Journal of The Institute of Electronics and Information Engineers Vol.54, NO.4, April 2017 https://doi.org/10.5573/ieie.2017.54.4.44
ISSN 2287-5026(Print) / ISSN 2288-159X(Online)

논문 2017-54-4-6

Cold Restart를 이용한 웨어러블 디바이스의 초저전력 핵심 기술

(Core Technology for Ultra Low Power Using Cold Restart in Wearable Devices)

김 선 태*, 박 형 준**, 박 호 준*, 우 덕 균*

(Seon-Tae Kim©, Hyoung Jun Park, Ho-Jun Park, and Duk-Kyun Woo)

요 약

헬스케어 및 의료분야를 위한 배터리 기반 웨어러블 디바이스가 많이 제품화되고 있으나, 효과적인 전력 관리 미흡으로 자주 충전해야 하는 사용자의 불편함이 존재한다. 본 연구에서는 HW에서 제공하는 전력 관리 회로도를 효과적으로 활용할 수 있는 Tickless 기반 운영체제 및 전력 관리 알고리즘을 제시하고, 보드 레벨에서 최소 전력을 소모하는 Cold Restart 기법을 제안한다. 제안된 기법의 운영체제는 기존 운영체제에 비해 웨어러블 디바이스의 응용을 모델링한 4가지 시나리오에서 2배에서 최대 33배의 전력 소모 절감을 가져왔다.

Abstract

There are many battery-based wearable devices for healthcare and medical applications, but there is a user's inconvenience to charge battery frequently due to insufficient power management. In this paper, we propose a tickless-based operating system and power management algorithm that can effectively utilize the power management provided by HW, and propose a cold restart method that consumes the minimum power at the board level. The operating system of the proposed technique has reduced the power consumption from 2 times to 33 times in the four scenarios modeling the wearable device application compared to the existing operating system.

Keywords : Cold Restart, Ultra Low Power, Operating System, nanoQplus-W, Wearable Devices

Ⅰ. 서 론

최근 휴대폰과 연동되는 스마트 워치, 피트니스 밴스 및 패치형 디바이스 등 웨어러블 디바이스가 헬스케어 및 의료 분야에 제품화되고 인기를 끌고 있다. 하지만, 센서에서 받은 정보를 수집/처리 및 전송하는 과정에서 많은 전력을 소모하여 사용자는 자주 충전을 해야 불편함을 겪게 된다. 이는 소형 디바이스의 공간 사이즈에 따른 배터리의 용량에 문제도 있지만, 보다 근본적으로는 응용에 따른 최적 전력 소모를 효과적으로 제어하지 못하는데 따른다.

지금까지 디바이스에서 소모되는 전력을 최소화하려는 연구는 다양한 영역에서 많은 방법들로 시도되어왔다. 크게는 HW(SoC) 영역과 SW(OS)영역에서 분리되어 시도되었다. 대체로 HW 영역에서는 응용의 성능에 따른 동작 주파수와 인가 전압을 조절하는 DVFS 방식, 응용의 활동 시간에 따른 전력모드를 동적으로 변경하는 DPM 방식과 에너지 절감을 위한 MCU내 캐시나 메모리 등의 마이크로 구조 변경 등이 있었다.[1~2]

하지만 HW적으로 지원되는 전력 관리 방식으로 실제 응용을 구동하고 제어하는 SW에서 무분별하게 적용

* 평생회원, 한국전자통신연구원(ETRI)
** 정회원, 한국기술연합대학원대학교(UST)
© Corresponding Author(E-mail : stkim10@etri.re.kr)
※ 이 논문은 산업통상자원부(MOTIE)의 산업현장핵심기술수시개발 및 산업기술혁신 사업의 지원을 받아 수행된 연구임.(No.10054668, '웨어러블 디바이스를 위한 경량 운영체제기반 초저전력 대기모드 핵심기술 개발 No.10056738, 다양한 응용서비스 적용이 가능한 개방형 웨어러블 디바이스 개발킷 핵심기술 개발)
Received ; November 10, 2016 Revised ; December 19, 2016
Accepted ; March 7, 2017

Copyright © The Institute of Electronics and Information Engineers.

