

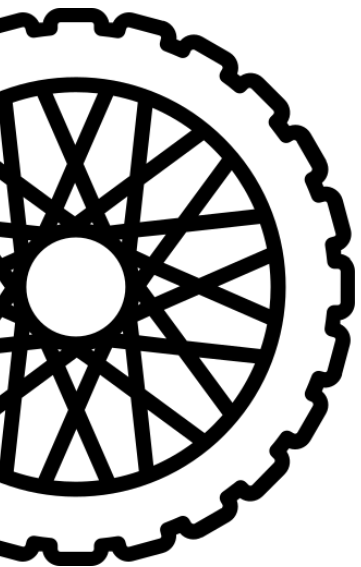
# Rötafy

휠체어 안전 알리미

# 목차

- 1 주제/기술 설명
- 2 데이터/코드
- 3 시연영상
- 4 발전 방향성

# Rōtafy?



Rōta + Notify

바퀴

알리다



# 주제/기술 설명



주행/멈춤

경사로  
/평지

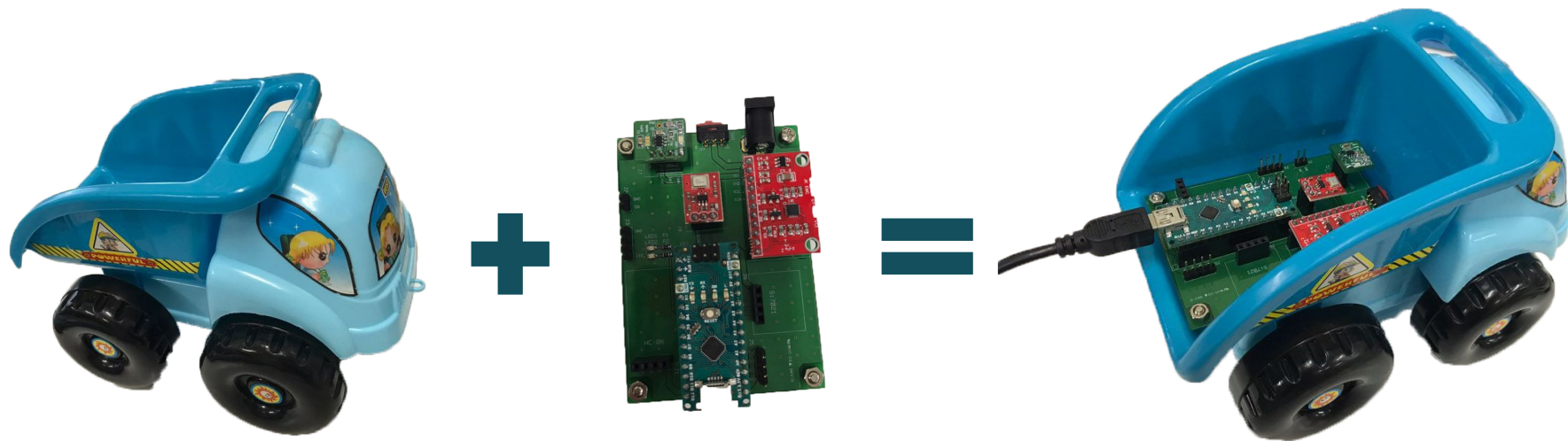
넘어짐

충돌

걸림

도움요청

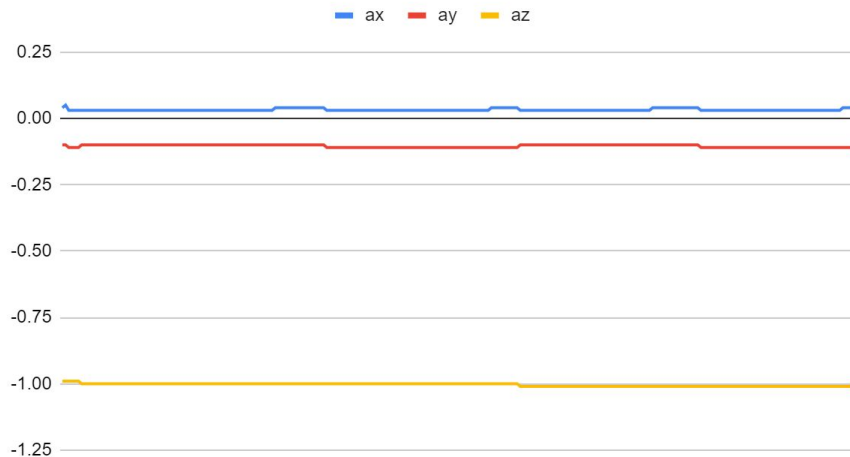
# 휠체어 대체품



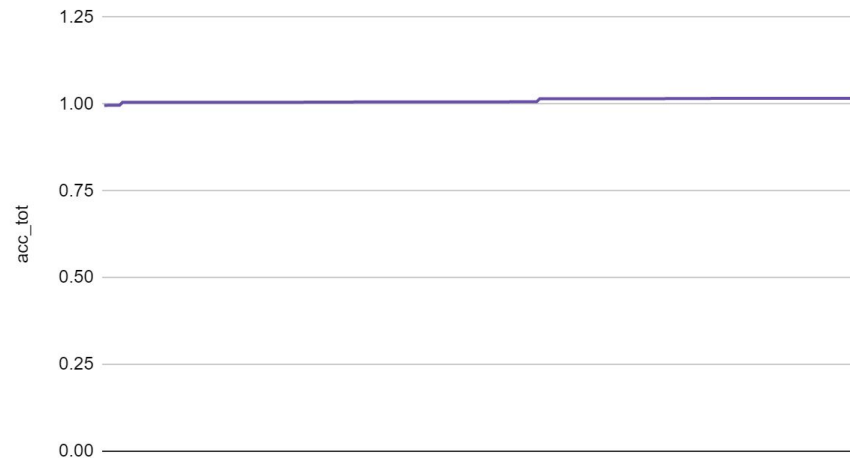
