컴퓨터 시스템과는 달리 논리적 정확성뿐만 아니라 시간적 정확성이 중요시 되는 시스템이다[1]. 그리고 고속의 하드웨어를 사용해 무조건적으로 처리속도를 증가시키기 보다는, 이벤트에 대한 처리시간을 보장할 수 있도록 연구하는 분야이다.

## 2. 실시간 운영체제

이전 절에서 언급한 실시간 시스템에서 사용하는 운영체제를 실시간 운영체제라한다. 일반적인 컴퓨터에서 사용하는 범용 운영체제와의 큰 차이점은 태스크 스케줄링에 있어서 자원의 공평성 보다는 시간적 제약사항에 더욱 중점을 두고 스케줄링을 하며, 이러한 실시간 운영체제의 종류로는 멀티 쓰레드 모델과, 멀티프로세서 모델이 존재한다[2]. 멀티 쓰레드 모델의 경우 운영체제 커널과 응용프로그램이 합쳐진 형태로 구현하기 쉽고 빠르다는 장점이 있으나, 시스템 메모리를 공유하기 때문에 사소한 버그가 시스템 전체를 파괴할 수 있다는 단점이 존재한다. 반대로 멀티 프로세서 모델은 커널과 응용프로그램이 독립적으로 동작하게 설계됨으로 구현하기는 어렵지만, 버그에 대해 안정적인 장점이 있다는 것이 특징이다.

### 2.1 cnu\_RTOS

cnu\_RTOS는 ARM9 기반의 플랫폼에서 동작하는 실시간 운영체제로서 멀티태스킹을 지원하며, 우선순위 기반의 선점형 실시간 운영체제로 필요에 따라 64, 256, 512의 우선순위를 제공한다. 또한 태스크간의 통신 및 동기화를 위해 세마포, 메시지큐, 메시지 메일박스 등을 제공하며, 정적, 동적 메모리 관리기법을 제공한다. 또한 LCD의 활용 여부에 따라 소비전류를 감소시키는 DPM(Dynamic Power Management) 정책을 적용하고 있다.

## 3. 저전력 기법

### 3.1 동적 전압 조절(Dynamic Voltage Scaling)

동적 전압 조절기법은 실시간 시스템에서 요구되는 종료시한이라는 제약조건을 만족시키는 범위 내에서 CPU의 구동전압을 조절하는 기법으로, 실시간 태스크들이 마감시간 전에 수행이 완료된다는 점을 기반으로 전력감소를 위한 유희시간 계산을 통해 실시간성을 유지하게 된다[3].

### 3.2 장치 전력 관리(Device Power Management)

장치 전력 관리기법은 시스템의 성능 감소 없이 전력 소비를 줄이는 효과적인 방법으로 서비스 요구가 없는 디바이스는 유희모드 (Idle)로 동작하다가 태스크로부터 디바이스 사용 요청이 들어왔을 때 요구한 디바이스를 가동시키는 저전력 기법이다.

### 3.3 동적 전력 관리(Dynamic Power Management)

IBM 저전력 연구센터와 리눅스 기술센터, 몬타비스타 소프트웨어사에서 제안한 동적 전력 관리기법은 일반적인 연성 실시간 시스템과 비실시간 시스템을 가정하여 리눅스 상에 구현되어있다. 동적 전력관리는 CPU의 구동속도와 시스템 버스 클럭, 그에 따른 전압을 조절하여, 장치 동작상태를 제어하여 소모전류를 감소시키는 동적 전압조절과 장치전력관리 기법이 융합된 전력관리 기법으로, 휴대용 PC에 가장 효율적인 전력 관리 기법이다[4].

## III. 본 론

본 논문은 IBM 저전력 연구센터에서 개발한 동적 전압 관리 기법을 태스크의 우선순위에 따라 적용시키고, S3C2450 칩셋에서 제공하는 장치 전력 관리를 이용해 저전력 모듈을 구현하였다.