This is an Open-Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (http://creativecommons.org/licenses/by-nc/3.0) which permits unrestricted non-commercial use, distribution and reproduction in any medium, provided the original work is properly cited.

된다면 별다른 절감 효과를 보지 못하고 만다. 따라서, HW 지원 전력관리 구조에 맞는 SW의 제어를 수행하는 OS-Directed 전력 관리 방식[3]이 연구되었으며, I/O 장치와 태스크의 수행 동작에 따른 동적 전력 관리 모듈을 제어하였다. 한편, 모바일 디바이스에서 소모되는 전력을 분석한 결과 메모리 접근 시에 전력 소모가 많은 것을 찾아내고 메모리 전력 관리를 제안[4]하기도 하였다.

최근에는 웨어러블 디바이스, 특히 헬스케어나 의료분야에서 경량 디바이스들이 많이 사용되고 배터리 장시간 사용 문제가 이슈로 부각됨에 따라, 이에 대한 분석 연구[5~6]들이 많이 추진되고 있다.

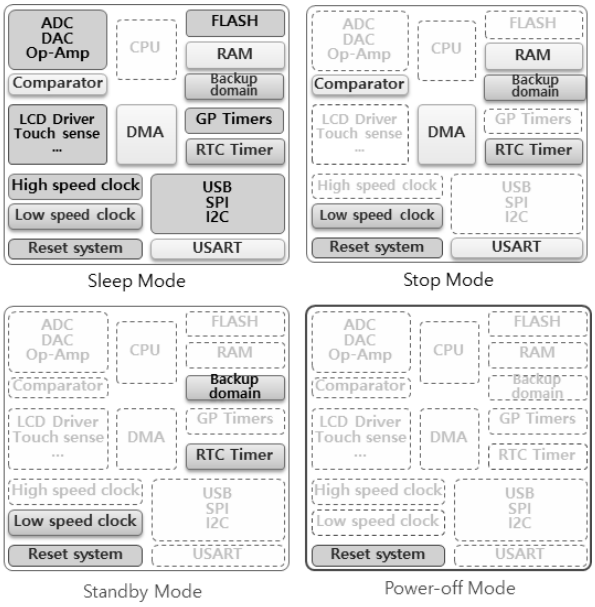


그림 1. STMicro에서 지원하는 전력 모드
Fig. 1. Power modes in STMicro MCU.

본 연구에서는 기존의 HW에서 지원하는 전력 모드를 이용해서 응용에 따라 최소 전력을 소모할 수 있도록 태스크를 관리하는 OS(nanoQplus)에 관한 것이다. 그림 1에서와 같이 대부분의 MCU에서는 4~5가지의 전력 관리모드를 제공하고 있으며, 그림은 STMicro의 전력 괸리모드에 따른 MCU 내부 모듈의 활성화 및 비활성화를 보여준다. 전력 소모가 적은 모드로 가면 반응 속도가 느려지나 비활성화 모듈이 많아 전력 절감 효과는 크게 되므로 구동 태스크에 따라 적절한 모드 선택이 요구된다. 관련해서 웨어러블 디바이스에서 많이 사용되는 FreeRTOS[7] 및 ChibiOS[8]는 기본적으로 전력모드의 첫 단계인 Sleep 모드만을 지원하고 있어, 충분한 전력 절감이 되지 않는다. 뿐만 아니라, 웨어러블 디바이스의 특성에 따라, 최소 전력 소모가 가능한 제품 개발 보드의 power-off 모드를 지원하여 전력 절감을 달성할 수 있는 방법을 제시한다.

Ⅱ. 본 론

1. HW기반 저전력 모드와 OS(nanoQplus-W[9])

앞서 언급했듯이, 배터리 기반 포터블 디바이스에서 핵심 이슈는 소모되는 전력 절감이다. 이를 위해서 그림 1에서와 같이 대부분의 HW에서는 각 MCU의 특성에 맞게 다양한 전력 모드를 지원하게 된다. 하지만 대부분의 응용 개발자는 각종 HW에서 지원하는 전력회로를 효과적으로 제어 및 활용을 하지 못하고 있다. 그림 2를 보면 각 전력모드에 따라 활성 모듈이 달라져서 전력 소모 레벨이 확연히 차이가 나는 것을 볼 수 있다.

