

# GPS + 칼만 필터 기반 헬스케어 게이미피케이션 플랫폼

---

연세대학교 미래\_김호중

## 칼만 필터 프로젝트

### 왜 하면 좋은가?

1. 헬스케어 도메인을 살리면서 차이점이 부각된다.

» 취업을 위해서는 명확한 차이점이 존재 해야 한다. 즉 뻔한 포트폴리오는 매력이 크게 없다.

뻔한 사람들과의 차별점이 될 수 있는 부분은 “무언가를 직접만드는” 사람이 되는 것이다.

2. 데이터를 직접 만들어 내고 다룰 수 있다.

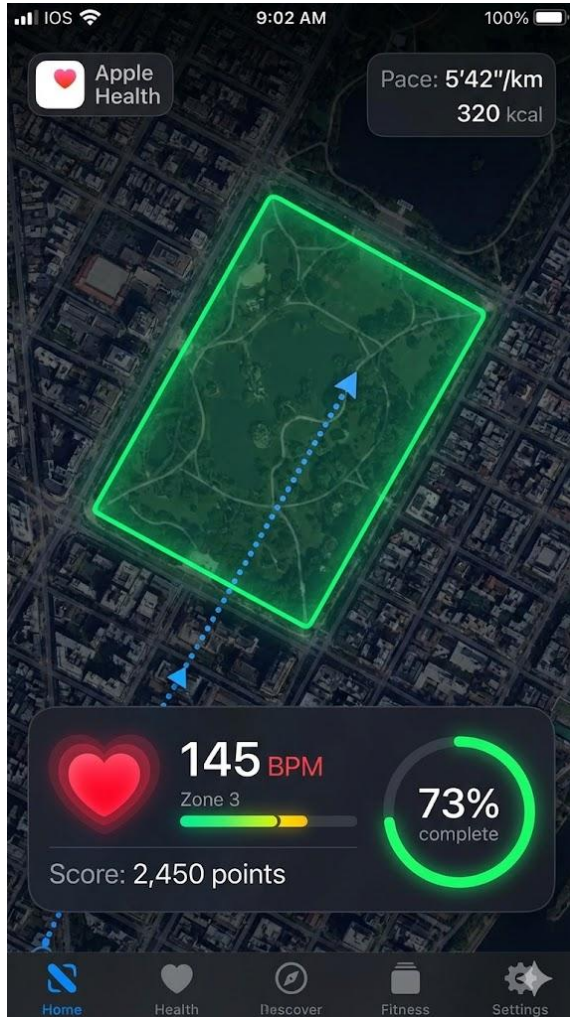
» 민감 정보로 취급되는 인간의 신체 데이터를 직접 수집하고 처리가능하다.

기기의 제작, 소프트웨어 제작 , 데이터 관리 및 분석 까지 이어지는 프로세스를 하나의 프로젝트에서 가능하게한다.

3. 만들기가 어렵다.

얼마나 어려운지도 모르겠다. 처음 해본다. 수학 엄청 잘해야 한다. 가우시안기반 해석은 아직도 모르겠다.

## 프로젝트 예상 결과



### 영역 감싸기

이거 만들려면 GPS사야한다.

지도 위 특정 영역(폴리곤)을 운동으로 "정복"

예: 한강공원 산책로, 아파트 단지, 동네 블록

완주 판정: Point-in-Polygon 알고리즘 + 경로 커버리지 계산

### 심박 기반 동적 보상

이거 만들려면 심박수 센서 사야한다.