사용자가 CreateTask() 함수를 이용해 태스크를 생성할 때, 각 태스크의 중요도에 따라 우선순위를 지정하게된다. 지정된 우선순위에 따라 저전력 정책이 시스템에 의해 설정되고 이후 각 태스크가 Tick Interrupt에 의해 스케줄링 될 때, CPU(Central Processing Unit) 점유권을 가지게 되는 태스크의 저전력 정책을 참조하여 PLL(Phase Locked Loop)의 출력 Frequency를 변경한다. PLL의 출력 Frequency가 변경됨으로써, Core에 유입되는 Frequency가 동적으로 조절되어 전체 소비전력을 감소시키게 된다. 아래의 [그림 1]은 이러한 동작과정을 구현한 소스코드 내용이다.

```

MK_Status_t MK_Change_Freq( MK_TASK* Next_task ){
    DPM_Struct update;
    MK_U32_t Flags;

    Flags = MK_InterruptDisable();

    if( Next_task->t_DPM_Status == High ){
        update.MPLL_Out = 600;
        update.MPLL_Value = ((300<<14)|(3<<5)|(1));
        update.Voltage = 5;
    }
    else if( Next_task->t_DPM_Status == Normal ){
        update.MPLL_Out = 450;
        update.MPLL_Value = ((225<<14)|(3<<5)|(1));
        update.Voltage = 4;
    }
    else if( Next_task->t_DPM_Status == Low ){
        update.MPLL_Out = 240;
        update.MPLL_Value = ((320<<14)|(4<<5)|(2));
        update.Voltage = 3;
    }

    rMPLLCON = update.MPLL_Value;

    __asm(
        nop;
        nop;
        nop;
        nop;
        nop;
        nop;
        nop;
    );

    MK_InterruptRestore(Flags);

    return MK_NO_ERROR;
} ? end MK_Change_Freq ?
    
```

[그림 1]. Frequency 동적변경 코드

Fig. 1. Dynamic change Source Code for Frequency

그림에서 볼 수 있듯이 Frequency를 변경한 뒤, nop 명령어를 7 Cycle 수행해야 되는데, 이는 Frequency가 변경된 뒤, 바로 연산을 수행하게 되면 불안정한 Frequency에 의해 CPU가 오작동될 수 있기 때문이다.

아래의 [표 1]은 본 논문에서 구현한 저전력 정책을 정리한 표이다.

표 1. 태스크 우선순위에 따른 구동 정책  
Table 1. Working Policy with Task priority

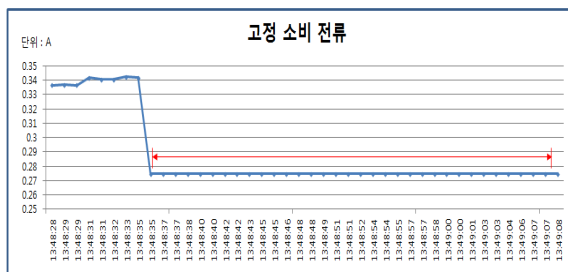
구동정책	공급주파수 (PLL)	코어주파수 (ARMCLK)	장치 전력 관리 기법
Crypto Work ( 0 ~ 64 )	800 MHz	400 MHz	Normal
Crypto Half-Work ( 65 ~ 128 )	400~600 MHz	200~300 MHz	Normal
Crypto Not-Work ( 129 ~ 255 )	240 MHz	120 MHz	Normal
Idle	240 MHz	0 MHz	Idle

[표 1]에 정의된 정책에 따라, 저전력 모듈이 수행되며 또한 Crypto Half-Work 정책에서는 사용자가 코어에 입력되는 Frequency와 PLL에서의 출력 Frequency를 설정할 수 있도록 하였다. 또한, 태스크들이 세마포어의 획득을 위해 실제 Ready 상태인 태스크가 없을 경우에는 Idle 태스크가 동작을 한다. 이때는 S3C2450에서 제공하는 장치 전력 관리 기법을 활용하여, 실제 Core의 구동 Frequency를 0 MHz로 만듦으로써 소비전류를 최소화 시키게 된다.

#### IV. 실험 환경 및 결과

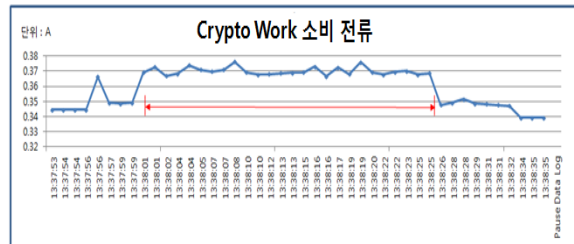
본 논문에서 구현한 저전력 기법의 성능평가를 위한 실험환경 구성은 `gnu_RTOS`를 `SMDK2450`보드에 포팅한 후 저전력 기능을 포함한 태스크들과 포함하지 않는 태스크들을 생성하여 실제 어느정도의 소비전류양이 감소되는지를 측정하였다.

해당 감소치를 정확히 확인하기 위해 Core를 제외한 모든 주변 장치를 정지 시킨 뒤 기본적으로 보드에서 사용되는 소비전류값을 계산한 뒤 그 감소량을 측정하였으며, 실제 감소되는 소비전류 값을 구하기 위해 보드자체적으로 사용되는 고정 소비전류를 측정한 결과는 아래의 [그림 2]와 같다.