이와 같이 전력 절감을 위한 회로의 활용성을 높이기 위해서는 다중 태스크를 효과적으로 수행하고 최소 전력모드로 진입할 수 있도록 해주는 스케줄러를 필요로 한다. 즉, 응용 개발자가 자신의 응용을 손쉽게 개발할 수 있도록 지원해 주는 운영제제를 필요로 한다.

	Normal	Sleep	Stop	Standby
MCU		Off	Off	Off
Flash				
PLL				
SRAM				
I/O Device	Active	Active	Off	
Wake-up	Active			
Power(mW)	475	295	233	220

그림 2. 전력모드별 소모 전력
Fig. 2. Power consumption for power mode.

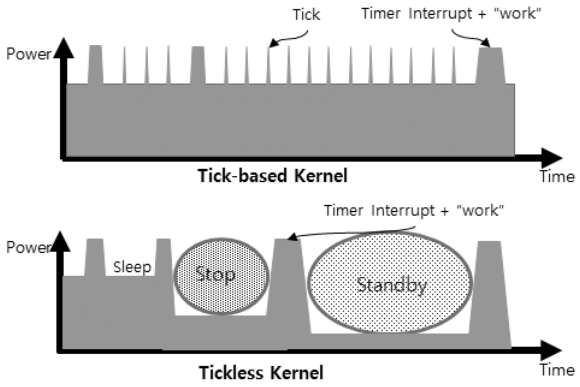


그림 3. 틱기반 및 틱리스 커널간 전력 소모비교
Fig. 3. Difference of Power consumption between Tick based and Tickless Kernel.

최근 페블 워치 등의 웨어러블 경량 디바이스에 많이 사용되는 FreeRTOS[7]에서는 Sleep 모드까지 지원하는 코드가 공개되어 있다. 이 운영체제는 기존의 센서용 IoT 디바이스에 적용되는 tick 기반의 운영체제와는 달리, 그림 3과 같이 이벤트 기반인 Tickless 커널로 되어있다. 하지만 MCU 시스템 클럭에서 제공하는 카운터 레지스터의 최대 지원 수마다 깨어나기 때문에 비활동성 구간이 긴 경우에는 전력 절감효과가 떨어지게 된다. 본 연구에서 개발한 nanoQplus-W 운영체제도 웨어러블 태스크 특성과 저전력 효과를 위해 Tickless 커널기반으로 구조 변경하였으며, 카운터 레지스터 최대값을 운영체제 내부에서 확장 처리하도록 하여 Tick 발생을 최대한 억제하도록 하였다. 이렇게 함으로써 그림 3처럼 전력모드 중간에 깨어나는 횟수를 최소화함으로써 불필요하게 소모되는 전력을 줄였다.

뿐만 아니라, HW MCU에서 standby mode를 지원하는데, 이는 MCU 내부 백업 SRAM을 이용하여 MCU 전원이 off 된 후 재시작할 때 필요한 상태 정보를 저장하고 복구하는 절차를 걸쳐야 하는데, 이를 위한 SW 처리 절차를 모두 지원하여 HW에서 제공하는 모든 전력 관리 모드를 활용할 수 있도록 하였다.

2. Cold Restart 모드 이용 전력 관리

저전력 달성을 위해서 HW에서 제공하는 전력 관리모드의 효과적 사용을 OS에서 제어하는 하는 것이 현재까지 주요 이슈였는데, 이런 수준의 전력절감 방식으로는 웨어러블 디바이스 충전 불편을 해소하지 못한다. 이는 단순하게 MCU의 소모 전력을 제어하는 의미하며, MCU 외부의 통신 모듈, 플래시 메모리 등의 전력 소모가 많은 부품에 대한 전력 소모 제어에는 여전히 문제가 존재하기 때문이다.

이런 문제점에 덧붙여 헬스케어 웨어러블 디바이스에 탑재되는 센서 응용이나 통신 태스크들은 대체로 주기적으로 수행되면서 장기간 사용되지 않은 경우가 자주 있음을 고려해서, 상황에 맞는 초저전력 모드의 제공 필요성이 있어 관련 전력 관리 모드를 외부 회로 및 SW 제어를 통해 제안하였다.