# 데이터/코드 - 멈춤/주행

$$\text{acc\_tot} = \sqrt{ax^2 + ay^2 + az^2}$$

ax, ay, az



acc\_tot



멈춤 MIN=	0.9958413528
멈춤 MAX=	1.016759559

# 데이터/코드 - 멈춤/주행

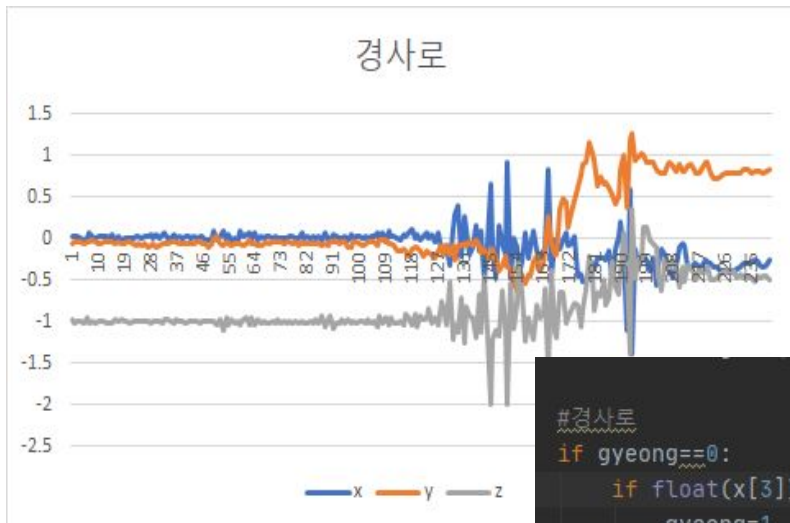
멈춤 MIN=	0.9958413528
멈춤 MAX=	1.016759559

```
# 멈춤
if float(acc_tot) <= 1.02 and float(acc_tot) >= 0.99:
    state = '멈춤'

else:
    state = '주행'
status.configure(text=state)
```



