

논문 2018-13-14

# 하드웨어 저전력 기능을 활용한 웨어러블 운영체제의 하이브리드 가버너 (Hybrid Governor for Wearable OS Using H/W Low-power Features)

이 성 업, 김 형 신\*

(SungYup Lee, HyungShin Kim)

Abstract : Wearable devices have become widespread. Fitness band is one of common wearable devices, providing useful functions. It helps users to monitor and collect their status such as heart rate and travel distance. Wearable devices, including fitness bands, are designed in small size and it ends up having small battery capacity. In that regard, it is necessary to expand the lifetime of wearable devices. Conventional power management scheme of wearable devices is based on DVFS Ondemand Governor and peripheral control by timeout event, such as turning off the LCD. In this paper, we propose a hybrid governor applying hardware supporting low power mode such as sleep mode to exploit the periodicity of fitness band task. In addition, we show hybrid governor outperforms in power consumption than conventional power management scheme of wearable devices based on Ondemand Governor through experiments.

Keywords : Power management governor, Wearable device, Wearable OS, Low power

## I. 서 론

웨어러블 컴퓨팅 기기에 대한 관심이 높아짐에 따라 시계 혹은 팔찌와 같은 다양한 형태의 웨어러블 기기가 널리 보급되었다 [1]. 웨어러블 기기는 사용 목적의 특성상 태생적으로 크기가 작다. 사용자의 몸에 착용했을 때 크기로 인한 불편함을 최소화해야하기 때문이다 [2]. 따라서 배터리의 용량이 작아 기기의 가용시간이 짧다. 상기 문제를 해결하기 위한 기기의 전력 관리 문제는 웨어러블 기기 관련 연구에서 가장 관심도가 높고 중요한 분야 중 하나이다 [3].

웨어러블 기기는 고성능의 스마트 워치부터 Fitbit과 같은 저성능의 피트니스 밴드까지 다양한

형태의 제품들이 출시되어 있다. 스마트 워치 제품군과 피트니스 밴드 제품군은 기능적인 성격이 다르다. 스마트 폰의 알림을 사용자에게 전달하고 빈번히 상호작용을 해야 하는 고성능의 스마트 워치와 달리 피트니스 밴드 제품군은 사용자의 심박 수 혹은 걸음 수, 이동 거리와 같은 사용자의 상태보를 모니터링 하고 수집한다. 상기의 사용자 상태 정보를 모니터링 하고 수집하기 위해서 피트니스 제품군의 작업은 주기적인 작업으로 구성되어 있으며, 그 특성상 사용자의 상호작용과 개입이 적다. 이런 작업의 특징을 잘 이용하면 피트니스 밴드와 같은 제품군에서의 전력 소모량을 줄일 수 있다.

기존의 웨어러블 기기용 운영체제의 전력 관리 기법은 고전적인 전력 관리 기법과 크게 다르지 않다. 전력 소모량을 제어하기 위해 CPU와 기기를 제어한다. CPU의 전력 소모량을 제어하기 위해 쓰이는 기법 중 가장 대표적인 것이 동적 전압 및 주파수 가변 기법 (DVFS)과 CPU의 저전력 모드이다. DVFS는 CPU가 동작중일 때 전압과 동작 주파수를 조절하여 소모 전력을 줄이고, 저전력 모드는 CPU를 사용하지 않을 때 동작을 잠시 멈추어 전력

\*Corresponding Author (hyungshin@cnu.ac.kr)

Received: Jan. 11 2018, Revised: May 21 2018,

Accepted: June 4 2018.

S. Lee: University of Science & Technology,  
H. Kim: Chungnam National University

※ 본 연구는 2016년도 충남대학교 학술연구진흥사업의 지원을 받아 수행되었음.

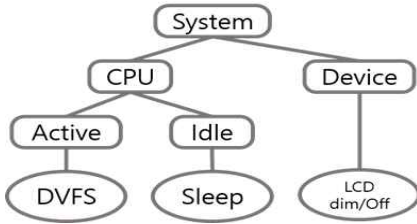


그림 1. 기존 운영체제의 전력 관리 기법

Fig. 1 Power management scheme of wearable OS

소모를 줄인다. 이 외에도 현재 사용하지 않는 특정 주변기기의 전원을 제어해 전력 소모를 줄인다. 대표적인 웨어러블 기기용 운영체제인 삼성사의 타이젠과 구글사의 안드로이드 웨어에서는 그림 1과 같이 CPU의 상태에 따라 DVFS를 적용하고, 기기의 상태에 따라 LCD를 끄거나, 밝기를 감소시키거나, CPU를 저전력 상태인 Sleep 상태에 두는 기법을 사용해 전력 소모량을 줄인다 [4]. 이런 전력 관리 기법은 현재 시스템의 상태에 따라 동적으로 전력 소모량을 제어하기 때문에 동적 전력 관리 기법으로 분류된다 [5].