상기 언급한 초저전력 모드인 Cold Restart 모드는 수행될 태스크가 오랜 시간동안 수행하지 않을 경우, 보드의 인가된 전원을 차단함으로써 MCU 외부의 장치까지 제어하는 전력 관리 모드이다. 이 전력모드가 구동되기 위해서는 보드의 전원을 차단할 때, 시스템의 컨텍스트(MCU의 레지스터 상태와 RAM에 저장된 데이터 등)를 전원이 차단되어도 데이터가 사라지지 않는 비휘발성 메모리인 플래시에 저장해야 한다. 그리고 전원 인가 시 그 상태를 그대로 복원하기 위해서, 보드의 부팅이 수행된 후 저장한 데이터를 저장 이전의 위치에 레지스터나 RAM으로 복구하고, 다음 상태(PC+1)에서 시작되도록 하여야 한다. 제안한 Cold Restart 모드가 다른 시스템의 컨텍스트 저장/복구 기능과 다른 것은 복원될 때 유휴시간만큼 시간이 흐른 뒤이므로, 복원시의 시점이 저장시보다 더 미래이다. 따라서 OS의 타이머 및 스케줄러는 복원 시 저장된 시점이 아니라 유휴시간이 지난 부분을 계산하여 스케줄링에 반영해야 한다.

nanoQplus-W에서는 이를 위해 컨텍스트 저장 및 파일 시스템 저장이 가능한 다중 파티션 지원 경량 플래시 파일 시스템을 지원하고, 태스크의 임의 스케줄링 변경이 가능한 이벤트 처리기를 사용하였다. 그리고, 컨텍스트 저장 파티션에서는 반응시간을 최소화하기 위해 간단한 bad block manager 및 순환 wear-leveling 방식을 채택하였다. 일반 파일시스템도 지원하여 센서 데이터 저장이 되도록 지원하였다.

또한, 시스템 내의 모든 컨텍스트를 저장/복구하는 완전 실시간 재시작(Full Cold Restart) 모드와 응용 프로그램 개발자가 지정한 일부 컨텍스트/메모리만 저장/복구하는 부분 실시간 재시작(Half Cold Restart) 모드를 동시 지원한다. 완전 실시간 재시작 모드는 개발자가 실시간 재시작에 대해 의식하고 준비하지 않아도 OS 단계에서 자동으로 지원하므로, 코드가 단순해지고 개발 단가가 저렴해 지는 이점이 있지만, 모든 컨텍스트를 저장해야 하므로 플래시 메모리의 많은 용량을 사용하여 플래시의 수명은 단축될 수 있고, 반응시간도 늘어나는 단점이 발생한다. 반면, 부분 실시간 재시작 모드는 용도에 따라 저장 용량을 최소화할 수 있으므로, 플래시 사용이 적어 더 긴 수명을 기대할 수 있다. 그러나 개발자가 어떤 컨텍스트를 저장/복원해야 할지 선택하고 운용해야 하므로 응용 구현이 복잡해진다.

표 1은 모든 컨텍스트가 포함되고 센서 데이터도 저장될 경우를 대비한 최악의 상황을 지표하는 1MB와 일반적인 경우인 128KB의 데이터가 플래시 파일로 저장/복구할 때의 시간을 나타낸 것이다. 대부분의 경량 웨어러블 디바이스에 탑재되는 응용들에서는 저장 컨텍스트가 수십B~수십KB 이내일 것이므로, 10msec 이내의 충분한 반응시간 내로 최소의 전력을 소모하게 된다.

표 1. 플래시 메모리 저장/복구 시간
Table1. Time Laps for storing/loading data between flash memory and register/RAM.

