
타이젠 기반 스마트폰 파워 매니저의 현재 LCD 상태에 새로운 상태 추가를 통한 에너지 절약 기법

이상준* · 권영호* · 이병호**

*한양대학교

Mechanism for Energy Conservation by Adding New State to the Current LCD States of the Power Manager of Smartphones Based on Tizen

Sang-Jun Lee* · Young-Ho Kwon* · Byung-Ho Rhee**

*Hanyang University

E-mail : gonbangwa9@naver.com

요 약

모바일 운영체제는 대표적으로 애플과 안드로이드 운영체제로 구분되어 왔다. 삼성은 리눅스 커널을 기반으로 하는 자체 OS 인 Tizen을 만들어서 새로운 모바일 운영체제를 선보였다. 모바일 특성상 배터리 용량의 제한 때문에 모바일 운영체제는 자체적으로 저전력을 사용하는 기술을 발전시켜 왔다. 삼성 Tizen OS는 사용자 입력이나 타임 아웃 이벤트가 발생함에 따라 LCD 상태를 조절하는 파워매니저라는 저전력 기술을 갖고 있다. 하지만 사용자 입력이 빈번할수록 LCD 상태 변경에 따른 오버헤드가 증가해 CPU 클럭이 순간적으로 증가하여 사용자 입력 전후에 에너지 소모가 급증하는 단점이 있다. 본 논문에서는 기존 Tizen OS에서 사용하는 파워매니저에 현재 LCD 상태들의 중간 상태를 추가하여, 사용자 입력이 빈번할 때 LCD 상태 변경에 따른 오버헤드를 줄이는 기법을 제안한다. 본 논문에서 제안하는 기법을 Tizen 폰 커널단에 구현하고, 사용자 LCD 터치 입력에 대해 실험하여, 사용자의 빈번한 입력에 따른 CPU클럭 증가가 감소하여 에너지를 절감할 수 있음을 보였다.

ABSTRACT

Mobile operating systems have been typically classified into Apple and Android. Samsung showed its own new mobile OS developing Tizen based on Linux kernel. Mobile operating system has developed a technology using low-power by itself because of the limitation of capacity of battery, a feature of mobile. Samsung Tizen OS has a low-power technology called Power Manager controlling LCD states as users' inputs or time-out events occur. However, if users' input occurs frequently, energy consumption jumped before-and-after users' input because CPU clock is increased rapidly due to overhead increase for frequent LCD state changes. This paper proposes a mechanism to reduce the overhead for LCD state changes, when user's input is frequent, by adding a new state to the Power Manager the current Tizen OS is using. We have implemented the proposed mechanism at Tizen phone kernel layer in this paper and experimented the mechanism according to users' LCD touch inputs. The experiment shows that it is possible to decrease energy by reducing the CPU clock increase according to the frequent user's inputs.

키워드

Tizen, 파워 매니저, 상태 변화, 저전력

I. 서 론

스마트폰이 대중화 되면서, 스마트폰은 경제, 사회 등 모든 분야에 걸쳐 일상생활에서 반드시 필요한 기기가 되었다. 스마트폰의 전 세계적인 확산에 맞춰 우리나라의 스마트폰 보급률은 지난 3월 기준으로 83.0%에 달해 아랍에미리트(UAE · 90.8%), 싱가포르(87.7%), 사우디아라비아(86.1%)에 이어 세계 4위로 나타날 정도로 대중화된 스마트폰이 일상에 미치는 영향이 매우 컸다[1].

이에 따라, 사용자들의 스마트폰 성능에 대한 기대치가 높아지고 있다. 높아진 기대치를 충족시키기 위해 스마트폰은 나날이 고성능 기기들을 장착하고 있다. 하지만 모바일 기기가 갖고 있는 이동성 때문에 배터리 용량은 제한이 되는 상황이다. 고성능 기기들로 인해 소비 전력이 늘어나는 반면, 배터리 용량은 제한되기 때문에 스마트폰의 에너지 효율성을 높이는 저전력 기술이 매우 중요해지고 있다. 전력 소모를 줄이기 위해 iOS, 안드로이드, Tizen 등 다양한 스마트폰 OS들은 저전력 기법들을 제공하고 있다.