한편 웨어러블 기기에 활용할 수 있는 임베디드 시스템에는 기존의 시스템에서 활용하고 있는 것보다 다양한 저전력 기능들을 지원하고 있다. STMicro사의 STM32L4 계열의 임베디드 보드는 저전력 기능에 초점을 맞춘 시스템으로 CPU가 유휴 상태일 때 활용할 수 있는 다양한 저전력 모드를 지원한다. 종래의 Sleep 모드를 비롯해 Stop 모드, Shutdown 모드, Standby 모드 등 다양한 wake-up 지연시간과 소모 전력량 수준의 저전력 모드를 갖추고 있다. 더불어 상기 저전력 모드들은 CPU의 전력 소모량과 기기 전체의 전력 소모량까지 함께 제어할 수 있다 [6].

주기적인 작업이 주를 이루는 피트니스 밴드 제품군의 작업 성격은 이러한 다양한 저전력 모드를 적용하기 적합하다. 저전력 모드는 일반적으로 CPU를 정지시키기 때문에 프로그램의 실행이 불가능하다. 다시 프로그램을 실행 가능한 상태로 되돌리는데 걸리는 시간인 wake-up 지연시간 때문에 사용자와 상호작용이 빈번하다면 사용자의 사용감에 큰 영향을 준다. 반면에 피트니스 밴드 제품군은 사용자와의 상호작용이 적기 때문에 wake-up 지연시간이 크지만 전력 소모량 절감 수준이 높은 저전력 모드를 활용해도 사용감에 큰 영향을 주지 않는다.

본 논문에서는 DVFS를 운영체제 수준에서 제어

하는 리눅스의 가버너와 하드웨어에서 지원하는 다양한 저전력 기능들을 함께 활용하는 하이브리드 가버너를 구현하고, 실험을 통해 기존의 단순한 전력 관리 시스템 보다 성능이 뛰어난 것을 보인다.

본 논문의 2장에서는 관련 연구를 소개하고, 3장에서 본 연구에 필요한 배경 지식들을 설명한다. 4장에서는 본 연구에 대해 설명하고, 5장에서 실험의 방법과 결과를 보이고 6장에서 결론을 맺는다.

## II. 관련 연구

### 1. Ondemand 가버너

Ondemand 가버너는 리눅스 시스템에 탑재되어 있는 운영체제 수준의 전력 관리 가버너로, DVFS 기법을 활용한다. 일정 시간 단위로 현재 CPU의 사용률을 샘플링 하고 현재 사용률이 기준치 보다 높으면 CPU의 동작 주파수를 최대로 높인다. 반대로 현재 사용률이 일정 기준치 보다 낮으면 CPU의 동작 주파수를 20%씩 낮추는 방식으로 동작 한다 [7]. 서버 급의 고성능 시스템에서부터, 저성능의 웨어러블 시스템까지 전력 소모량을 절감하기 위한 기법으로써 DVFS를 활용하는 Ondemand 가버너를 활용 한다. 리눅스 커널을 활용하는 웨어러블 운영체제인 안드로이드 웨어와 타이젠에서도 이를 그대로 활용한다 [8].

### 2. 안드로이드 웨어와 타이젠

안드로이드 웨어와 타이젠은 각각 구글과 삼성에서 개발한 웨어러블 기기용 운영체제이다. 안드로이드 웨어는 기존 스마트폰의 안드로이드 운영체제를 웨어러블 기기에 맞추도록 수정한 것이다. 안드로이드 웨어와 타이젠은 모두 기존의 리눅스 커널을 기반으로 개발되어 리눅스 시스템의 전력 관리 가버너를 그대로 활용한다. 추가적으로 기기의 전력 소모량을 줄이기 위해 사용자의 입력 시간을 기반으로 타임 아웃 이벤트를 활용해 LCD의 밝기를 줄인다. 계속해서 사용자의 입력이 발생하지 않고 타임 아웃 이벤트가 다시 발생한다면, LCD를 끈다. 또 다시 사용자의 입력 없이 시간이 지나 타임 아웃 이벤트가 발생하면 기기를 Sleep 모드에 두는 방법으로 동적 전력 관리 기법을 적용한다 [9].

