

마이크로프로세서

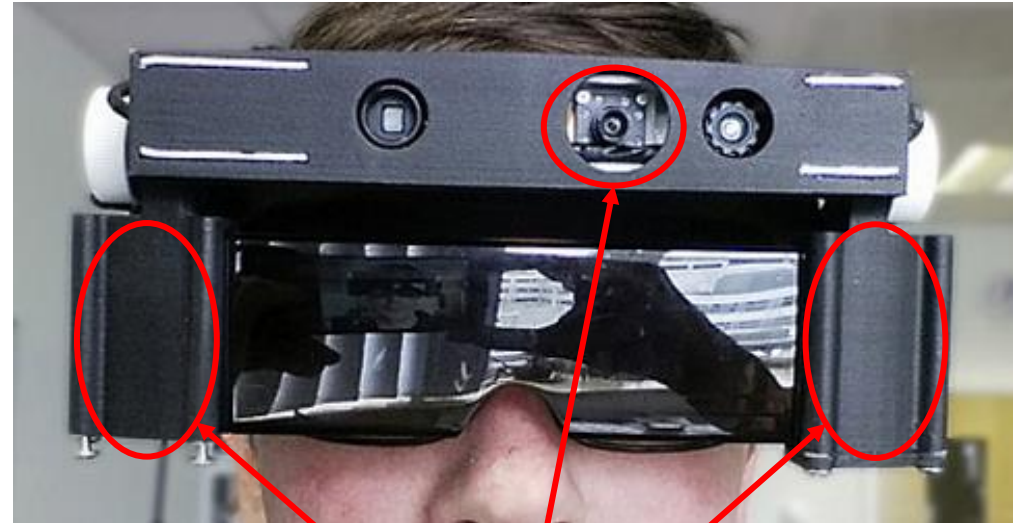
윤승민, 정OO

시각장애인을 위한 충돌방지 안경

결과 예상도 (각 부품 예상 위치, 디자인)



피에조 부저



적외선 센서

필요한 자원

라즈베리파이 4

안경 테

적외선 거리 측정센서 X 3

피에조 부저 X 2

적외선 센서 사용 코드

```
int distance = 0; //Cm로 계산된 거리값을 저장해두기위해 변수를 선언합니다.

void setup() {

    Serial.begin(9600); //거리값을 PC모니터로 쉽게 확인하기위해 시리얼통신을 선언해줍니다.

}

void loop() {

    int volt = map(analogRead(A0), 0, 1023, 0, 5000); /*
                                                    거리값을 cm로 변환해주기 쉽게 하기위해서
                                                    map 함수를 사용해서 데이터값을 변환후 volt 변수에 넣어 줍니다.
                                                    */

    distance = (27.61 / (volt - 0.1696)) * 1000; /*
                                                    거리값을 cm로 변환하는 계산공식 입니다.
                                                    변환된 cm값을 distance 변수에 저장합니다.
                                                    */

    Serial.print(distance); //거리값을 시리얼모니터로 출력해줍니다.
    Serial.println(" cm");
    Serial.println(" ");

    delay(500);
}
```