데이터 사이즈	1MB	128KB
복구 시	83.97msec	10.57msec
저장 시	149.15msec	19.78msec

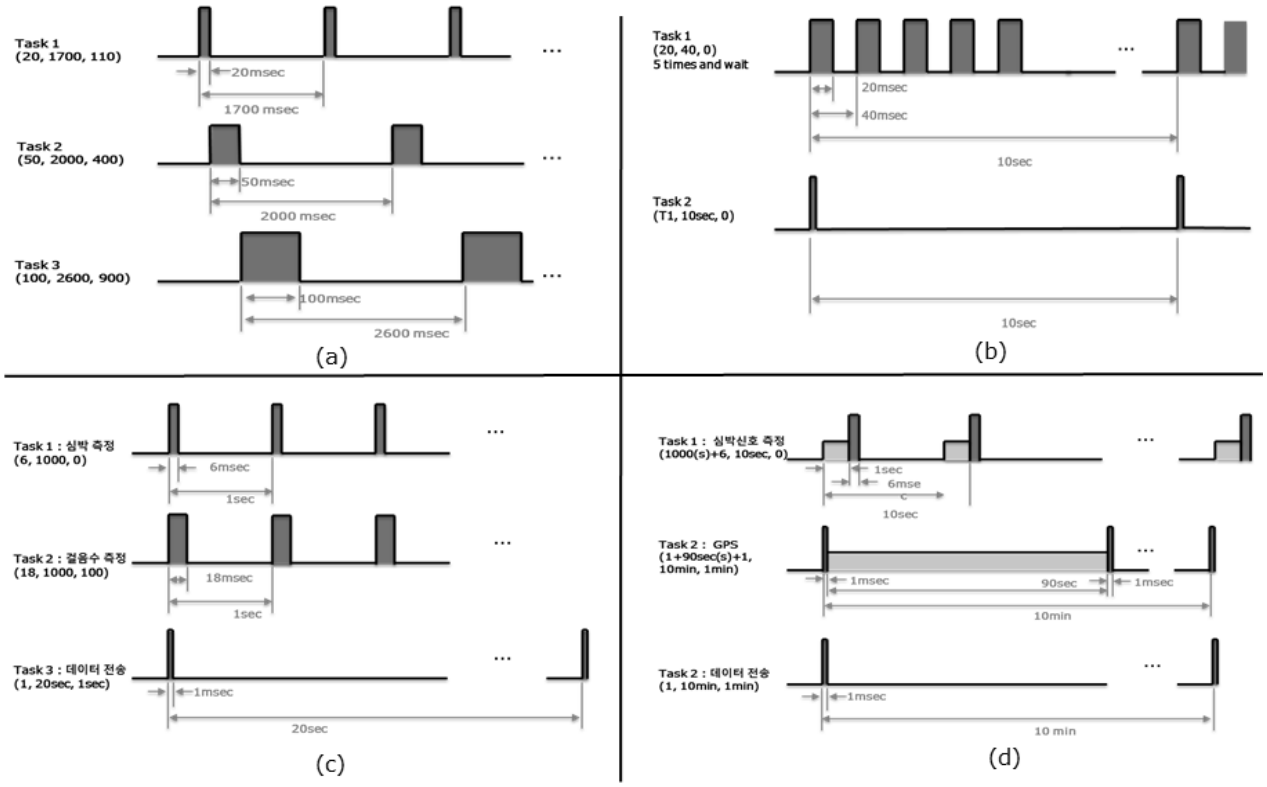


그림 4. 각 OS의 성능 분석을 위한 4가지 시나리오
Fig. 4. number of 4 Scenarios for performance analysis on three Operating systems.

Ⅲ. 실 험

1. 시험 환경

경량 웨어러블 디바이스에서 많이 사용되는 AP는 SRAM과 플래시 메모리가 포함되어 있으며 Low Power/ Low Cost을 위해 자원이 한정적이다. 본 연구에서 사용하는 AP는 STMicro에서 제공하는 STM32F4 AP로 1MB 플래시 메모리와 192KB SRAM를 포함하고 있으며, 동작 주파수는 168Mhz로 설정하였다. 메모리의 제한 사항으로 고용량을 요구하는 uClinux 등의 운영체제는 탑재되지 못하며, 전력 소모관련 성능 시험을 위해서 본 연구에서 개발한 nanoQplus-W와 FreeRTOS 및 ChibiOS를 사용하였다.

보드에 소모되는 전력 측정은 몬슨 파워 모니터를 사용하였으며, 이는 초당 5000 샘플의 전력 소모 데이터를 수집이 가능하며, 반응시간 측정을 위해서는 50Mhz의 오실로스코프를 이용하였다.