### 3. 하드웨어 지원 저전력 기능

“Cold Restart를 이용한 웨어러블 디바이스의 초저전력 핵심기술” 연구에서는 기기의 저전력 모

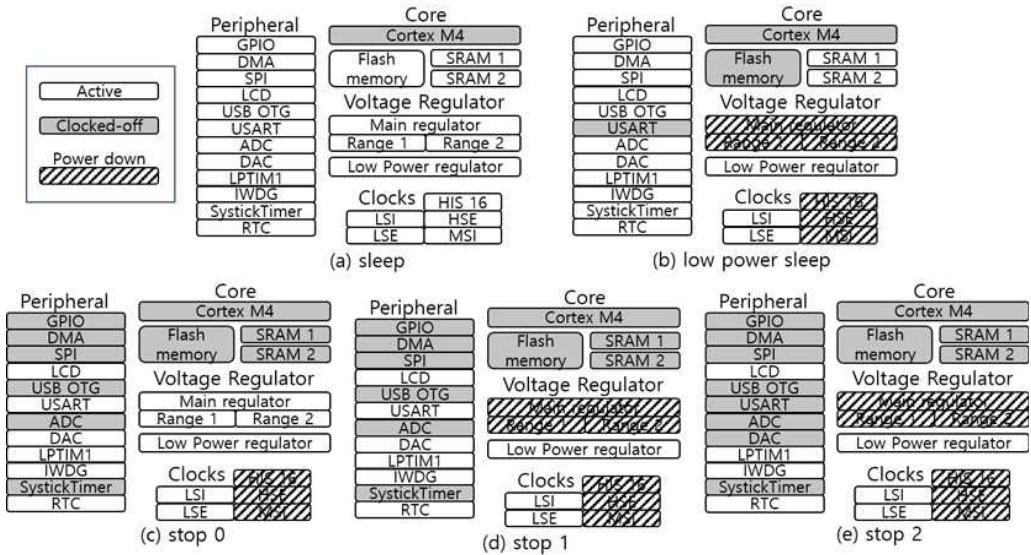


그림 2. STM32L4의 저전력 모드  
Fig. 2 Low-Power modes of STM32L4

드를 극한으로 끌어올리는 연구를 소개했다 [10]. 현재 실행중인 문맥 정보를 모두 플래시 메모리에 저장한 뒤, 외부 타이머 저전력 모드 시간 정보를 넘겨주고, 보드 자체의 전원을 차단한다. 그 후, 외부 타이머의 만료되면 다시 전원을 공급하여 시스템을 부팅하고, 플래시 메모리에 저장되어있는 문맥 정보를 복원하는 방식으로 저전력 모드 시 소모 전류를 외부 타이머의 소모 전류량으로 최소화 시킨 연구이다.

### III. 배경 지식

#### 1. STM32L4의 저전력 기능

STM32L4 Discovery 보드는 STMicro 사에서 개발한 저전력 기능에 초점을 맞춘 임베디드 보드로써, Cortex-M4를 사용한다. STM32L4는 다음과 같은 다양한 저전력 모드를 지원한다 [11, 12]. 현재 보드에서 지원하는 다양한 저전력 모드를 활용하기 위해서는 개발자가 직접 저전력 모드들의 특성을 확인하고, 펌웨어로써 개발해야한다. 본 연구에서는 이 저전력 기능들을 운영체제가 제어하도록 한다.

STM32L4의 저전력 모드들은 기본적으로 코어를 멈추고, 각각의 주변장치들에 공급되는 클럭을 클럭 게이팅을 통해 제어한다. 또한 Wait For

Interrupt (WFI) 혹은 Wait For Event (WFE) 명령어를 통해 저전력 모드에 진입할 수 있으며, 인터럽트 혹은 이벤트를 통해 깨어날 수 있다. STM32L4의 저전력 모드들은 그림 2와 같다. 그림 2의 (a)부터 (e)는 sleep 모드부터 stop2 모드까지 보드의 코어 및 전원 공급부, 클럭 소스들의 상태를 도식화 한 것이다. sleep 모드가 전력 소모가 가장 크고, stop2 모드가 전력 소모가 가장 적으며, wakeup 지연 시간은 반대로 sleep 모드가 가장 짧으며 stop2 모드가 가장 길다 [13].

