모바일 시스템 프로그래밍

10 Energy-efficient Mobile Sensing System Design 4

2017 1학기

강승우

Adaptive Duty Cycling Implementation Example

- Duty Cycling 적용했던 이전 예제를 기반으로 Adaptive Duty Cycling을 적용하는 예제
 - 10초 주기로 1초간의 데이터를 관찰
 - 움직임 있음 → 5초로 duty cycle 감소
 - 움직임 없음 → +5초로 duty cycle 증가 / 최대 30초까지
 - (10초), 15초, 20초, 25초, 30초
- 예제 프로젝트 이름
 - MSP16ADCStepMonitor
- 동작
 - Active time 동안 수집된 가속도 데이터를 처리하여 움직임 여부 판단
 - 움직임 여부에 따라 다음 Alarm 시간 설정
 - Activity 화면에 움직임 여부 표시

Example Application

MainActivity

- 버튼 이벤트 처리
- Broadcast 수신
- 움직임 여부 표시

시작/종료 (startService / stopService)

움직임 여부 Broadcast 전송

가속도 데이터 업데이트 시 rms를 계산하여 기준값을 넘는지 확인

• 가속도 rms 값이 기준값을 넘는 횟수 비율을 계산하여 최종 판단 (50%)

ADCMonitorService

- 최초 Alarm (period 10초) 등록
- Alarm 발생 시 wakelock 획득
- 1초 간 가속도 센싱 처리
- 움직임 확인 후 다음 alarm 시간

 설정
 • setNextAlarm 함수
- wakelock 해제
 - onStart / isMoving

StepMonitor (SensorEventListener)

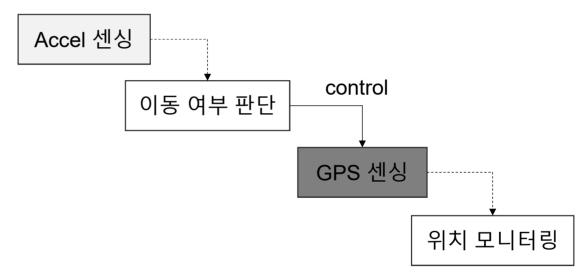
- SensorEventListener 등록 / 등록 해제
- 가속도 센서 데이터 수신 및 rms 계산
- rms 값을 바탕으로 움직임 여부 ___판단

Code

- 소스 코드를 봅시다
 - MSP16ADCStepMonitor

Hierarchical Sensing Implementation Example

• GPS를 이용한 Location Tracking을 Hierarchical Sensing 방법을 적용하여 수행하는 예제



- 예제 프로젝트 이름
 - MSP17HSLocationTracking
- 이동 여부 판단 로직
 - 앞의 예제 코드 이용

Example Application

MainActivity

- 버튼 이벤트 처리
- Broadcast 수신
- 움직임 여부 표시
- 위치 데이터 표시

시작/종료 (startService / stopService)

움직임 여부 / 위치 데이터 Broadcast 전송

- 가속도 데이터 업데이트 시 rms를 계산하여 기준값을 넘는지 확인
- 가속도 rms 값이 기준값을 넘는 횟수 비율을 계산하여 최종 판단 (50%)

HSMonitorService

- 최초 Alarm (period 10초) 등록
- Alarm 발생 시 wakelock 획득
- 1초 간 가속도 센싱 처리
- 움직임 확인 후 GPS location update 요청 제어
- 움직임에 따라 다음 alarm 시간 설 정
- wakelock 해제
 - onStart / onStop

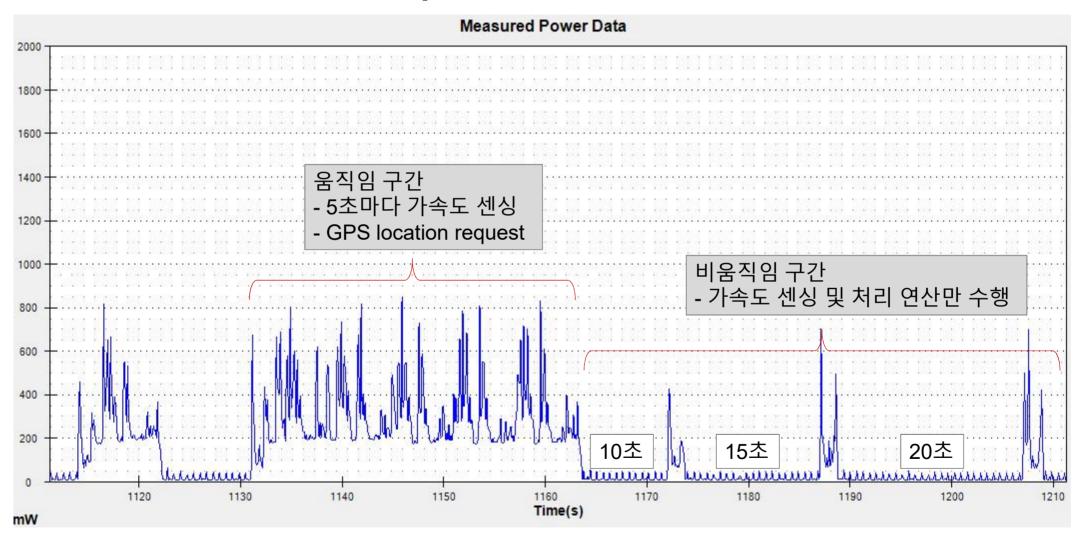
isMoving

StepMonitor (SensorEventListener)

- SensorEventListener 등록 / 등록 해제
- 가속도 센서 데이터 수신 및 rms 계산
- rms 값을 바탕으로 움직임 여부 판
 단

- requestLocation 함수
- cancelLocationRequest 함수
- LocationListener 구현

Power Consumption (Nexus 5 테스트 결과)



Code

- 소스 코드를 봅시다
 - MSP17HSLocationTracking