이를 통해 거리 값을 받아와 일정 거리 안쪽으로 접근하면 피에조 부저 작동

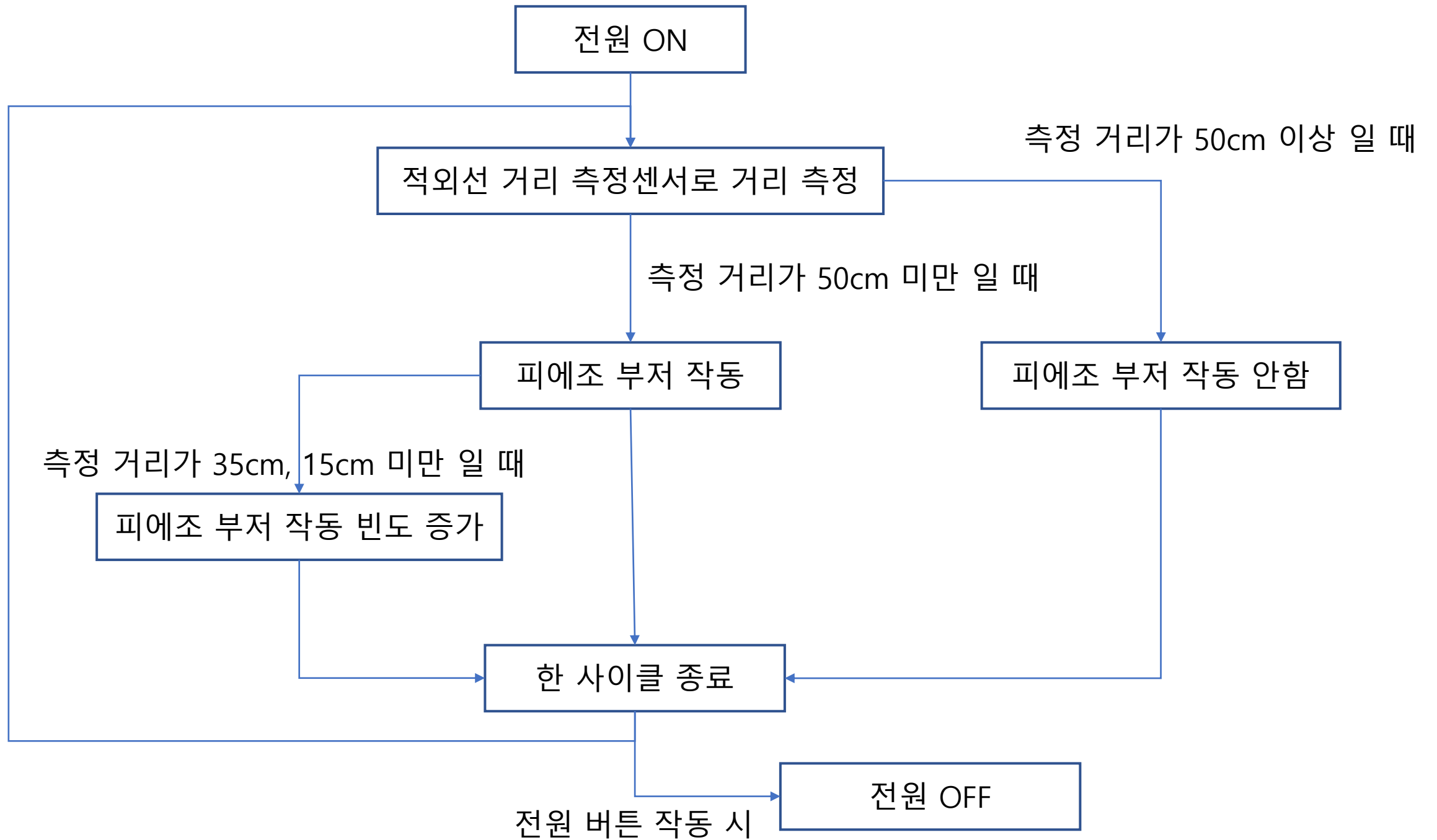
기존 작품, 특허, 논문

시각 장애인용 안전 안경 (특허 등록번호 : 20 - 492633 - 0000)

'OrCam MyEye' AI 시각 보조기기

옥스포드 대학 연구팀 - Smart Glasses [자료](#)

시각장애인을 위한 보행 안내 스마트 안경 플랫폼 설계 (이재범 외 2인, 동의대학교) [논문](#)