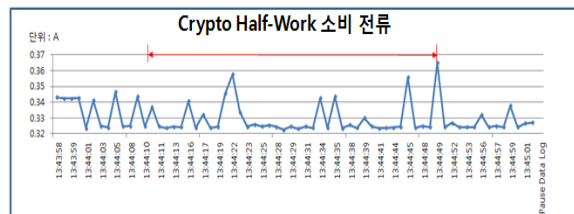


[그림 2]. 보드의 고정 소비 전류 측정 결과  
Fig. 2. Static Power consume result

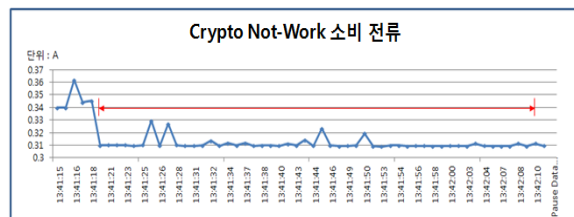
아래의 [그림 3]는 Crypto Work 정책에서의 소비전류를 측정 한 결과이며, [그림 4], [그림 5]는 각각 Crypto Half-Work, Crypto Not-Work로 동작 했을 경우 각각의 소비 전류를 측정 한 결과이다. 이 결과를 종합하여 실제 감소된 전류의 비율과 측정치를 기록한 것이 [표 2]와 같다. [표 2]의 결과 내용은 각 정책에서의 소비 전류와 기본 소비 전류의 차를 구한 뒤, Crypto Work 즉, 저전력 기법을 사용하지 않았을 경우에 대비 어느정도 감소했는지 백분율로 나타내었다.



[그림 3]. Crypto Work 정책에 따른 소비전류 측정  
Fig. 3. Result of consume current as Crypto Work



[그림 4]. Crypto Half-Work 정책에 따른 소비 전류 측정  
Fig. 4. Result of consume current as Crypto Half-Work



[그림 5]. Crypto Not-Work 정책에 따른 소비전류 측정  
Fig. 5. Result of consume current as Crypto Not-Work

표 2. 정책에 따른 소비전류 감소 비율  
Table 2. Result of consume current rate with policy

	기본 소비 전류	Crypto Work	Crypto Half-Work	Crypto Not-Work
평균 소모 전류	0,2750A	0,3699A	0,3299A	0,3111A
(평균 소비 전류 - 기본 소비 전류)	-	0,0949A (A)	0,0549A (B)	0,0361A (C)
감소율	-	-	42% (A-B)/A * 100	62% (A-C)/A * 100

[표 2]에서 볼 수 있듯이, 최대 62% 까지 소비전류를 감소시킬 수 있다. 소비 전류가 낮아짐에 따라 소비전력도 감소되는데 소비전력은 그래프에서의 면적으로 소비전류의 감소량에 비례하게 된다.

## V. 결론

최근 무선 통신과 컴퓨팅 기술의 발달로 인해 휴대용 PC에 E-Mail, 웹 검색 등 다양한 서비스들이 제공되고 있으며, 동영상 재생과 같은 멀티미디어 서비스를 제공하기 위해 하드웨어 성능이 높아지고, 기기의 구성품들이 다양해 지고 있다. 이에 따라 기기에서 사용되는 소비전력이 높아져 실제 기기를 사용할 수 있는 시간이 더욱 제한된다는 문제점이 발생하였다. 이러한 문제점을 해결하기 위해 휴대용 PC에 사용되는 운영체제에는 저전력 기법이 필수로 되었으며, 본 논문에서는 실시간 운영체제인 `gnu_RTOS`에 동적 전력 기법과 장치 전력 관리기법을 통해 소비전류를 동적으로 낮춤으로써, 최대 62%의 감축 효과를 확인하였다. 향후 연구 과제로는 개발보드에서 제공하는 각 주변장치들에 대해 선택적으로

구동시키는 장치 전력 관리기법을 적용하여, 사용자가 필요에 의해 소비 전력을 더욱 감소시킬 수 있도록 구현하는 것이다.

## 참고문헌

- [1] JANE W.S. LIU, "REAL-TIME SYSTEMS" Prentice Hall pp.34-84, 2000
- [2] SILBERSCHATS and GALVIN, "Operating System Concepts" Hong-Reung, pp53-91, 1999
- [3] MunHaeng Jo, JaeSuk Lee, Cheol-hoon Lee, "eRTOS: The Low-Power Real-Time Operating System for wearable Computers", International Symposium on Consumer Electronics(ISCE2009), Japan, pp1015-1019
- [4] MunHaeng Jo, MyungJo Jung, YoungHee Kim, Cheol-hoon Lee, "Low-Power technique base on timeout policy for portable systems", International Conference on Information Technology and Applications(ICITA, 2008),Australia, pp.619-623