# 데이터/코드 - 경사로



y값의 변화

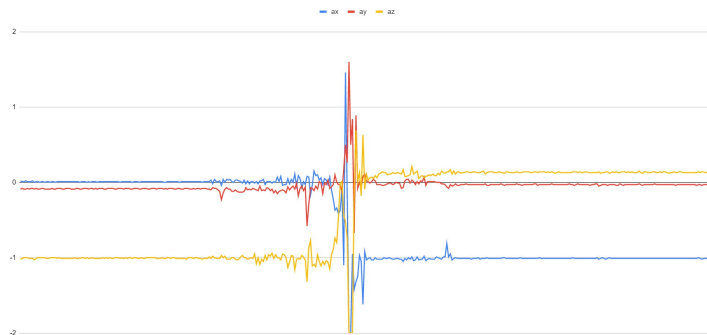
```
#경사로
if gyeong==0:
    if float(x[3]) >= 0.35:
        gyeong=1
        gtime_1=datetime.now()
    else:
        gtime_2=datetime.now()
        gtime=(gtime_2 - gtime_1).total_seconds()
        if gtime>=1:
            status_g.configure(text='경사로', fg="red")
        if float(x[3])<=0.4:
            gyeong=0
            status_g.configure(text='평지', fg="gray")
```





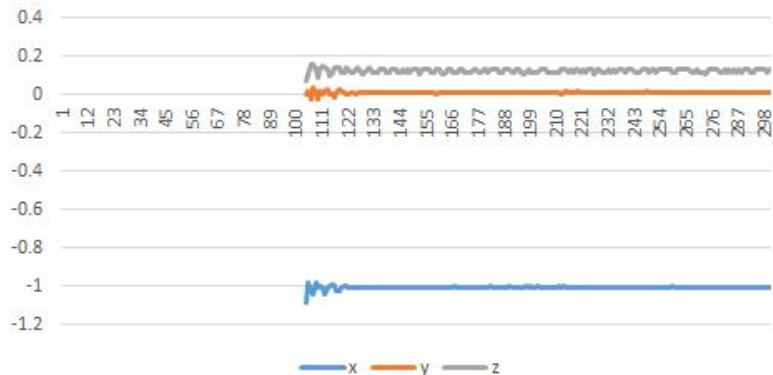
# 데이터/코드 - 넘어짐

넘어짐 그래프

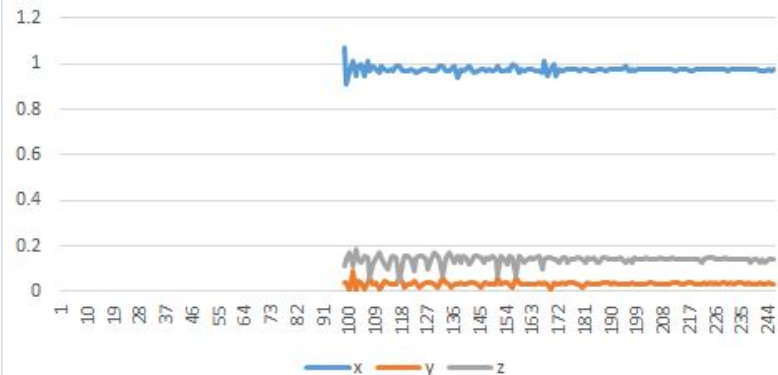


가속도 x와 z값의 변화가 생김  
왼쪽 오른쪽 모두  
두 개의 값으로 넘어짐 구분

왼쪽으로 넘어짐 가속도 변화



오른쪽으로 넘어짐 가속도 변화



# 데이터/코드 - 넘어짐

왼쪽으로 넘어졌을 때

ax min = -1.08

ax max = -0.98

az min = 0.07

az max = 0.16

오른쪽으로 넘어졌을 때

ax min = 0.91

ax max = 1.07

az min = 0.05

az max = 0.19

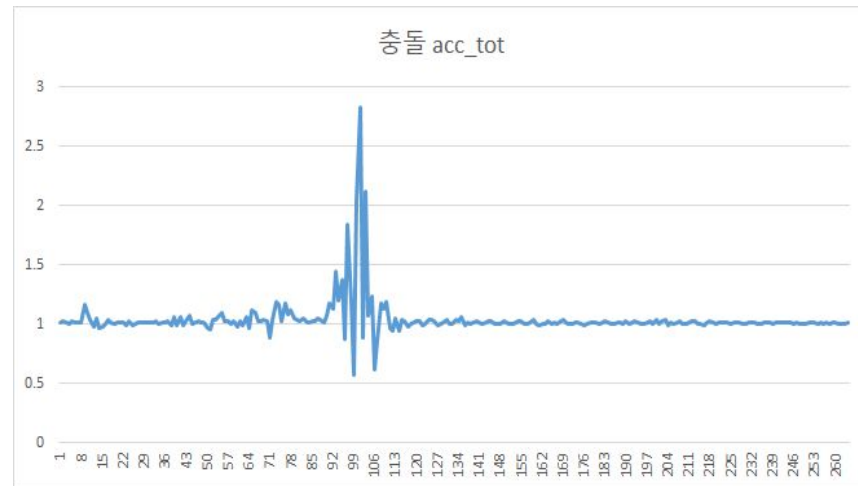
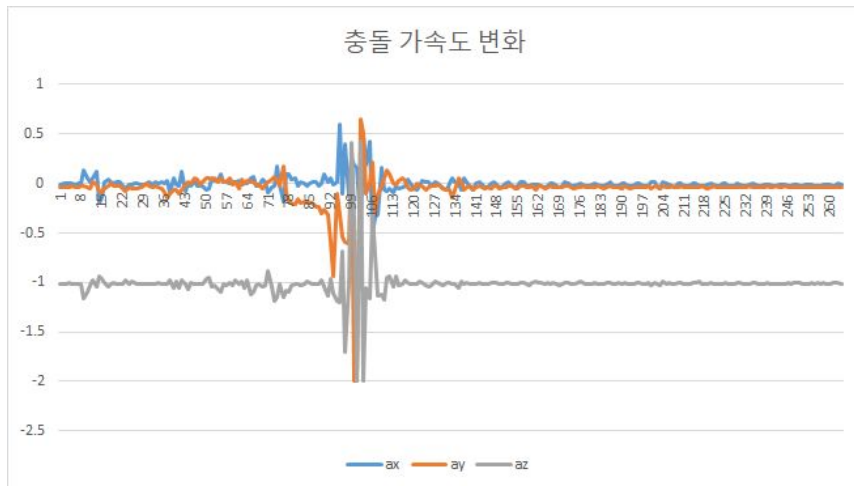
```
#넘어짐
if a2 == 0:
    if (float(x[2])>=0.9 and float(x[2])<=1.08 and float(x[4])>=0.05 and float(x[4])<=0.19) or (float(x[2])>=-1.08