전력 측정구간은 몬슨 모니터의 최대 저장 용량을 고려하여 210초 동안 측정하였으며, 이를 누적 소모 전력(mWh)으로 환산하여 상호 비교 하였다.

2. 사용자 응용기반 시나리오

HW에서 지원하는 전력모드를 웨어러블 응용에 따라 각 전력모드를 운영체제가 어떻게 효과적 제어하는 여부를 확인하기 위해서, 4 가지의 시나리오를 만들었다. 각 시나리오는 웨어러블 디바이스에 적용되는 응용을 운영체제의 스케줄링 태스크로 모델링하였으며, 각 태스크들은 주기적인 성질을 가지고 있다.

시나리오는 크게는 HW 전력모드 지원에 대한 기본 기능 동작을 확인하는 시나리오 2개와 실제 제품의 응용이 수행되는 과정을 모사한 2개의 시나리오로 구성되어 있다. 그림 4에서 (a)는 STM 개발보드에서 지원하는 전력모드가 모두 수행되는지 확인하는 시나리오이며, (b) Task 안에 Task가 수행되는 경우로, 센서로 부터 데이터를 수집한 후 모아서 한 번에 통신모듈에서 전송하는 시나리오로 이 둘은 기본 동작 확인 시나리오이다. (c)는 Fitbit Charge HR 제품에서 심박 측정과 걸음 수 측정을 하는 Task를 모사한 것으로 일정 시간마다 데이터를 폰으로 전송하는 시나리오이며, (d) 국내 N사의 독거노인 움직임을 모니터링하는 제품의 Task를 모사한 것으로 심박, GPS, 데이터 통신(LoRA)의 기능을 포함한 시나리오이다. 그림 4에서 보이는 각

Task의 숫자들은 Run 모드 수행시간, 태스크 주기, offset 시간을 의미하며, 시간의 흐름에 따라 태스크의 연속 수행 및 전력 모드 결정이 수행되며 이를 시각적으로 그림으로 표현하였다. 실제 시나리오 구동에서는 태스크의 주기와 활성 구간들의 겹침으로 동시에 연속적으로 태스크들이 수행되는 경우가 발생한다.

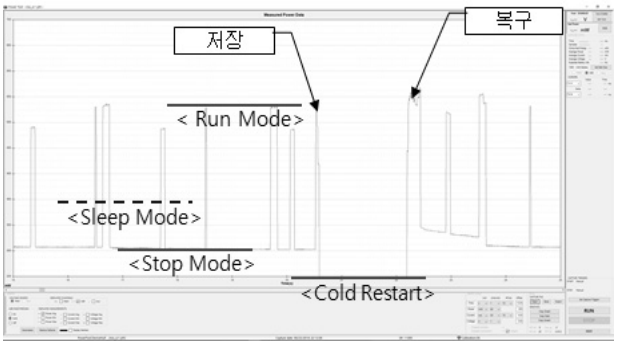


그림 5. 시나리오 1에서 전력 관리 모드 동작
Fig. 5. Power management mode in scenario 1.

그림 5는 시니리오 1의 다양한 태스크들이 수행될 때 나타나는 전력모드로 3가지의 전력모드가 제대로 동작됨을 보여주고 있으며, Sleep 모드의 경우에는 반응시간이 태스크가 요구하는 시간에 Stop 모드로 대체 가능하여 수행되지 않고 있음을 확인할 수 있었다. 제안한 Cold Restart 모드시에는 메모리의 저장/복구로 오버헤드가 존재함을 보여준다.

3. OS별 전력소모 비교 및 분석

성능 비교를 위해 사용된 OS는 3 가지로, ChibiOS는 전력 관리 모드를 전혀 지원하지 않으며, Sleep해야 하는 경우 NOP idle로 동작한다. FreeRTOS는 Sleep 모드까지 지원한다. nanoQplus-W는 HW에서 지원하는 4가지 전력 관리 모드 및 보드 레벨의 power-off 모드인 Cold Restart를 지원한다.