같은 거리라도 운동 강도에 따라 점수 차등

Zone 1-2 (걷기): 기본 점수  $\times 1.0$

Zone 3-4 (조깅/달리기): 점수  $\times 1.5 \sim \times 2.0$

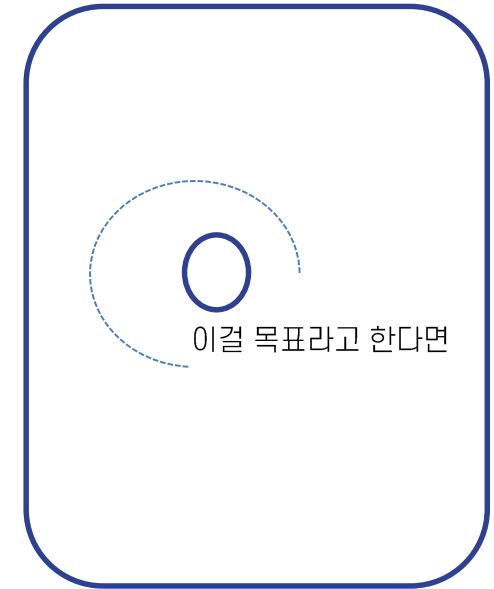
### 실시간 피드백 & 게이미피케이션

현재 심박존, 누적 점수, 예상 칼로리 실시간 표시

영역 완주까지 남은 거리/시간 안내

일간/주간 리더보드, 친구와 경쟁

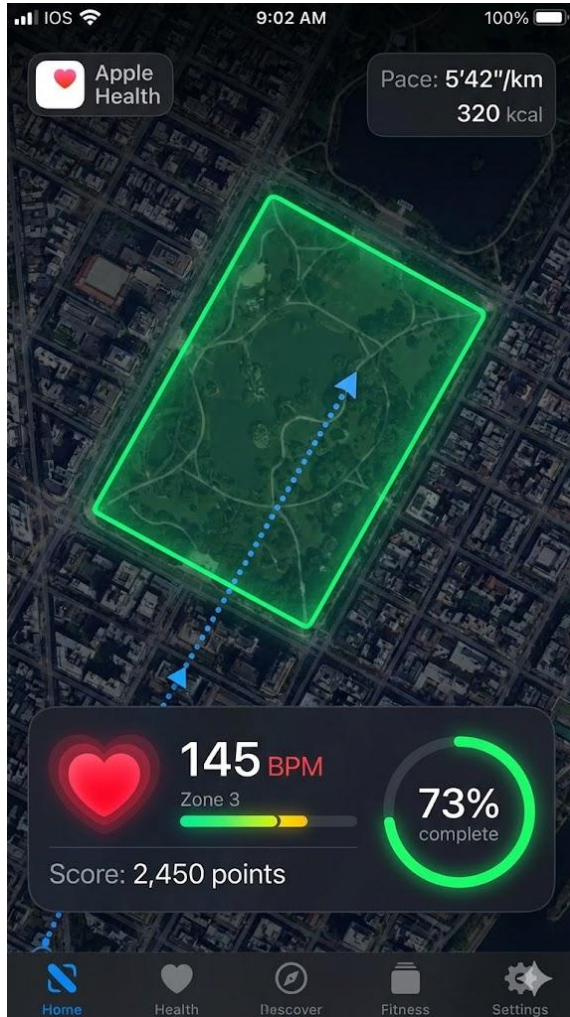
이게 좀 고민인데 데이터수집과 처리에 집중할지 or 트랙잭션 만들지 골라야한다.



이걸 공원이라 하고

점선처럼 휘감아서 점수를 얻는다

## 프로젝트 예상 결과



이걸 목표라고 한다면

이걸 공원이라 하고  
점선처럼 휘감아서 점수를 얻는다

## 영역 감사가- 칼만필터

이걸 그냥하면 정확도가 구리다 그래서 칼만필터라는 것을 쓴다.  
대충 수학 기반 추정이다.

$$x = [p_x, p_y, v_x, v_y, \theta]^T$$

위치가 상태벡터 라는 애인데 애를 사용해서 상태를 추정한다.

$$\hat{x}_{k|k-1} = f(\hat{x}_{k-1}, u_k)$$

$$\hat{x}_k = \hat{x}_{k|k-1} + K_k(z_k - H\hat{x}_{k|k-1})$$

좀 쉽게 풀어서 설명하면  
현재 나의 상태(t)는 이전상태(t-1)와 이전상태에서의 동작으로  
추정가능하다 라는 말이다.