        a2 = 1
        status_fd.configure(fg="red")
else:
    if (float(x[2])>=0.9 and float(x[2])<=1.08 and float(x[4])>=0.05 and float(x[4])<=0.19) or (float(x[2])>=-1.08
        continue
    else:
        a2 = 0
        status_fd.configure(fg="gray")
```



# 데이터/코드 - 충돌

충돌 MIN = 0.571577

충돌 MAX = 2.832949



# 데이터/코드 - 충돌

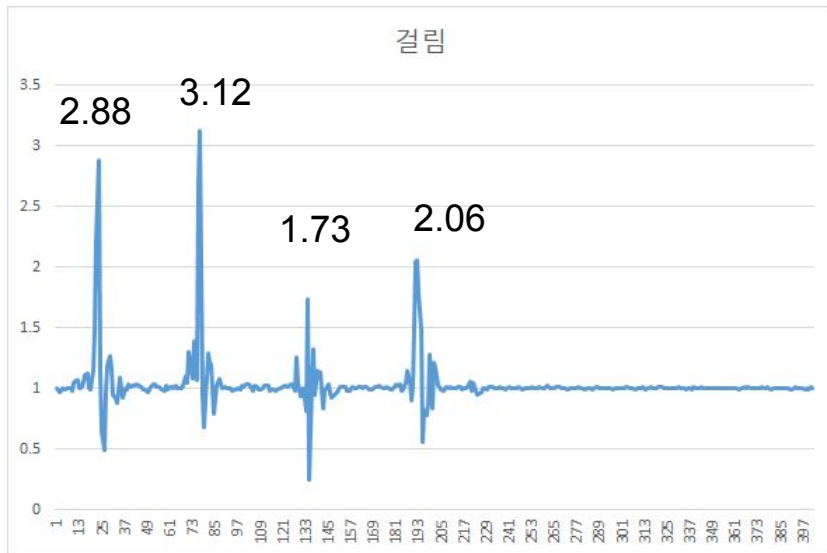
충돌 MAX = 2.832949

충돌 기준을 2.7정도로  
잡음

```
#충돌시간
if choong==1:
    ctime_2 = datetime.now()
    choongtime=(ctime_2 - ctime_1).total_seconds()
    if choongtime >= 2.3:
        status_c.configure(fg="gray")
#충돌
if a3==0:
    if float(acc_tot)>=2.7:
        a3=1
        ctime_p=datetime.now()
    else:
        if float(acc_tot)<= 2.0:
            a3=0
            ctime_c=datetime.now()
            ctime = (ctime_c - ctime_p).total_seconds()
            if ctime < 0.2 and ctime > 0:
                choong=1
                status_c.configure(fg="red")
                ctime_1=datetime.now()
```



