library

from pop import LiDAR

import math

lidar = LiDAR.Rplidar()

lidar.connect()

lidar.startMotor()

degree\_num = 36 # 장애물을 측정하는 각도의 개수

degree = [i for i in range(0, 360, 360//degree\_num)]

serbot\_width = (1 / math.sqrt(3)) \* 2 \* 100

serbot\_width = 500

# 거리에 따른 측정 각도를 반환하는 함수

def calcAngle(length):

    radian = math.atan((serbot\_width/2) / length)

    angle = radian \* (180 / math.pi)

    return angle

# 거리에 따른 장애물을 측정하는 함수

def collisonDetect(length):

    vectors = lidar.getVectors()

    angle = calcAngle(length)

    detect = [False] \* degree\_num # 장애물이 없으면 False, 있으면 True

    for i in range(degree\_num):

        h = []

        an\_a = i\*(360/degree\_num) - angle

        an\_b = i\*(360/degree\_num) + angle

        if an\_a < 0:   an\_a = 360 + an\_a

        if an\_a > 360: an\_a = an\_a - 360

        if an\_b < 0:   an\_b = 360 + an\_b

        if an\_b > 360: an\_b = an\_b - 360

        for v in vectors:

            if an\_a < an\_b:

                if(v[0] >= an\_a and v[0] <= an\_b):

                    h.append(v[1])

            else:

                if(v[0] >= an\_a or v[0] <= an\_b):

                    h.append(v[1])

        if len(h) > 0:

            if min(h) < length: detect[i] = True # 장애물이 있다.

            else: detect[i] = False

    return detect

if \_\_name\_\_ == "\_\_main\_\_":

    while True:

        de\_list = collisonDetect(800)

        print(de\_list)

main

from pop import Pilot

from nh import degree\_num, collisonDetect, lidar

import random, math

speed = 30 # 속도

bot = Pilot.SerBot()

bot.setSpeed(speed)

state = True

mode = 0 # 주행 방향

bot.move(mode \* (360/degree\_num), speed)

while state:

    try:

        if collisonDetect(300)[mode]:

            bot.stop()

            continue

        de\_list = collisonDetect(800)

        print(mode, de\_list)

        if sum(de\_list) == degree\_num:

            bot.stop()

            print("detected all")

            continue

        if de\_list[mode] == True:

            de\_false\_list = [i for i, val in enumerate(de\_list) if not val]

            mode = random.choice(de\_false\_list)

        bot.move(mode \* (360/degree\_num), speed)

    except KeyboardInterrupt:

        state = False

lidar.stopMotor()

bot.stop()

print("finish")