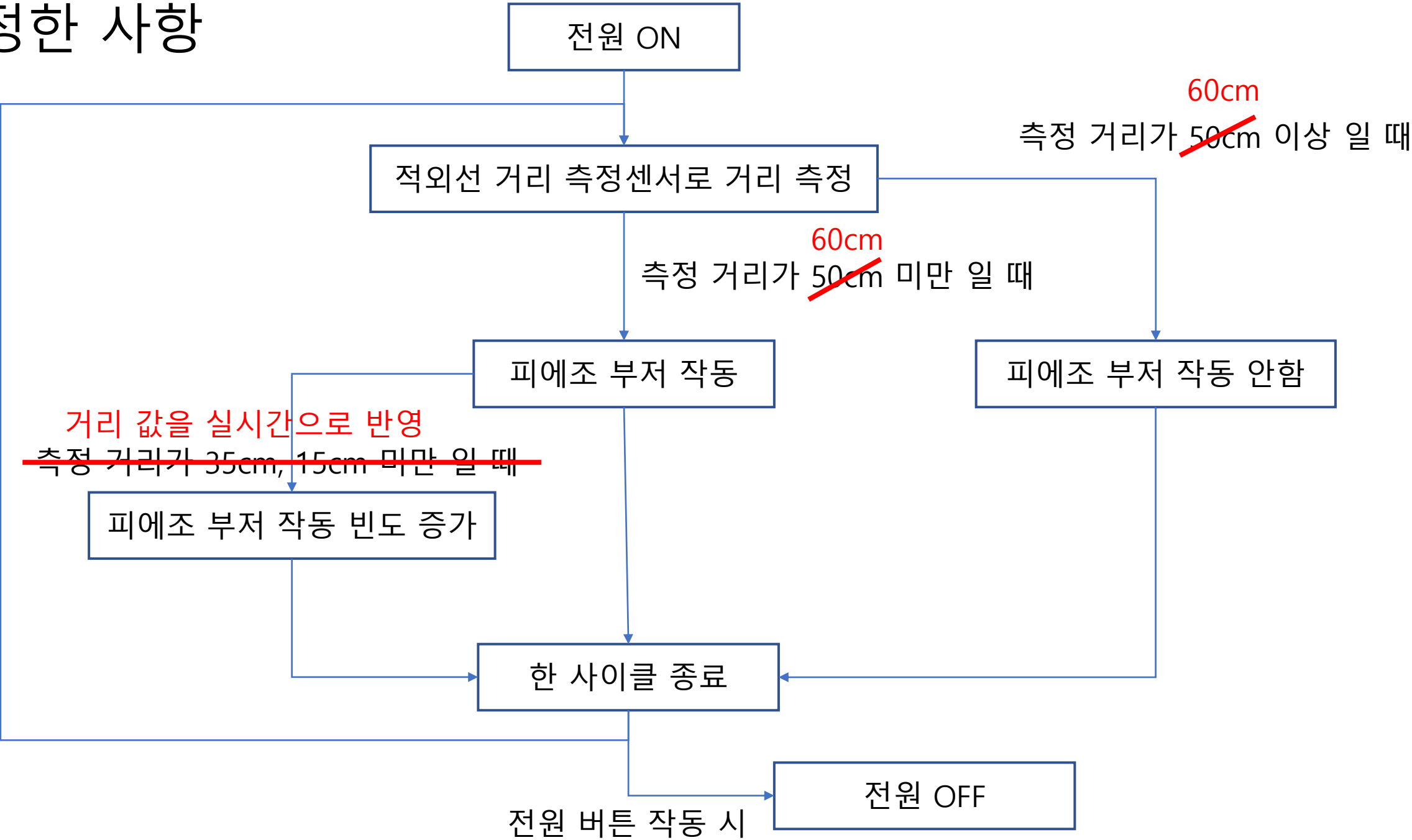


사용한 소스코드

```
1 import spidev
2 import time
3 import RPi.GPIO as GPIO
4
5 # 피에조부저 핀 번호 지정
6 piezoLeft = 20
7 piezoRight = 16
8
9 GPIO.setwarnings(False)
10
11 GPIO.setmode(GPIO.BCM)
12 GPIO.setup(piezoLeft, GPIO.OUT)
13 GPIO.setup(piezoRight, GPIO.OUT)
14
15 spi = spidev.SpiDev()
16 spi.open(0, 0)
17 spi.max_speed_hz = 1000000
18
19 def readChannel(channel):
20     val = spi.xfer2([1, (8 + channel) << 4, 0])
21     data = ((val[1] & 3) << 8) + val[2]
22     return data
23
24 # 채널에서 데이터 값을 받아오고 이를 사용한 센서에 맞는 거리변환 공식을 사용해 거리값으로 변환
25 def distance(channel):
26     vol = readChannel(channel) / 1023.0 * 3.3
27     distance = (16.2537 * vol**4 - 129.893 * vol**3 + 382.268 * vol**2 - 512.611 * vol + 301.439) / 2
28     return distance
29
30 # 거리에 따라 피에조 부저 작동 빈도수를 조절 (값이 낮을수록 더 자주)
31 def delay(distance):
32     delaytime = distance / 100
33     time.sleep(delaytime)
34
35 if __name__ == "__main__":
36     try:
37         pwnLeft = GPIO.PWM(piezoLeft, 262)
38         pwnRight = GPIO.PWM(piezoRight, 262)
39
40         while True:
41             distL = distance(0)
42             distM = distance(1)
43             distR = distance(2)
44             print("Left: %.2f cm" % distL)
45             print("Middle: %.2f cm" % distM)
46             print("Right: %.2f cm" % distR)
47
```