더 알고 싶다면 히든 마르코프 모델과 베이지안 필터  
칼만필터를 공부해야한다. 나도 잘 모른다. 어렵다 매우

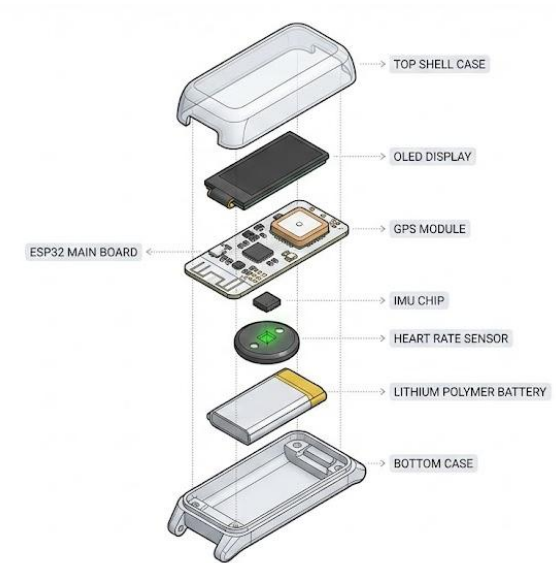
## 웨어러블 기기

우리가 선택할 수 있는 선택지가 대충 4개가 있다.

1. 기성품 웨어러블 + API 연동
  2. 스마트폰 온리, 내장 gps랑 가속도 카메라 써서
  3. 최소 하드웨어 + 시뮬레이터
  4. 하드웨어 덕지덕지
- » 4. 하드웨어 덕지덕지 이게 가장 이상적이지만 돈도 들고 하드웨어 파트에서 얼마나 시간 쏟을지 모른다.

GPS만 하면 5만원 안쪽 심장박동과 이것저것 하면 15만원으로 구매 가능하다

이걸 하고자 하는 이유는 센서데이터 직접 획득과 칼만필터 적용을 위해서



매우 이상적인 예상도

현실은 무언가 전자기기와 덕지덕지된 센서를 예상한다.

데이터 살펴보기

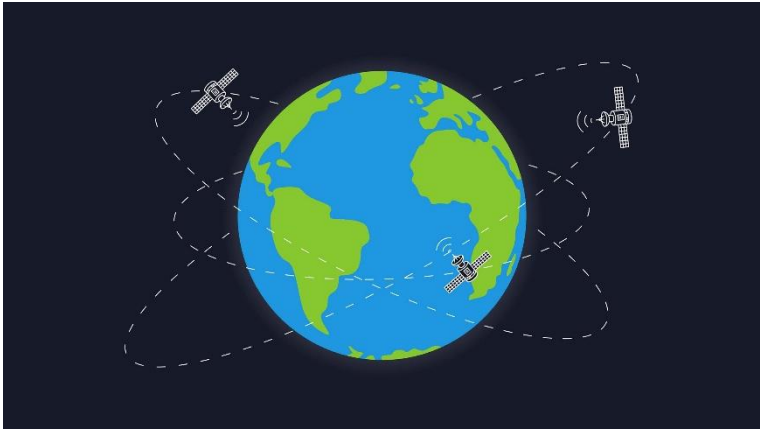
NMEA GPS 모듈은 위성에서 신호를 받아서 현재 위치를 계산한 다음, 그 결과를 문자열로 보낸다 아래가 그 예시

```
>> $GPGGA,092750.000,3723.4657,N,12202.2694,W,1,8,1.03,61.7,M,-21.3,M,,*5C
$GPRMC,092750.000,A,3723.4657,N,12202.2694,W,0.02,31.66,280423,,,A*43
```

즉 우리가 받을 수 있는 GPS 데이터는 위꼴이다. 이걸 DB에서 처리해야한다. 이후 API와 연동할 수 있게

```
$GPRMC,092750.000,A,3723.4657,N,12202.2694,W,0.02,31.66,280423,,,A*43
```

시간	유효	위도	경도	속도	방향	날짜
09:27:50		37도23.4657N	122도02.2694W	(노트)		
읽는 방법						
보통 1초에 1번 업데이트 (1Hz)						



## 데이터 살펴보기

### MPU-9250은 세 종류의 센서가 하나로 합쳐진 것

가속도계(Accelerometer): 어느 방향으로 얼마나 가속되고 있는지 측정 가만히 있어도 중력 때문에 아래 방향으로 약  $9.8\text{m/s}^2$ 가 측정  
이걸 이용해서 기기가 어느 쪽으로 기울어져 있는지 알 수 있어요.