#### 2. FreeRTOS

FreeRTOS는 현재 시장을 선도하는 실시간 운영체제로써, GPL을 기반으로한 FreeRTOS 라이선스하에 배포된다. 오픈소스 프로젝트로, 커널을 수정하지만 애플리케이션 상용목적으로 무료로 사용가능하다는 특징이 있다 [14].

FreeRTOS는 가볍고 간단한 구조로 이루어져 있으며, 커널은 3개의 C파일에 모두 구현되어 있다. FreeRTOS는 현재 다중 쓰레드/태스크와 뮤텝스, 세마포어와 소프트웨어 타이머를 지원하며, 저전력 기능으로 틱리스 (Tick-less) 모드를 지원한다. 틱리스 모드는 systick 인터럽트의 발생 간격을 늘려 소모 전력을 줄인다. 현재 DVFS를 활용한 전력 관리 가버너가 구현되어 있지 않다 [15].

### 3. 동적 전압 및 주파수 가변 기법

동적 전압 및 주파수 가변 기법 (DVFS)은 입력 전압과 CPU의 동작 주파수를 낮추어 소모 전력을 줄이는 기술이다. CPU 내부의 CMOS 소자가 소비하는 에너지는 다음과 같다.

$$E \cong E_{dynamic} = C \times f \times V^2t \quad (1)$$

CMOS의 충전 및 방전 시 소모 에너지  $E_{dynamic}$ 은 CMOS 소비 에너지의 대부분을 차지하며, 전압  $V$ 와 동작 주파수  $f$ 에 따라 커진다. 따라서 CPU의 동작 주파수와 입력 전압을 줄여 전체 에너지 소모량을 크게 줄일 수 있다. DVFS는 실시간으로 동작 주파수와 입력 전압을 조절해 전력 소모량을 줄이는 기법이다 [16].

## IV. 하이브리드 가버너

하이브리드 가버너는 저전력 성능이 뛰어난 하드웨어 지원 저전력 모드의 느린 응답속도를 보완하기 위해, 태스크와 태스크 사이의 Idle 시간이 짧은 경우 DVFS 기능을 일종의 완충 장치로 사용하는 저전력 가버너이다.

STM32L4 및 FreeRTOS를 위한 하이브리드 가버너는 시스템의 전력 소모량을 관리하기 위해 두 가지 방법을 사용한다. 첫 번째는 CPU의 동작 주파수를 동적으로 변경하는 DVFS이고, 두 번째는 시스템을 저전력 모드로 진입시키는 것이다. 두 가지 기능은 현재 CPU의 사용률과 태스크 블록 시간을 기반으로 적용된다.

### 1. 하이브리드 가버너의 기능

STM32L4 및 FreeRTOS를 위한 하이브리드 가버너가 시스템의 전력 소모량을 관리하기 위해 활용하는 DVFS는 현재 CPU의 사용률을 바탕으로 동작하며, 저전력 모드는 실행 중인 태스크들의 블록 시간을 바탕으로 적용된다.

#### 1.1 하이브리드 가버너의 DVFS 관리

하이브리드 가버너는 현재 작업량에 따라 CPU의 동작 주파수를 동적으로 변경하기 위해 CPU의 사용률을 활용한다. FreeRTOS의 스케줄러는 선점 및 우선순위, 타임 슬라이스 기반의 스케줄러이다. 때문에 실행할 태스크가 없을 시 윈도우 및 리눅스 운영체제와 같이 idle 태스크를 실행한다. 따라서

스케줄러 호출 시 현재 실행되는 태스크가 idle 태스크인지 여부를 가려 샘플링을 시행하면 다음과 같이 계산하여 CPU 사용률을 계산할 수 있다.

$$Usage_{CPU} = \frac{TickCount_{total} - TickCount_{idle}}{TickCount_{total}} \quad (2)$$

여기서  $Usage_{CPU}$ 는 CPU 사용률이고,  $TickCount_{total}$ 은 샘플링 윈도우 크기이자 샘플링 중에 발생한 systick 인터럽트 횟수이며,  $TickCount_{idle}$ 은 샘플링 윈도우 내에서 idle 태스크가 실행되는 동안 발생한 systick 인터럽트 횟수이다. 계산한 CPU 사용률을 바탕으로 기준 값 보다 CPU 사용률이 높으면 CPU 동작 주파수를 최대로 설정한다. 반대로 CPU 사용률이 기준 값 보다 낮으면, CPU 동작 주파수를 한 단계씩 감소시킨다.