스마트폰 OS 기기들 중에 Tizen 기반 스마트폰은 전력 소모를 줄이기 위해서 Power Manager라는 저전력 기법을 사용한다. Tizen OS가 제공하는 Power Manager는 5가지의 default 전력 상태를 갖는다. Power Manager는 스마트폰의 LCD 상태와 전력 관련 system operation 상태를 제어함으로써 전력을 관리한다. 그림 1은 Power Manager의 5가지 default 전력 상태, 전력 상태들 간의 전이와 전이 조건들을 나타낸다. 전력 상태 전이 과정은 호출 검사, 상태 전이, action 함수 진입, 이 3가지 단계에 걸쳐서 진행된다. 호출 검사 단계에서는 다음 상태로의 전이가 가능한지를 검사한다. 가능할 경우, 발생한 event에 따라서 현재 상태를 다음 상태로 변경한다. 표 1은 default 전력 상태들이 Timeout과 Input event가 발생했을 때 전이될 다음 상태를 나타낸 Transition table을 보여준다. 상태 전이 후에 각 상태에 맞는 적절한 작업을 수행할 action 함수가 호출된다. NORMAL 상태에서는 backlight on 또는 이 전 밝기를 회복시키는 작업을 수행한다. LCDDIM 상태에서는 밝기를 어둡게 조절한다. LCDOFF 상태에서는 backlight off 작업을 수행한다. 이와 같이, Tizen OS의 Power Manager는 system operation 상태만이 아니라, 스마트폰의 LCD 상태도 적절하게 변경시키면서 전력을 관리한다[2,3].

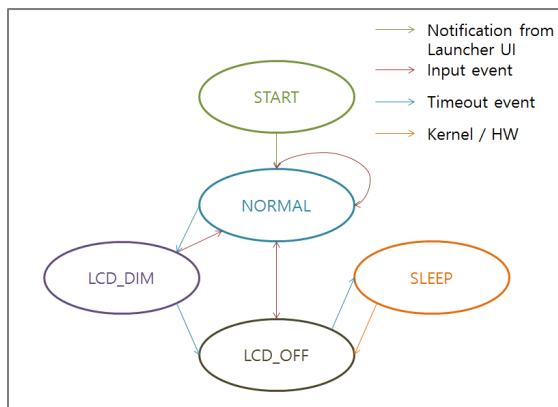


그림 1. 기존 Power Manager의 전력 상태 변화도

표 1. Tizen Power Manager의 Transition table

State	Timeout	Input
START	START	START
NORMAL	LCDDIM	NORMAL
LCDDIM	LCDOFF	NORMAL
LCDOFF	SLEEP	NORMAL
SLEEP	LCDOFF	NORMAL

본 논문에서는 스마트폰의 LCD 밝기를 조절해서 전력 소비의 효율성을 높이는 기법을 제안하므로 전체 전력 상태 변화가 아닌, 그림 1의 상태 변화도 중에서 NORMAL, LCD_DIM, LCD_OFF 상태들이 포함된 부분에만 해당하는 LCD 상태 변화에 초점을 둔다. 본 논문에서는 기존의 Tizen OS가 기기의 LCD 상태를 변경하는 과정을 수정해 전력 소비의 효율성을 높이는 기법을 제안한다.

II. 기존 Tizen 저전력 기법의 한계점

사용자가 뉴스 기사 또는 SNS에 올라 온 소식 등을 읽거나 사진첩의 사진을 보는 행위를 하는 경우에는 일정 시간이 지나 어두워진 기기의 밝기를 되돌리기 위해 주기적으로 Touch event가 발생한다. 이러한 경우에는 기존 Tizen OS의 Power Manager의 방식을 사용하는 스마트폰에서는 LCD 상태 변화로 인한 CPU 이용률의 순간적인 증가가 빈번히 발생하는데, 이번 연구에서는 위의 상황에서의 기존 Tizen Power Manager를 사용한 스마트폰의 CPU 이용률 변화를 측정, 분석해보았다.