자이로스코프(Gyroscope): 얼마나 빠르게 회전하고 있는지 측정 단위는 초당 각도( $^\circ/\text{s}$ )예요. 손목을 휙 돌리면 큰 값이 나오고, 가만히 있으면 0에 가까운 값이 측정

자력계(Magnetometer): 지구 자기장을 감지해서 나침반처럼 방향을 알 수 있다.

가속도:  $X=-0.05, Y=0.12, Z=9.81 \text{ (m/s}^2\text{)}$

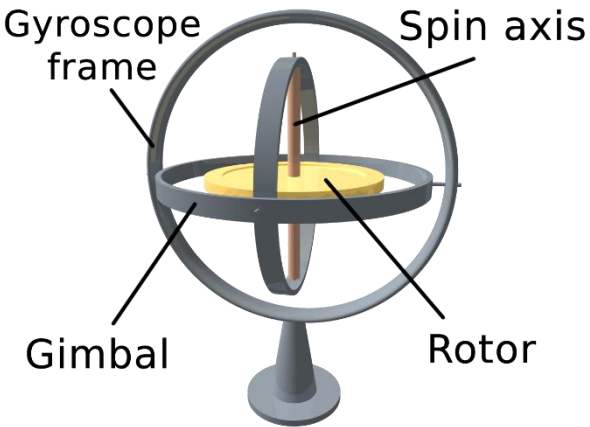
자이로:  $X=0.5, Y=-0.3, Z=0.1 \text{ (}^\circ/\text{s)}$

자력계:  $X=25.3, Y=-12.1, Z=40.5 \text{ (}\mu\text{T)}$

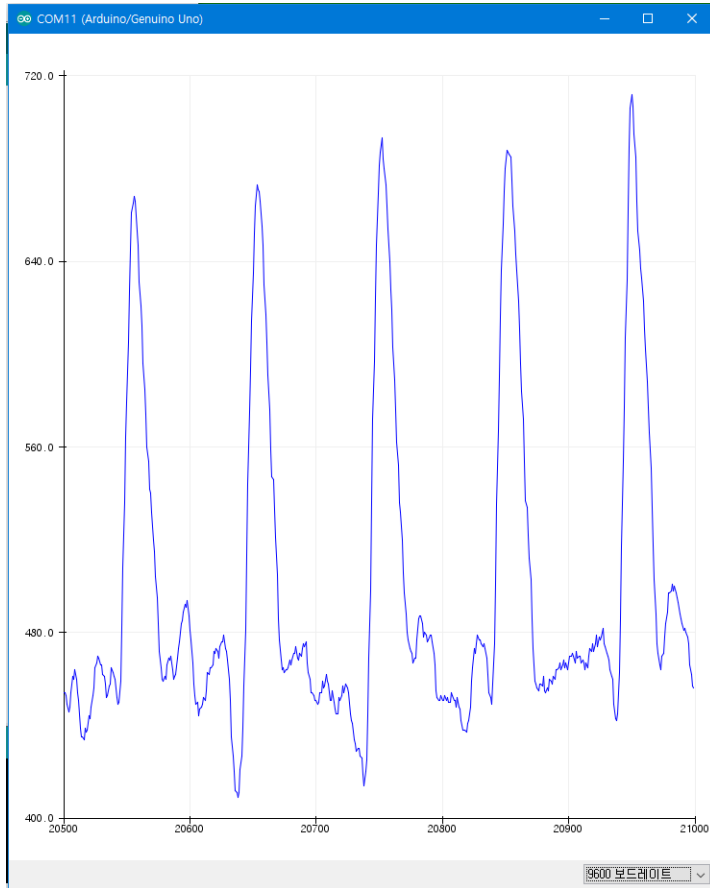
데이터 예시 이걸 벡터로 사용해서 칼만 적용

보통  $100\text{Hz}$ (초당 100번)로 샘플링

» 여기서 보면 샘플링 횟수가 다르기에 어떻게 저장해서 어떻게 연산할지 많은 고민이 필요하다.