### 1.2. 하이브리드 가버너의 저전력 모드 관리

하이브리드 가버너는 시스템의 전력 소모량을 관리하기 위해 STM32L4 보드에서 지원하는 몇 가지 저전력 모드를 활용한다. 일반적인 저전력 모드와 마찬가지로 STM32L4의 저전력 모드 또한 코어 클럭 공급이 중단되고, 코어의 동작이 멈추기 때문에 어떠한 프로그램도 실행될 수 없다. 그렇기 때문에 STM32L4 보드의 저전력 모드를 활용하기 위해 CPU가 동작할 필요가 없는 상황을 활용해야 한다. 이를 위해 태스크들의 블록 시간을 계산해야 할 필요가 있다. 웨어러블 응용의 태스크들은 주기적인 태스크가 주를 이루고, 태스크의 주기는 태스크를 블록 상태로 만든다. FreeRTOS에서는 이를 블록 태스크 큐에서 관리한다. 따라서 블록 태스크 큐를 통해 태스크들의 블록 시간을 계산할 수 있다. 이렇게 계산한 태스크 블록 시간을 바탕으로 길이에 따라 wakeup-latency를 고려해 적당한 저전력 모드를 활용할 수 있다.

### 2. 하이브리드 가버너의 동작

STM32L4 및 FreeRTOS를 위한 하이브리드 가버너는 그림 3과 같은 실행 흐름을 가진다. 시스템이 부팅되고 systick 인터럽트 및 코어 클럭 설정 등 시스템을 정상적으로 동작시키기 위한 설정들이 진행된다. 그 후 주기적으로 systick 인터럽트가 발생하는데, FreeRTOS는 이를 타임 슬라이스로 활용하여 스케줄링을 수행한다. 그림 3의 Scheduler 부분과 같이 systick 인터럽트가 발생하고, systick 인터럽트 핸들러에서 FreeRTOS의 스케줄러를 호