# 데이터/코드 - 걸림



1.7을 기준으로 잡음  
13초 이내에 **acc\_tot**가 1.7을  
넘으면 걸림으로 분류

```
#걸림
if b==1:
    b_cur=datetime.now()
    b_time=(b_cur - b_prev).total_seconds()
    if b_time >= 3:
        status_b.configure(fg='gray')

if a4==0:
    if float(acc_tot)>=1.7:
        a4=1
        btime_p=datetime.now()

else:
    if float(acc_tot)<= 1.7:
        a4=0
        btime_c=datetime.now()
        btime_t = (btime_c - btime_p).total_seconds()
        if btime_t < 0.2:
            x2+=1

if x2 == 1:
    t_prev = datetime.now()

if x2 == 5:
    t_cur = datetime.now()
    elapsed_t = (t_cur - t_prev).total_seconds()
    if elapsed_t <= 13.0:
        x2 = 0
        b_prev = datetime.now()
        status_b.configure(fg="red")
        b = 1
        rec_start()
    else:
        a4 = 0
```

# 데이터/코드 - 걸림

#걸림 저장

```
if save_flag == 1:
    time_now = datetime.now() #현재시간
    elapsed = (time_now - time_prev).total_seconds()
    data_save.writelines(str(t_prev) + " - "+str(t_cur)+" 걸림상황 ")
    rec_stop()
```

test1.txt - Windows 메모장

파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)

2020-06-16 10:04:24.220591 - 2020-06-16 10:04:26.107243 걸림상황

test1.txt  
test2.txt  
test3.txt  
test4.txt  
test5.txt

2020-06-16 오전 ...  
2020-06-16 오전 ...  
2020-06-16 오전 ...  
2020-06-16 오전 ...  
2020-06-16 오전 ...



걸림상황에서 처음 걸린 시간 - 걸림 끝나는 시간  
텍스트 파일로 저장됨

# 데이터/코드 - 도움요청

```
# 도움요청
if float(x[1])==0:
    x1=0

if x1==0:
    if float(x[1]) == 1:
        touch+=1
        x1=1

if touch==1:
    t_prev= datetime.now()
if touch==3:
    t_cur = datetime.now()
    elapsed_t = (t_cur - t_prev).total_seconds()
    if elapsed_t <= 4.0:
        touch = 0
        state_h = '도움요청'#도움요청을 5초간 띄우고 싶다
        helptime_1= datetime.now()
        help=1
        status_h.configure(fg="red")
        #ser.write(b'2')
    else:
        touch=0
```

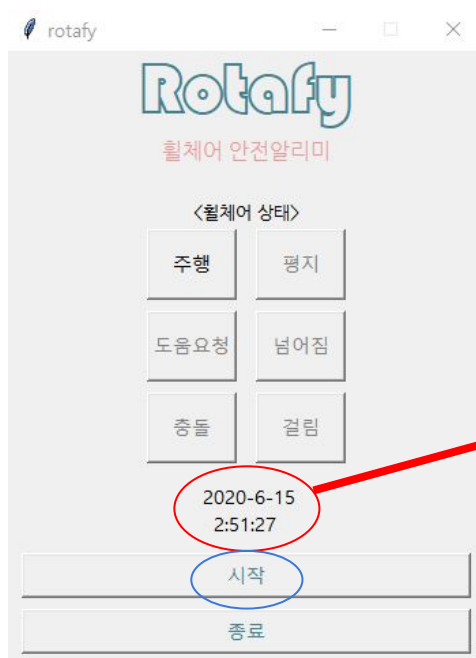
```
#도움요청시간
if help==1:
    helptime_2 = datetime.now()
    helptime=(helptime_2 - helptime_1).total_seconds()
    if helptime >= 5.0:
        #ser.write(b'3')
        status_h.configure(fg="gray")
```