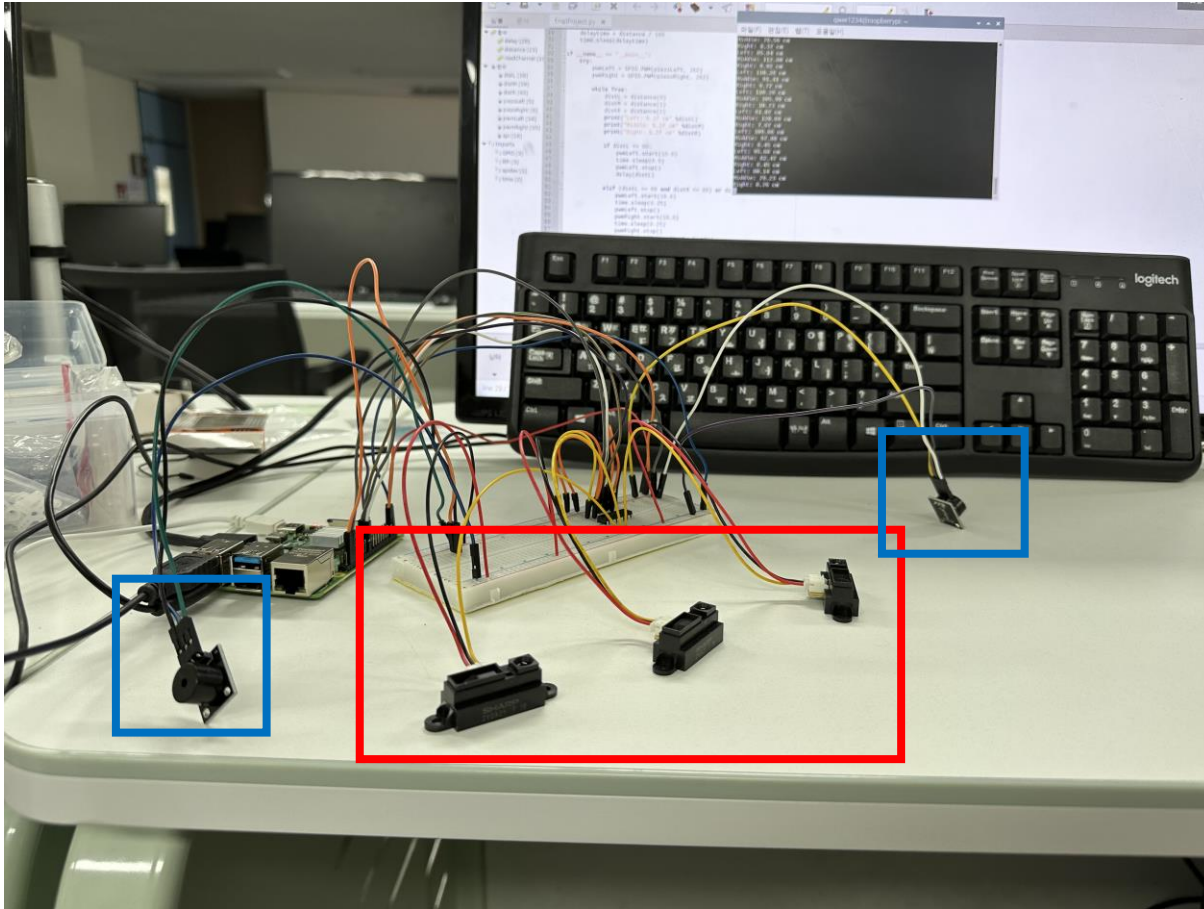
```
48
49
50 # 60cm 안에 물체가 인식된 경우
51
52 # 왼쪽 센서에만 물체가 인식되었을 때
53 if distL <= 60:
54     pwmLeft.start(10.0)
55     time.sleep(0.5)
56     pwmLeft.stop()
57     delay(distL)
58
59 # 가운데(또는 왼쪽과 오른쪽 동시에) 물체가 인식되었을 때
60 elif (distL <= 60 and distR <= 60) or distM <= 60:
61     pwmLeft.start(10.0)
62     time.sleep(0.25)
63     pwmLeft.stop()
64     pwmRight.start(10.0)
65     time.sleep(0.25)
66     pwmRight.stop()
67     delay(min(distL, distM, distR))
68
69 # 오른쪽 센서에만 물체가 인식되었을 때
70 elif distR <= 60:
71     pwmRight.start(10.0)
72     time.sleep(0.5)
73     pwmRight.stop()
74     delay(distR)
75
76 else:
77     time.sleep(0.5)
78
79 # 종료
80 except KeyboardInterrupt:
81     pwmLeft.stop()
82     pwmRight.stop()
83     GPIO.cleanup()
84     pass
```

수정한 사항



결과

시연 동영상



적외선 거리 측정 센서, **피에조 부저모듈**

