

PaceTry 센서 시스템 문서

목표

사용자의 실제 보행 속도를 측정하여 도착 시간을 정확하게 예측한다.

이를 위해 확인해야 할 것:

1. 걷는 시간:

사용자가 실제로 걸은 시간 (대중교통, 신호 대기 제외)
2. 이동 거리:

사용자가 걸어서 이동한 거리
3. 보행 속도:

이동 거리 ÷ 걷는 시간

1. 기획 의도

해결하려는 문제

GPS만으로 "걷는 중"을 판단하면 **오판이 발생**한다.

상황	GPS가 보는 것	실제 상태	문제
버스 탑승 중	빠르게 이동 중	앉아있음	"걷는 중"으로 오판
신호 대기 중	정지	서있음	정상
실내 이동	위치가 튼	걷는 중	"정지"로 오판
지하철 탑승	GPS 신호 없음	앉아있음	판단 불가

핵심 문제

대중교통을 타면 몸은 멈춰있지만, GPS는 "이동 중"이라고 판단한다.

이 상태에서 속도를 계산하면:

- 버스로 이동한 거리가 "걸은 거리"에 포함됨
- 보행 속도가 비정상적으로 빨라짐 (시속 30km/h 등)
- 도착 예정 시간 예측이 완전히 틀어짐

해결책: "걷는 시간"과 "멈춘 시간"을 분리

만보기(Step Counter)를 핵심 판단 기준으로 사용

상황	만보기	GPS	최종 판정
버스 탑승 중	✗ 걸음 없음	이동 중	정지
신호 대기 중	✗ 걸음 없음	정지	정지
실내 걷는 중	✓ 걸음 감지	위치 튼	걷는 중

상황	만보기	GPS	최종 판정
야외 걷는 중	<input checked="" type="checkbox"/> 걸음 감지	이동 중	걷는 중

→ GPS가 아무리 빠르게 이동해도, 걸음이 없으면 "걷는 시간"에서 제외

2. 왜 만보기를 1순위로 선택했나?

테스트 결과

GPS, 가속도계, 만보기 각각을 1순위로 설정하여 테스트했다.

1순위 센서	결과	문제점
GPS	<input checked="" type="checkbox"/> 실패	대중교통 탑승 시 "걷는 중"으로 오판
가속도계	<input checked="" type="checkbox"/> 실패	버스 흔들림을 "걷는 중"으로 오판, 민감도 조절 어려움
만보기	<input checked="" type="checkbox"/> 성공	실제 걸음만 정확하게 감지

각 센서의 한계

GPS를 1순위로 했을 때

- 버스/지하철 탑승 중에도 "이동 중"으로 판단
- 실내에서 위치가 튕면서 "걷는 중"으로 오판
- 정지 상태에서도 GPS 드리프트로 미세하게 이동 감지

가속도계를 1순위로 했을 때

- 버스가 흔들릴 때 "걷는 중"으로 오판
- 손에 들고 있으면 작은 움직임에도 반응
- 임계값 조절이 어려움 (너무 높으면 걷기 미감지, 너무 낮으면 오감지)

만보기를 1순위로 했을 때

- 실제 걸음만 감지 (버스 탑승 중에는 반응 안 함)
- 하드웨어 기반이라 오차가 적음
- 단점: 첫 값까지 15초 대기 필요 → 워밍업으로 해결

결론

"걸고 있다"를 판단하는 데 가장 정확한 센서는 만보기

3. 센서별 역할과 확인 사항

센서가 확인하는 것

센서	확인하는 것	사용 목적
만보기	걸음 수	"지금 걷고 있는가?" 판단

센서	확인하는 것	사용 목적
가속도계	흔들림 정도	만보기 백업용
GPS	현재 위치	위치 추적 (이동 거리 계산은 TMap 사용)

각 센서로 계산하는 것

만보기 → 걷는 시간 계산

최근 3초간 걸음 수 ≥ 1 보 → "걷는 중"
"걷는 중" 상태의 시간을 누적 → 걷는 시간

GPS → 현재 위치 확인

1초마다 위도/경도 수신
→ 지도에 현재 위치 표시
→ 경로 이탈 여부 확인

TMap API → 이동 거리

출발지 ~ 도착지 경로 검색
→ 도보 구간 거리 제공 (예: 1.2km)
→ 이 값을 "이동 거리"로 사용

최종 속도 계산

실제 보행 속도 = TMap 도보 거리 ÷ 걷는 시간

예시:

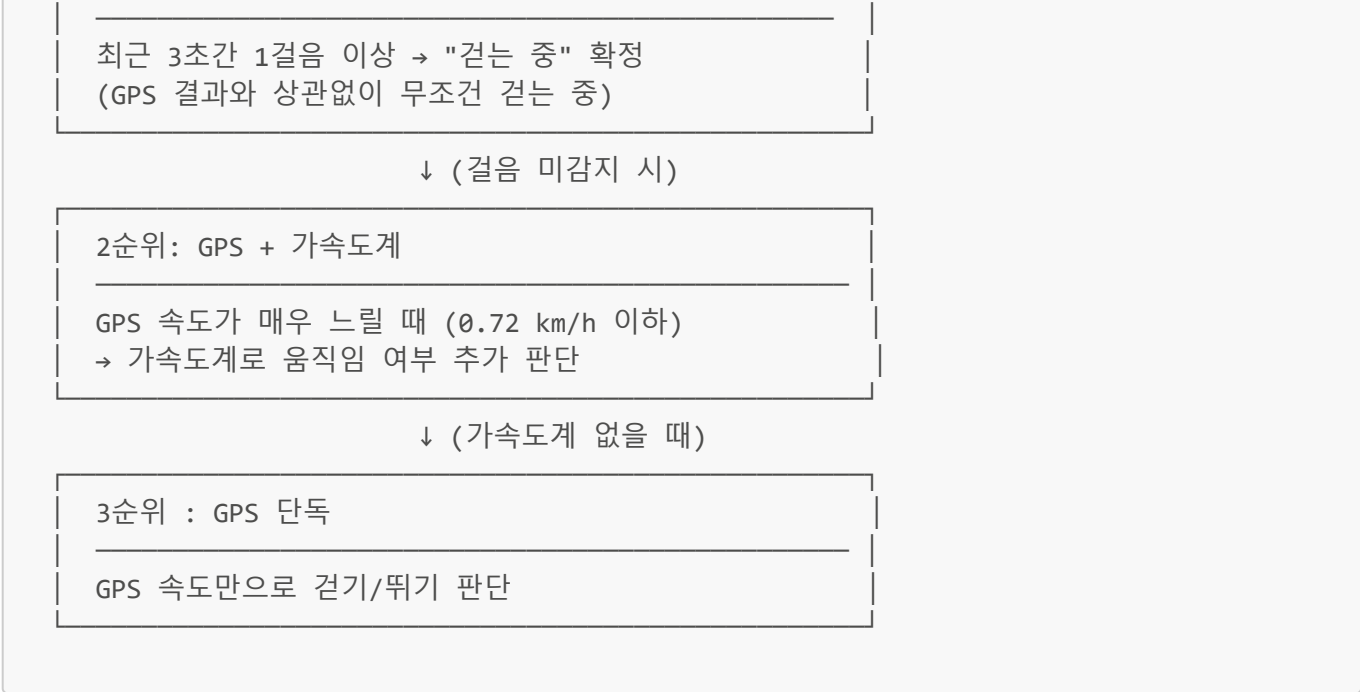
- TMap 도보 거리: 1,200m
- 걷는 시간: 900초 (15분)
- 보행 속도: $1,200 \div 900 = 1.33 \text{ m/s}$ (4.8 km/h)

4. "걷는 중" 판단 방법

3단계 우선순위

상위 단계에서 판정되면, 하위 단계는 실행되지 않는다.

1순위: 만보기 (Step Counter)



판정 기준표 (포어그라운드 - TypeScript)

우선순위	조건	결과
1순위	최근 3초간 1걸음 이상	✔ 걷는 중 (확정)
2순위	걸음 미감지 + GPS 느림 + 가속도 변화 있음	✔ 걷는 중
2순위	걸음 미감지 + GPS 느림 + 가속도 변화 없음	✖ 정지
3순위	걸음 미감지 + GPS 속도 < 7.2 km/h	✔ 걷는 중
3순위	걸음 미감지 + GPS 속도 ≥ 7.2 km/h	🏃 뛰는 중

판정 기준표 (백그라운드 - Kotlin)

조건	결과
최근 3초간 1걸음 이상	✔ 걷는 중 (walking)
최근 3초간 0걸음	✖ 정지 (paused)

참고: 백그라운드에서는 **running** 상태가 없다. 오직 **walking** / **paused** 두 가지만 존재한다.