분석은 Tizen 스마트폰이 기본으로 제공하는 사진첩 어플리케이션을 대상으로 진행하였다. 실험은 Tizen SDK의 tool인 Dynamic analyzer [4]를 사용하여 해당 어플리케이션을 실행 시키고 일정 주기로 사용자의 입력을 넣는 방법으로 진행하였다. 그림 2는 총 3분 동안 일어나는 CPU의 이용률의 변화를 1초마다 측정된 실험 결과를 그래프로 나타낸 것이다. 그래프의 실선은 CPU 이용률의 변화를 나타내며, 점은 사용자의 Touch event가 발생한 시점을 나타낸다. Touch event는 LCD 상태가 LCDDIM 상태로 변화할 때마다 그 즉시 발생한다. 그림 2의 그래프에서 볼 수 있듯이 Touch event가 발생한 직후마다 CPU 이용률이 급격하게 증가하는 현상을 보여준다.

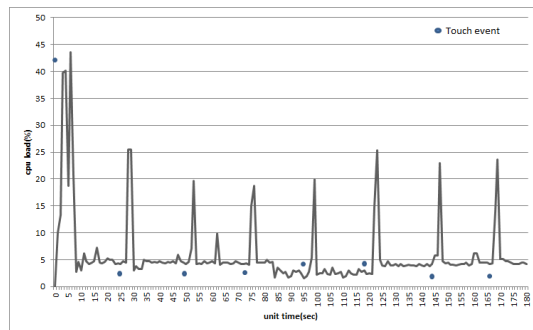


그림 2. 기존 Tizen 방식의 사용자 입력에 따른 CPU 이용률 변화

III. SEMIDIM 상태 추가를 통한 저전력 기법

앞서 분석한 CPU 이용률 증가폭을 바탕으로 기존 Tizen 스마트폰의 방식은 CPU 이용률의 증가폭이 크기 때문에 일정 주기로 사용자의 입력이 빈번하게 발생하면 그만큼 전력 소비가 커진다는 단점을 보였다. 본 논문에서는 그림 3과 같이 기존의 Tizen OS Power Manager의 상태 변화도에 추가적인 SEMIDIM 상태를 구현해 일정 주기로 사용자 입력이 빈번히 발생하는 경우에서의 전력 소비 효율성을 높이는 기법을 제안한다. SEMIDIM 상태는 NORMAL 상태와 LCDDIM 상태의 중간 상태를 의미한다. SEMIDIM 상태에서는 LCD의 밝기를 NORMAL 상태일 때의 밝기의 절반 정도로 설정함으로써 기존의 LCDDIM 상태의 backlight 밝기보다 밝게 설정한다. 반면, LCD의 상태가 LCDOFF 상태로 될 때까지의 시간이 길면 오히려 소비 전력이 늘어날 수 있기 때문에 SEMIDIM 상태의 지속시간은 기존의 LCDDIM 상태 지속 시간보다 짧게 설정한다. 제안된 기법에서는 표 2의 제안된 기법의 Transition table에서 볼 수 있듯이 NORMAL 상태에서 Timeout event가 발생하면 SEMIDIM 상태로 상태 전이가 일어나도록 설정한다. 또한 SEMIDIM 상태에서 Timeout event가 발생하면 LCDDIM 상태로, Input event가 발생하면 NORMAL 상태로 전이하도록 설정한다. NORMAL 상태였던 기기의 LCD 밝기는, Timeout event가 발생해 SEMIDIM 상태로 바뀌면서, 줄어들지만 LCDDIM 상태의 밝기보다는 밝게 유지된다. SEMIDIM 상태에서 사용자의 입력이 바로 발생할 때의 CPU 이용률은 기존의 LCDDIM 상태에서 NORMAL 상태로 바뀔 때의 CPU 이용률보다 낮아 에너지 사용량을 줄이는 효과를 보일 수 있다.