표 2. 시나리오별 각 운영체제의 전력소모 비교
Table2. Power Consumption on each OS according to Scenario.

시나리오	1	2	3	4
ChibiOS	11.23	11.0	11.09	12.61
FreeRTOS	5.75	5.13	5.22	6.06
NanoQplus-W	2.87	0.23	1.6	2.92

표 2는 시나리오에 따른 각 OS에서 소모되는 전력을 나타내는 것으로 모든 시나리오에서 본 논문에서 제안한 운영체제가 확실한 전력 절감을 보였다. 특히, 시나리오 2에서는 22.3배나 47.8배의 극단적인 절감효과를 보이는데, 이는 태스크의 장기간 idle 모드 진입으로 Cold Restart 모드를 많은 부분에서 적용할 수 있었기 때문이다. 시나리오 1, 4의 경우에는 다양한 전력모드가 수행되지만 초저전력을 달성할 수 있는 Cold Restart 모드 구간이 상대적으로 짧아, 각 운영체제에 비해 2배, 4배 정도의 절감효과를 보였다. 시나리오 3의 경우에는 짧고 빈번한 태스크의 수행으로 Cold Restart 모드가 수행되지 않지만 반응성이 뛰어나고 전력 절감 효과가 좋은 Stop, Standby 모드가 적용되어 3배 이상의 소모 전력을 절감하였다.

Cold Restart 모드 및 각종 전력 관리 모드가 적용되는 nanoQplus-W를 이용해서 전력 관리 모드가 추가됨에 따라 소모되는 전력을 살펴보면 그림 6에 나타난다. 시나리오 2에서는 Cold Restart의 효과를 확인할 수 있으나, 시나리오 4의 경우에는 적용되는 횟수가 너무 작아 상대적으로 큰 효과를 얻지 못하고 있다. 하지만, 각각의 전력모드가 적용됨에 따라 제시한 시나리오에서 기울기가 감소하는 것으로 보아 상당한 전력 소모 절감을 가져올 수 있었다.

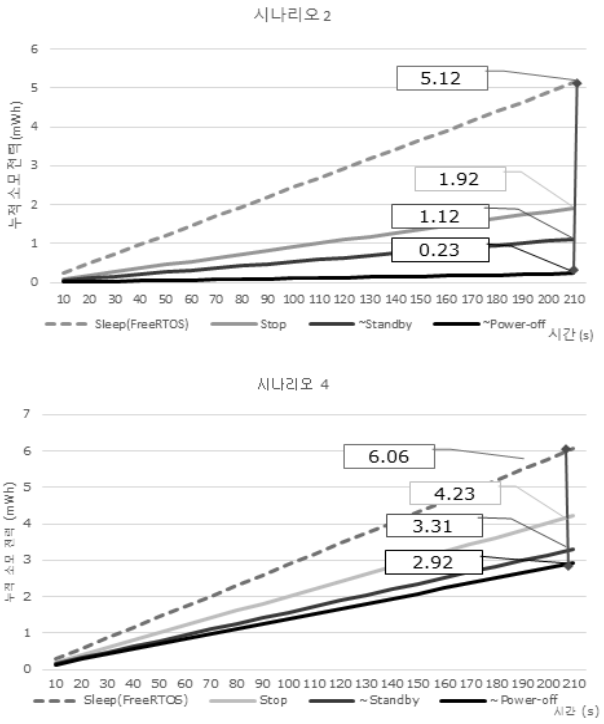


그림 6. Cold Restart 성능 비교
Fig. 6. Performance Comparison for Cold Restart.

한편, 최소 전력 소모를 위해서는 극강의 전력 관리 모드인 Cold Restart를 적용해야 하지만, 플래시 메모리를 사용해서 저장/복구를 해야 하므로 반응시간이 다른 전력관리 모드에 비해 현저히 떨어진다. 뿐만 아니라, 플래시 메모리를 구동해야 하므로 저장/복구 시에 상대적으로 저장/복구 시간 동안 전력 소모 오버헤드가 존재한다. 또한, 플래시 메모리의 쓰기 횟수 제한(Single Level Cell 경우 5000번)을 고려하면 제품프의 수명이 단축될 수 있어, 시나리오 2처럼 장기간 비활성 태크스에 적용될 경우 유리하다. 덧붙여 그림 6의 시나리오 2, 4에서 보듯이, Standby 모드까지 적용으로도 충분한 전력 절감 효과를 볼 수 있으므로 응용 및 제품 사양에 따른 고려가 필요하다.