## 데이터 살펴보기



MAX30102는 손가락이나 손목에 빛을 쏘고, 반사되어 돌아오는 빛의 양을 측정

IR(적외선)과 Red(빨간빛) 두 채널의 값

» IR: 50213, 50198, 50245, 50312, 50456, 50523, 50489, 50401, 50287, 50234...  
Red: 48123, 48156, 48201, 48267, 48398, 48445, 48412, 48334, 48223, 48167...

시계열 데이터라서 마찬가지로 저장과 관리 처리가 좀 까다롭습니다. 도메인 지식과 연관지어 분석 엔지니어링이 가능하며 그렇게 해야합니다. 헬스케어 학과의 장점을 보여줘야 경쟁력이 생긴다.

샘플링은 보통 50-100Hz

## 중요

우리가 해야 하는 것은 시계열 데이터와 센서 퓨징입니다. 센서를 어떻게 활용해서 알고리즘을 짤 것인지의 싸움 모션에 대한 보정이 반드시 필요하다.



이런것들 필요하다

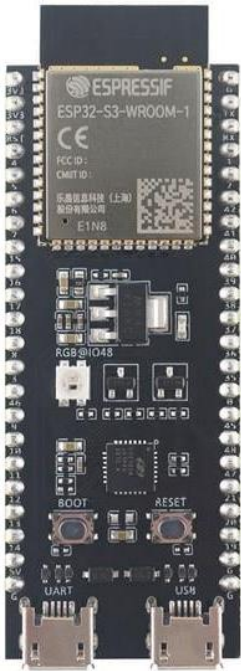
구분	컴포넌트 모델	주요 역할	주요 사양 및 특징
메인 프로세서 (MCU)	ESP32-S3	시스템 제어 및 통신	<ul style="list-style-type: none"><li>Dual Core 240MHz</li><li>BLE 5.0 / Wi-Fi 내장</li><li>FreeRTOS 지원</li></ul>
GPS 모듈	u-blox NEO-M9N	위치 추적	<ul style="list-style-type: none"><li>1.5m CEP 정확도</li><li>GNSS 멀티밴드 수신</li><li>25mW 저전력</li></ul>
IMU 센서	MPU-9250	모션/자세 감지	<ul style="list-style-type: none"><li>9축 (가속도+자이로+지자기)</li><li>200Hz 고속 샘플링</li></ul>
심박 센서	MAX30102	생체 신호 측정	<ul style="list-style-type: none"><li>PPG 방식 (광학식)</li><li>SpO2(산소포화도) 측정</li></ul>
배터리	LiPo 500mAh	전원 공급	<ul style="list-style-type: none"><li>3.7V 리튬 폴리머</li><li>약 8시간 연속 사용</li></ul>

기능 블록	상세 내용	데이터 흐름
센서 데이터 수집	GPS(위치), IMU(자세), HR(심박) 데이터 동기화 및 수집	Sensors → ESP32 (I2C/UART)
신호 처리 (Filter)	칼만 필터 (Kalman Filter) 적용	센서 노이즈 제거 및 위치/자세 추정 오차 보정
연산 (Logic)	스코어 계산 (Score Calc)	수집된 데이터를 기반으로 사용자 활동 점수화
통신 (Stack)	BLE 5.0 Stack	가공된 데이터를 스마트폰으로 전송 (저전력)

대충 이런것들 필요하다 느낌으로 받아들여 주세요.