4초 안에 3번을 연속으로 터치하면  
도움요청이 5초동안 빨간색으로 변함



# 그 외의 기능

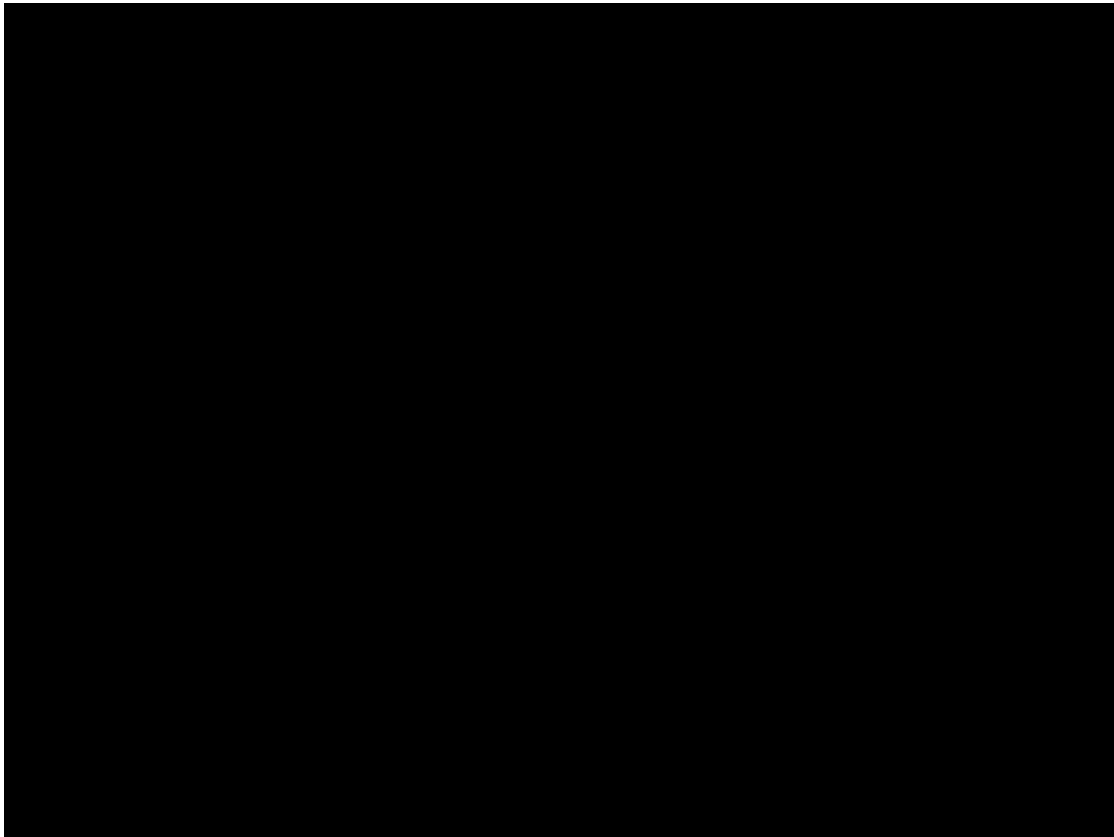
정지/시작 기능 및 실시간으로 현재시간 표현



현재 시간



# 영상



# 기대효과

- rotafy를 통하여 상황을 인지하고 상황에 따른 도움을 제공하기
- 즉각적인 대처를 통한 부상, 더 큰 사고를 방지하기
- 사고발생률/사고 피해 감소시키기
- 휠체어 장애인의 이동에 대한 자유도가 높아지고 자주성이 높아짐

# 발전방향

- 휠체어 사고 시, 직접적으로 도움을 줄 수 있는 **기술이 발전**한다면 그 기술에 **센서**로 쓰일 수 있다.
- **사고의 값을 저장 + 위치센서**와 함께 쓰인다면, 누적된 데이터를 통해 **사고다발지역**을 선정하여 개선할 수 있다.

# 아쉬운 점

- 걸림상황 데이터 저장 기능 개선

한 텍스트 파일에 값이 저장되도록

- 충돌과 걸림 기준이 모호

걸림과 충돌이 함께 나올 때가 있음

- 멈춤 주행이 연속적으로 표현X

값이 매끄럽게 표현되지 않음

감사합니다