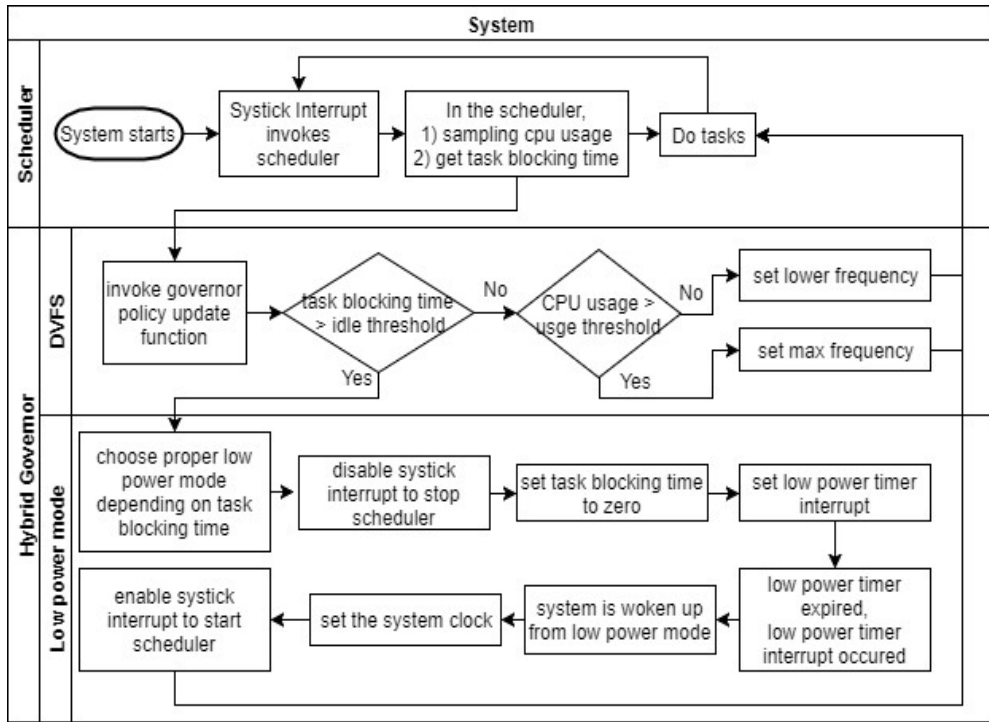


그림 3. 하이브리드 가버너의 순서도  
Fig. 3 Flow chart of hybrid governor

출 한다. 스케줄러에서는 1) CPU 사용률을 샘플링 하고, 2) 현재 태스크 큐에서 다음 태스크가 깨어날 시간을 계산한다. 앞서 계산한 CPU의 사용율과 태스크 블록 시간을 하이브리드 가버너의 정책 갱신 함수를 호출하며 매개변수로 넘겨준다. 하이브리드 가버너의 정책 갱신 함수에서는 태스크 블록 시간을 미리 설정된 idle threshold 값과 비교한다. 태스크 블록 시간이 충분히 길지 않으면 CPU 사용률을 확인하고, 정책에 따라 최대 동작 주파수로 설정하거나, 미리 설정된 정책에 따라 한 단계씩 동작 주파수를 낮춘다. 이 부분이 그림 3의 DVFS를 활용하는 하이브리드 가버너의 전력관리 기능이다.

만약 태스크 블록 시간이 시스템이 저전력 모드에 진입해도 좋을 만큼 충분히 길다면, 태스크 블록 시간의 길이에 따라 sleep, low power sleep, stop0, stop1, stop2 모드 중 적당한 저전력 모드를 선택한다. 저전력 모드의 진입에 앞서, systick 인터럽트가 저전력 모드 탈출을 야기하므로, systick 인터럽트를 비활성화 시킨다. 또한 FreeRTOS의 시간은 systick 인터럽트 횟수를 의미하는 tick count를 기반으로 계산되므로 systick 인

터럽트가 중지됨에 따라 스케줄링이 중지되기 때문에, 다음 태스크가 깨어날 시간을 0으로 설정해줘야 한다. 이 후, 시스템을 저전력 모드에서 탈출시키기 위한 인터럽트가 필요로 하다. 이것은 STM32L4의 저전력 타이머를 활용한다. 저전력 타이머가 만료되어 인터럽트가 발생하고 이에 따라 시스템은 저전력 모드에서 깨어난다. 이후 시스템을 정상 동작시키기 위해 초기화를 진행한다. 다시 systick 인터럽트를 활성화 시키면, 다시 주기적으로 스케줄러가 호출되기 시작하며 시스템이 동작한다. 여기까지가 그림 3의 저전력 모드를 활용하는 하이브리드 가버너의 기능 동작이다.

## V. 실험

실험 환경은 다음과 같이 구성되어 있다. 개발 IDE로 IAR사의 Embedded workbench를 사용했으며, 개발 보드는 STM32L476VG Discovery를 이용했으며, 전력 소모량 계측을 위해 Monsoon Solution사의 Monsoon power monitor를 사용했다.

1. 실험 시나리오

하이브리드 가버너의 성능을 측정하기 위해, 본 논문에서는 피트니스 제품군의 대표적인 기능 3가지를 실시간 심박수 모니터링 기능, 운동 모니터링 기능, 동기화 기능으로 정의하고, 이들 기능의 주기를 모사한 태스크를 구성하였다. 태스크는 FreeRTOS의 태스크로 구현하였다. 실제 피트니스 제품군 중 하나인 Fitbit을 관찰한 결과, 실시간 심박수 모니터링은 약 1초의 주기, 운동 모니터링 태스크는 2~3초의 주기를 가지고 있었으며, 본 논문에서는 이를 바탕으로 태스크 3개를 각각 주기 1.3초, 2.3초의 주기 태스크와 실험 시간 동안 단 한번 실행되나, 실행 시간이 긴 하나의 태스크로 구현하였다.

2. 실험 결과

그림 4는 상기 실험 시나리오에 대해 하이브리드 가버너를 적용하지 않았을 때, DVFS 기능만 적용했을 때, 하이브리드 가버너 전체 기능을 적용했을 때의 3가지 상황에 대해 60초간의 누적 전력 소모량을 비교한 그래프이다.

60초간의 누적 전력 소모량은 하이브리드 가버너를 적용하지 않았을 때 5.99mWh의 전력 소모량으로 가장 높았고, DVFS 기능만 적용했을 시 5.32mWh, 마지막으로 본 논문에서 제안하는 하이브리드 가버너의 모든 기능을 적용했을 때 4.29mWh로 가장 낮았다. 전력 관리 기능을 아무것도 사용하지 않았을 때에는, 실제 의미 있는 작업을 하는 태스크가 없더라도 운영체제에서 idle 태스크를 실행하기 때문에, 항상 CPU가 활성화되어 있는 상태가 유지되어 전력 소모량이 크다. 그에 비해 하이브리드 가버너의 DVFS 기능을 활용하면, 운영체

제에서 실행하는 idle 태스크를 낮은 CPU 동작 주파수에서 실행시키기 때문에 전력 소모량이 줄어든다. 여기에 하드웨어에서 지원하는 저전력 기능을 더 적극적으로 활용하면, 전력 소모량을 더욱 크게 줄일 수 있다.