알리랑 국내랑 가격도 봐야하고 호환이 되는건지 조립이 어려운지 납땜 필요한지 등 알아봐야할 것들이 좀 많아서

## 부품 소개



ESP32-S3

3.3V를 공급받는 구조

3.7V LiPo 배터리를 직접 연결하려면 별도의 전원 관리 회로가 필요

"핀헤더 납땜 완료"인지 확인

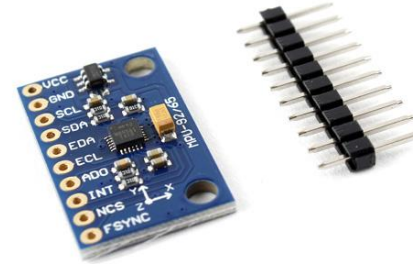


GPS 모듈(NEO-M9N)

34,380원

ESP32-S3에서 UART1이나 UART2를 사용

안테나가 온보드인지, 외장 안테나 연결 단자(U.FL 또는 SMA)가 있는지 확인



IMU(MPU-9250)

MPU-9250은 가속도계, 자이로, 자력계가 모두 들어있는 9축 센서

16,000

연결은 I2C로

기본 주소는 0x68 "9축"이라고 표기된 제품구매

## 부품 소개

심박 센서(MAX30102)

MAX30102는 빛을 쏘고 반사되는 양을 측정해서 심박을 알아내는 방식(PPG)

1700원

배터리  
반드시 "보호회로 내장" 제품



납땜이 필수인가요?

브레드보드 + 점퍼선 방식을 사용하면 납땜 없이도 충분히 진행가능

베이지안 필터

$$p(x_t \mid z_{1:t}, u_{1:t}) = \frac{p(z_t \mid x_t, z_{1:t-1}, u_{1:t}) \cdot p(x_t \mid z_{1:t-1}, u_{1:t})}{p(z_t \mid z_{1:t-1}, u_{1:t})}$$



$$bel(x_t) = \eta \cdot p(z_t \mid x_t, z_{1:t-1}, u_{1:t}) \cdot p(x_t \mid z_{1:t-1}, u_{1:t})$$

분모 = 노말라이저(n)로 정리하면 위와 같이 표현된다.

u\_{1:t}: 처음부터 시간 t까지의 모든 제어 입력



$$bel(x_t) = \eta \cdot p(z_t \mid x_t) \underbrace{p(x_t \mid z_{1:t-1}, u_{1:t})}_{\text{Prediction Term}}$$