Ⅳ. 결 론

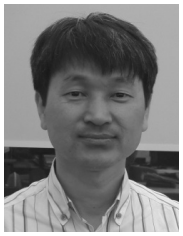
본 논문에서는 현재까지 HW에서 지원되는 전력모드를 효과적으로 SW에서 제어될 수 있도록 기존 운영체제를 수정하고 전력 관리 알고리즘을 제안하였다. 뿐만 아니라 보드 레벨에서 전력을 제어할 수 있도록 추가적으로 극강의 전력모드 Cold Restart를 제시하였으며 시나리오 4개를 통해 성능을 제시하였다. 각 시나리오에서 전력 관리 모드를 지원함에 따라 소모되는 전력량을 상당히 절감할 수 있음을 보여주었으며 기존 운영체제보다 월등한 성능을 보임을 데이터로 제시하였다.

또한 본 연구에서 개발한 운영체제는 많은 다른 연구자들이 경량 디바이스의 저전력 연구를 효과적으로 진행할 수 있도록 소스코드 형태로 웹에 공개[9]되어 있다.

REFERENCES

[1] C.Wang, W.Lu, M.R.Narayanan, S.J.Redmond, and N.H.Lovell, “Low-power technologies for wearable telecare and telehealth systems: A Review,” Vol. 5, issue 1, pp. 1-9, March 2015.
[2] B.S Song, “Low power Solution for smart sensors of IoT services,” The Magazine of the IEIE, Vol. 43, no. 1, pp. 24-31, 2016
[3] S. Mittal, “A Survey of techniques for Improving Energy Efficiency in Embedded Computing Systems,” International Journal of Computing Aided Engineering and Technology, vol. 6, no. 4, pp. 440-459, Jan 2014.
[4] V.Delaluz, N.Vijaykrishnan, A.Sivasubramaniam, and M.J.Irwin, “Memory energy management using software and hardware directed power mode control,” Technical Report CSE-00-004. Dept. of Computer Science and Engineering, Pen State University, April 2000.
[5] Q. Qiu, Q.Wu and M.Pedram, OS-directed Power management for Mobile Electronic Systems, in Proc. of 39th Power Source Conf., pp. 506-509, 2000.
[6] J.M Whang, B.K Nam, “Trends on Wearable Operating System,” The Magazine of the IEIE, Vol.42, No.6, pp 46-55, 2015.
[7] FreeRTOS, http://www.freertos.org/
[8] ChibiOS, www.chibios.org/
[9] nanoQplus-W, https://github.com/nanoQplus/nanoQplus-W

저 자 소 개



김 선 태(평생회원)-교신저자
1997년 KAIST 전자 및 전자공학과 학사
2000년 서울대학교 전기공학부 석사
2012년 고려대학교 메카트로닉스 박사
2000년～현재 ETRI 임베디드SW연구부 실장
<주요관심분야: 경량 OS, 저전력 IoT 네트워킹, 실시간 처리기술, 경량 웨어러블 디바이스, SW-SoC융합플랫폼>

박 형 준(정회원)
2014년 상명대학교 컴퓨터과학 학사
2017년 과학기술연합대학원대학교 컴퓨터소프트웨어학과 석사
<주관심분야: RTOS, Wearable 디바이스, 초저전력 제어 기술>

박 호 준(정회원)
2005년 건국대학교 컴퓨터공학과 박사
2006～현재 ETRI 임베디드SW연구부 선임연구원
<주관심분야: RTOS, Wearable 디바이스, 전력관리>

우 덕 균(정회원)
1993년 홍익대학교 컴퓨터공학과 학사
1995년 홍익대학교 컴퓨터공학과 석사
2001년 홍익대학교 컴퓨터공학과 박사
2001년～현재 ETRI 임베디드SW연구부 책임연구원
<주관심분야: 컴파일러, 실시간 운영체제, 웨어러블 디바이스 개발킷>