전력 관리 기능이 아무것도 없을 때 보다 DVFS를 적용했을 때 약 12%의 전력 소모량을 줄일 수 있었고, 하이브리드 가버너의 전력 관리 기능을 모두 사용했을 때 약 30%의 전력을 아낄 수 있었다. 또한, DVFS만을 적용하는 것 보다 하드웨어에서 지원하는 저전력 모드를 활용할 때, 약 20%의 전력 소모량을 더 줄일 수 있었다. 표 1은 상기의 내용들을 정리한 표이다.

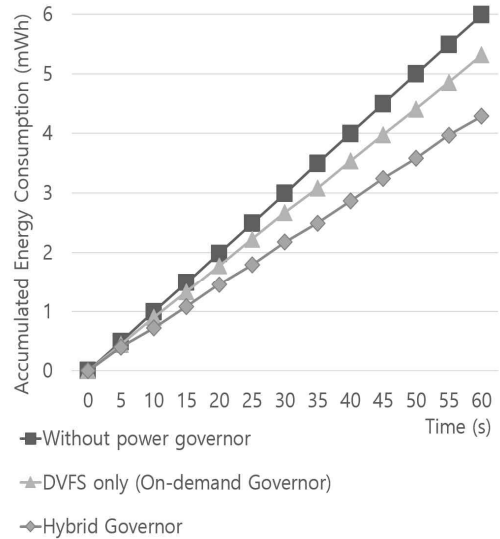


그림 4. 기능 적용 여부 별 누적 소모 전력  
Fig. 4 Power consumption of Hybrid Governor

표 1. 하이브리드 가버너 기능 적용에 따른 누적 전력 소모량 및 기기 가용 시간  
Table 1. Power consumption and device usage time of Hybrid Governor

	Power Consumption (mWh)	Battery life time <sup>1)</sup>	Improvements
w/o power governor	5.99mWh	13.65 hrs	-
DVFS only	5.32mWh	15.37 hrs	12% <sup>2)</sup>
Hybrid Governor	4.29mWh	19.02 hrs	30% <sup>2)</sup> , 20% <sup>3)</sup>

1) 1000mAh  
2) 하이브리드 가버너 미적용 대비  
3) DVFS 적용 대비

## VI. 결 론

본 논문에서 제안한 FreeRTOS 및 STM32L476 VG Discovery 보드를 위한 하이브리드 가버너는 다양한 시스템에서 널리 쓰이는 DVFS를 활용하는 Ondemand 가버너에 STM32L476VG Discovery 보드에서 지원하는 Sleep, Low power sleep, Stop0, Stop1, Stop2의 다양한 저전력 모드를 추가로 활용하는 새로운 전력 관리 가버너이다. 기존 상용 웨어러블 운영체제인 타이젠, 안드로이드 웨어의 경우 단순히 타임아웃 기법을 활용해 하드웨어 저전력 모드를 활용하지만, 본 논문에서 제안한 하이브리드 가버너의 경우, 웨어러블 기기의 작업들이 주기적이라는 특성을 활용해 운영체제 수준에서 스케줄러를 통해 현재 실행되고 있는 태스크들의 상태를 반영해 적절한 저전력 모드를 활용하였다. 따라서 기존의 타임 아웃을 활용한 모델에 비해 더 자주, 정확히 저전력 모드를 활용해 성능이 더욱 뛰어난 전력 관리 기법을 선보였다.

본 논문에서 제안한 하이브리드 가버너는 실험을 통해 전력 관리 기법을 적용하지 않은 경우에 비해 최대 30%, Ondemand 가버너만 적용한 경우에 비해 20% 소모 전력을 줄였음을 보였다.