마르코프 가정에 의해 한번 더 정리한 꼴

$$p(z_t \mid x_t, z_{1:t-1}, u_{1:t}) = p(z_t \mid x_t)$$

위 식은 마르코프 가정에 의하여 다음과 같은 형태로 변환할 수 있다.  
현재의 관측값 z\_t는 오직 현재의 상태 x\_t에만 의존

$$p(x_{t-1} \mid z_{1:t-1}, u_{1:t-1})$$

u\_t는 시각 t에 가한 제어이므로, 과거인 시각 t-1의 상태 x\_{t-1}에는 영향을 줄 수 없음



$$p(x_{t-1} \mid z_{1:t-1}, u_{1:t-1})$$



$$= bel(x_{t-1})$$
 위와 같이 정리가능

베이지안 필터

$$\overline{bel}(x_t) = \sum_{x_{t-1}} p(x_t \mid x_{t-1}) bel(x_{t-1})$$



$$bel(x_t) = \eta_t p(z_t \mid x_t) \overline{bel}(x_t)$$

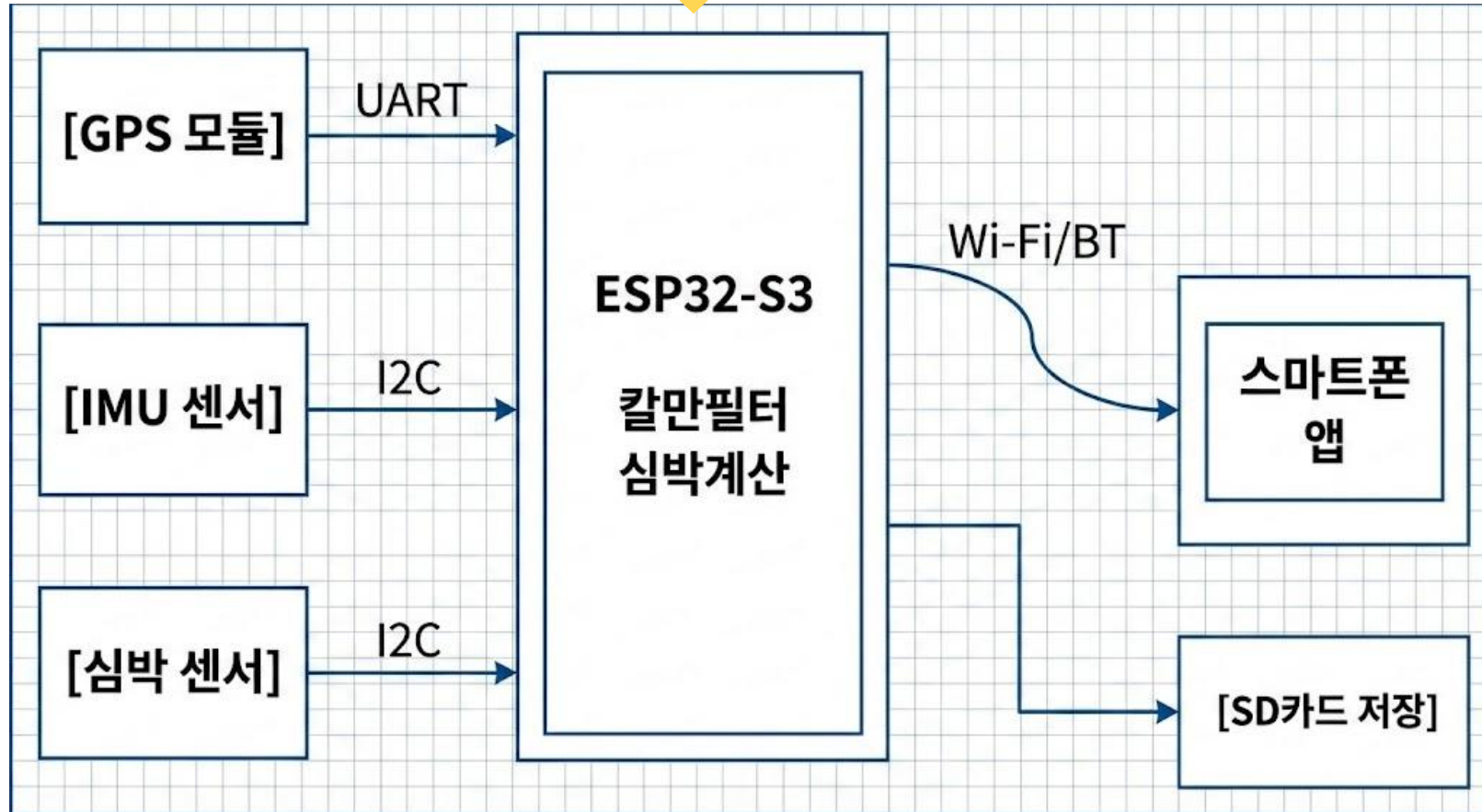
```
1: Algorithm Bayes_filter(bel(x_{t-1}), u_t, z_t):
2:   for all x_t do
3:     bel(x_t) = \int p(x_t \mid u_t, x_{t-1}) bel(x_{t-1}) dx
4:     bel(x_t) = \eta p(z_t \mid x_t) bel(x_t)
5:   endfor
6:   return bel(x_t)
```

```
1: Algorithm Discrete_Bayes_filter({p_{k,t-1}}, u_t, z_t):
2:   for all k do
3:     p_{k,t} = \sum_i p(X_t = x_k \mid u_t, X_{t-1} = x_i) p_{i,t-1}
4:     p_{k,t} = \eta p(z_t \mid X_t = x_k) p_{k,t}
5:   endfor
6:   return {p_{k,t}}
```

알고리즘으로 구현 하면 된다.

도식화

이거까지 하는게 1단계



여기까지가 1단계 입니다.

---