5. 속도 계산 방법

공식

실제 보행 속도 = 이동 거리 ÷ 걷는 시간

항목	출처	이유
----	----	----

항목	출처	이유
이동 거리	TMap 계획 경로 거리	GPS 측정은 오차가 큼
걸는 시간	만보기 판정 시간 합산	대중교통/신호 대기 제외

계산 예시

[상황] 집 → 학교 경로 (버스 환승 포함)

TMap 정보:

- 전체 거리: 3.5km
- 도보 구간: 1.2km (집→버스정류장 0.5km + 정류장→학교 0.7km)

실제 이동:

- 전체 소요 시간: 25분
- 버스 탑승 시간: 10분
- 신호 대기 시간: 2분
- 걷는 시간: 13분 (780초) ← 만보기로 측정

계산:

- 보행 속도 = 1,200m ÷ 780초 = 1.54 m/s (5.5 km/h)

왜 GPS 측정 거리를 안 쓰나?

문제	설명
GPS 드리프트	정지해도 좌표가 튀어서 거리가 늘어남
실내 오차	신호 불량으로 위치가 부정확
경로 이탈	실제와 다른 거리가 계산됨

6. 센서 워밍업

문제

만보기는 앱 시작 후 첫 값이 나오기까지 약 17~18초가 걸릴 수 있다.

해결책

경로 안내 시작 전에 센서 워밍업을 진행한다.

- 앱 시작 시 만보기를 미리 시작
- 첫 걸음 감지 시 워업 완료 처리
- 별도의 타임아웃 없이 센서가 준비되면 바로 사용 가능

7. 주요 설정값

만보기 (Step Counter) — 1순위

항목	값	의미
걸음 판정 시간	3초	최근 3초간의 걸음 수로 판단
최소 걸음 수	1보	3초간 1걸음 이상이면 "걷는 중"

GPS — 2~3순위 (만보기 미감지 시에만 사용)

항목	값	의미
최소 속도	0.72 km/h	이하면 "정지"로 간주
최대 속도	13 km/h	이상이면 비정상 (차량 등)
뛰기 속도	7.2 km/h	이상이면 "뛰는 중"으로 판단
업데이트 주기	1초	1초마다 위치 수신

GPS 속도를 함께 사용하는 이유

만보기가 "걷고 있는가?"를 판단하는 1순위이다.

포어그라운드 (TypeScript):

- 걸음 감지 → 무조건 "걷는 중" (GPS 속도 무시)

백그라운드 (Kotlin SensorService):

- 걸음 감지 → 무조건 "걷는 중" (GPS 속도 무시)
- `running` 상태 없음, `walking` / `paused` 두 가지만 존재

GPS 속도는 "걷기/뛰기 구분"이 아니라 **거리 누적 필터링**에 사용된다:

- `walking` 상태 + GPS 속도 < 13 km/h → 거리 누적
- `walking` 상태 + GPS 속도 ≥ 13 km/h → 거리 누적 안 함 (차량 오감지 방지)

8. 성능 평가 지표

두 가지 정확도를 분리한 이유

지표	측정 대상	영향 요인
전체 시간 정확도	출발~도착 전체	보행 + 대중교통 지연 + 신호 대기
보행 시간 정확도	순수 걸은 시간	보행 속도만

전체 시간 정확도는 **버스 지연, 신호 대기** 등 외부 요인에 영향받는다. **보행 시간 정확도**가 순수하게 알고리즘 성능을 평가한다.

9. 관련 파일

TypeScript (포어그라운드)

파일	역할
<code>movementTrackingService.ts</code>	센서 통합 + 상태 판정 + 속도 계산
<code>nativeSensorService.ts</code>	안드로이드 네이티브 센서 연동
<code>locationService.ts</code>	GPS 위치 추적
<code>backgroundLocationTask.ts</code>	백그라운드 GPS 수집

Kotlin (백그라운드 네이티브)

파일	역할
<code>SensorService.kt</code>	백그라운드 센서 서비스 (만보기 + GPS + 가속도계)
<code>SensorServiceModule.kt</code>	React Native ↔ Kotlin 브릿지

문서 작성일: 2025-12-04