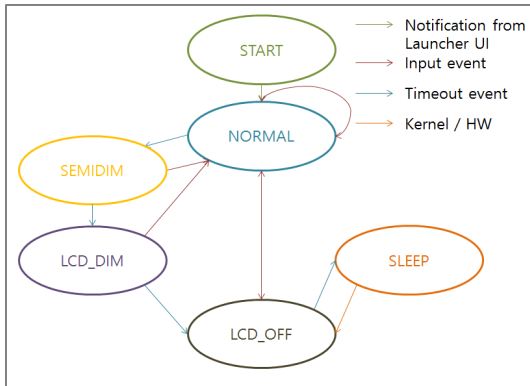


그림 3. 제안된 Power Manager의 상태 변화도

표 2. 제안된 기법에 따른 새로운 Transition table

State	Timeout	Input
START	START	START
NORMAL	SEMIDIM	NORMAL
SEMIDIM	LCDDIM	NORMAL
LCDDIM	LCDOFF	NORMAL
LCDOFF	SLEEP	NORMAL
SLEEP	LCDOFF	NORMAL

IV. 평가 및 분석

실험 장비로는 Tizen 레퍼런스 폰인 Tizen_Ref. Device-PQ 2대를 사용하였다. 에너지 소비량 측정을 위하여 Tizen SDK의 tool인 Dynamic Analyzer를 사용하였으며, 테스트 어플리케이션으로는 Tizen 스마트폰에 기본으로 제공되는 사진첩 어플리케이션을 사용하였다. 해당 어플리케이션을 실행 시킨 후 3분 동안 화면을 보면서 도중에 LCD가 어두워지면 즉시 터치 입력을 가해 LCD 밝기를 원상태로 되돌리는 방식으로 실험을 진행하였다.

첫 번째로 앞서 분석한 기존의 Tizen OS가 제공하는 LCD 상태 변화 기법에서의 CPU 이용률의 증가폭과 제안한 기법을 이용해 새롭게 CPU 이용률을 측정하여 비교하였다. 분석은 이전 실험에서 기존의 방식을 이용했을 때의 CPU 이용률 변화 분석 방법과 동일하게 진행하였다. 그림 4은 기존의 방식과 제안된 방식을 이용했을 때 사용자의 입력에 따른 CPU 이용률 증가폭에 대한 결과를 그래프로 비교한 것이다. 진한 실선 그래프는 기존의 방식을 사용했을 때 CPU 이용률의 변화를 나타낸 그림 2의 그래프와 동일하다. 옅은 실선 그래프는 제안된 기법을 사용한 경우의 CPU 이용률의 변화를 나타낸다. 그림 4의 두 그래프를 비교해서 확인할 수 있듯이 제안된 기법을 사용했을 때의 CPU 이용률의 증가폭이 기존의 방식을 사용했을 때의 CPU 이용률의 증가폭보다 좁다는 것을 알 수 있다.

두 번째로 기존의 방식과 제안된 기법을 이용해 사용자가 사진첩 어플리케이션을 사용할 때의 에너지 사용량을 측정하였다. 그림 5, 6의 그래프들은 각 기법 별로 어플리케이션을 실행할 때의 총 3분 동안의 어플리케이션의 에너지 사용량을 1초 마다 나타낸다. 그림 5의 그래프에서 확인할 수 있듯이 기존 방식을 사용하는 경우에는 사용자의 입력에 따라서 증가하는 CPU 이용률이 크기 때문에 사용자의 입력이 발생한 직후에 소비되는 에너지 사용량이 많아지는 것을 확인할 수 있다. 그림 6의 그래프를 통해서 확인할 수 있듯이 제안된 기법은 사용자의 입력이 발생해 LCD 상태가 NORMAL로 돌아갈 때 증가하는 CPU 이용률이 낮기 때문에 기존의 방식보다 에너지 사용량을 줄일 수 있다. 표 3에서 두 기법의 총 에너지 사용량을 비교해서 제안된 기법을 사용해 사용자의 입력이 주기적으로 빈번히 발생할 경우에 에너지 소비량을 줄일 수 있음을 확인할 수 있다.