## References

- [1] Joseph Wei, "How Wearables Intersect with the Cloud and the Internet of Things: Considerations for the Developers of Wearables," IEEE Consumer Electronics Magazine, Vol. 3, No. 3, pp. 53-56, 2014.
- [2] F. Gemperle, C. Kasabach, J. Stivoric, M. Bauer, R. Martin, "Design for Wearability," Proceedings of Second International Symposium on Wearable Computers, pp. 116-122, 1998.
- [3] T. Starner, "The Challenges of Wearable Computing: Part 1," in IEEE Micro, vol. 21, no. 4, pp. 44-52, 2001.
- [4] S. K. Datta, C. Bonnet, N. Nikaein, "Android Power Management: Current and Future Trends," Proceedings of the First IEEE Workshop on Enabling Technologies for Smartphone and Internet of Things (ETSIoT), pp. 48-53, 2012.
- [5] B.T. Oh, K.H. Park, B.G. Lee, S.Y. Noh, J.S. Ik, "Energy Awareness & Energy Saving Techniques for IT," Communications of the Korean Institute of Information Scientists and Engineers, Vol. 30, No. 7, pp.17-26, 2012 (in Korean).
- [6] STM32L4 Reference Manual
- [7] V. Pallipadi, A. Starikovskiy, "The Ondemand Governor," Proceedings of Linux Symposium, Vol. 2, No. 00216, pp. 215-230, 2006.
- [8] J.H. On, B.S. An, B.G. Lee, S.M. Jang, S.I. Jun, "An Energy Efficiency comparison Analysis of CPU DVFS Governor in Virtualized Environments," Proceedings of Korea Information Science Society, pp. 1239-1241, 2016 (in Korean).
- [9] "Tizne, Device Settings and Systems, Power States", Available: <https://developer.tizen.org/development/guides/web-application/device-settings-and-systems/power-states>.
- [10] S.T. Kim, H.J. Park, H.J. Park, D.K. Woo, "Core Technology for Ultra Low Power Using Cold Restart in Wearable Devices," Journal of the Institute of Electronics and Information Engineers, Vol. 54, No. 4, pp. 44-49, 2017 (in Korean).
- [11] "STM32L4 Datasheet", Available: [http://www.st.com/content/st\\_com/en/products/microcontrollers/stm32-32-bit-arm-cortex-mcus/stm32-ultra-low-power-mcus/stm32l4-series/stm32l4x6/stm32l476vg.html](http://www.st.com/content/st_com/en/products/microcontrollers/stm32-32-bit-arm-cortex-mcus/stm32-ultra-low-power-mcus/stm32l4-series/stm32l4x6/stm32l476vg.html)
- [12] "STM32L4 Programming Manual", Available: [http://www.st.com/content/st\\_com/en/products/microcontrollers/stm32-32-bit-arm-cortex-mcus/stm32-ultra-low-power-mcus/stm32l4-series/stm32l4x6/stm32l476vg.html](http://www.st.com/content/st_com/en/products/microcontrollers/stm32-32-bit-arm-cortex-mcus/stm32-ultra-low-power-mcus/stm32l4-series/stm32l4x6/stm32l476vg.html)
- [13] "STM32L4 Power Control", Available: [http://www.st.com/content/st\\_com/en/products/microcontrollers/stm32-32-bit-arm-cortex-mcus/stm32-ultra-low-power-mcus/stm32l4-series/stm32l4x6/stm32l476vg.html](http://www.st.com/content/st_com/en/products/microcontrollers/stm32-32-bit-arm-cortex-mcus/stm32-ultra-low-power-mcus/stm32l4-series/stm32l4x6/stm32l476vg.html)
- [14] "FreeRTOS about FreeRTOS", Available: <https://freertos.org/RTOS.html>
- [15] "FreeRTOS Low Power Support", Available: <https://freertos.org/low-power-tickless-rtos.html>
- [16] K. Choi, "Dynamic Voltage and Frequency Scaling for Energy-efficient System Design," University of Southern California, 2005.

**SungYup Lee (이 성 업)**

He is received B.S. and M.S. degree in computer science and engineering from Chungnam National University in 2016 and 2018. respectively. He

is currently Ph.D. student in University of Science & Technology.

Email: sylee549@gmail.com

**Hyungshin Kim (김 형 신)**

He is received his B.S. and Ph.D. degrees in computer science, in 1990 and 2003, respectively, from Korea Advanced Institute of Science and Technology

(KAIST). He received the M.Sc. degree in satellite communication engineering from University of Surrey, UK in 1990. He joined Chungnam National University in 2004 and he is a Professor in the department of computer science and engineering. His research interests include avionics system, energy-aware computing, and embedded system software.

Email: hyungshin@cnu.ac.kr