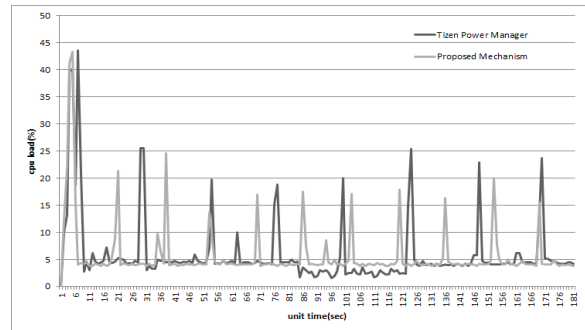


그림 4. 기존의 방식과 제안된 기법의 CPU 이용률 비교

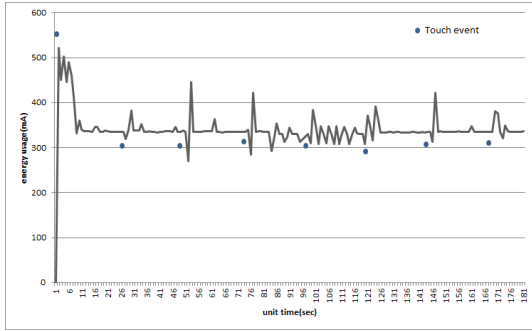


그림 5. 기존 Tizen Power Manager의 에너지 사용량

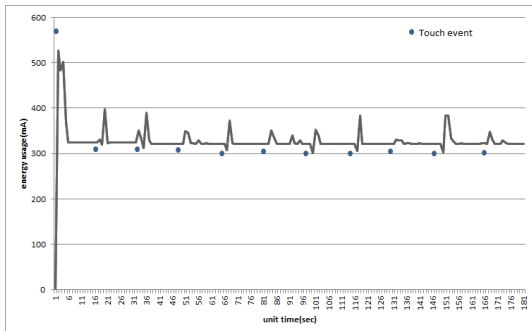


그림 6. 제안된 기법의 에너지 사용량

표 3. 총 에너지 사용량

기법	총 에너지 사용량(mA)
Tizen Power Manager	61504.8
Proposed Mechanism	59183.4

본 논문에서는 시뮬레이션을 통해서 제안한 기법을 검증한 결과 3분 동안 기존의 방식은 8번 정도의 상태 event가 발생하였고 제안된 기법에서는 11번 정도의 상태 event가 발생하였다. 이는 그림 2의 결과에서 볼 수 있듯이 NORMAL 상태에서 SEMIDIM 상태로의 변화가 일어날 때 까지 걸리는 시간이 기존 방식에서의 LCDDIM 상태로 변화할 때까지 걸리는 시간보다 짧아서 NORMAL에서 SEMIDIM으로의 상태 변화가 NORMAL에서 LCDDIM으로의 상태 변화보다 더욱 빈번히 일어나기 때문이다. 비록 제안된 기법이 시간에 비례해 상태 event 횟수를 증가하지만, 표 3에서 확인 할 수 있듯이 제안된 기법에 총 에너지 사용량이 기존 Tizen OS Power Manager 총 에너지 사용량보다 3.77% 감소한 것을 알 수 있다.

V. 결론 및 향후연구

본 논문에서는 기존에 Tizen 기반 스마트폰이 제공하는 LCD 상태 변화 알고리즘에 한 가지 상태를 추가해서 소비 전력을 줄일 수 있는 기법을 제안하였으며 이를 Tizen 기반의 스마트폰에 구현하였다. 그 결과 사용자의 입력이 빈번히 받

생하는 경우에는 제안된 기법을 사용하는 것이 기존의 Tizen 스마트폰의 방식과 비교하여 소비 전력을 줄일 수 있다는 것을 확인하였다.

향후 연구에서는 제안된 기법에 상태 변화 이벤트를 줄이기 위해서 사용자 입력과 프로그램 동작 상태와 연동해서 LCD 화면 밝기를 제어할 수 있는 기술에 대해서 연구하고 한다.

참고문헌

- [1] 현윤경 (2015). “한국, 스마트폰 보급률 83.0%…세계 4위”
<http://www.yonhapnews.co.kr/bulletin/2015/07/07/0200000000AKR20150707175600017.HTML?input=1195m> (검색일: 2015.08.02).
- [2] 함명주 (연도불명). “Tizen Kernel & System Frameworks for Vendors and Enthusiasts.”
<http://www.krnet.or.kr/board/data/dprogram/1784/C1-2-KRnet2013.pdf> (검색일: 2015.05.16.)
- [3] Darrenhjung (2012). “Power manager section of Tizen Porting Guide/System.”
https://wiki.tizen.org/wiki/Porting_Guide/System#Power_manager (검색일: 2015.05.15).
- [4] Tizen Developers (연도불명). “Tools in Tizen Developments.”
<https://developer.tizen.org/development/tools/native-tools/dynamic-analyzer> (검색일: 2015.05.25.).