|  |
| --- |
| #!/usr/bin/env python3 |
|  |  |
|  | ##Librerias a usar## |
|  | import matplotlib.pyplot as plt |
|  | import math as mt |
|  | import zmq |
|  | import time |
|  | from time import sleep |
|  | from PIL import Image |
|  | ##Declaramos el contexto y creamos los Socket## |
|  | context = zmq.Context() |
|  | s = context.socket(zmq.REP) |
|  | s1 = context.socket(zmq.REP) |
|  | s2 = context.socket(zmq.REP) |
|  | s3 = context.socket(zmq.REP) |
|  |  |
|  | ##Direcion y puerto## |
|  | s.bind("tcp://192.168.1.90:7525") |
|  | s1.bind("tcp://192.168.1.90:6763") |
|  | s2.bind("tcp://192.168.1.90:6625") |
|  | s3.bind("tcp://192.168.1.90:5376") |
|  |  |
|  | def datos(): |
|  | archi=open('datos/dato.txt', 'a') |
|  | archi.write('Fecha y Hora: '+ time.strftime("%x ")+ time.strftime("%X")+ '\n') |
|  | archi.close() |
|  |  |
|  | ##Parametros iniciales## |
|  | j,k = 0,0 |
|  | d,an = 0,0 |
|  | cm = 0 |
|  | ##Entrando al ciclo## |
|  | print ("Esperando respuesta:") |
|  | while True: |
|  |  |
|  | v=s.recv() |
|  |  |
|  | vf = float(v) |
|  |  |
|  | s.send\_string(str(vf)) |
|  |  |
|  | if vf > 35: |
|  |  |
|  | lum=s1.recv() |
|  | s1.send\_string(str(lum)) |
|  |  |
|  | ma=s2.recv() |
|  | rue = (float(ma) / 360) #vueltas aproximadas que han dado las ruedas |
|  | cm =(float(ma) \* 0.0275) ##Convercion a cm por cada grados de vuelta |
|  | m = (cm \* 0.01) ##convercion de cm por metros |
|  | ft = (m \* 3.28084) ##convercion de cm por metros |
|  | s2.send\_string(str(cm)) |
|  |  |
|  | if vf > 65: |
|  | print ("No hay objeto detectado", "luz ambiental detectada" + str(lum) + "%") |
|  | print ("N° de giros de rueda: "+ str(rue),"Distancia recorida en: "+ str(cm) +"cm"+", "+ str(m) +"m"+", "+ str(ft) +"ft") |
|  | def gdatos(): |
|  | archi=open('datos/dato.txt', 'a') |
|  | archi.write("No hay objeto detectado"+ "y" "luz ambiental detectada" + str(lum) + "%"+'\n') |
|  | archi.write("N° de giros de rueda: "+ str(rue)+ "y"+"Distancia recorida en: "+ str(cm) +"cm"+", "+ str(m) +"m"+", "+ str(ft) +"ft"+'\n') |
|  | archi.close() |
|  | datos() |
|  | gdatos() |
|  |  |
|  | else: |
|  | print ("Objeto detectado a: "+ str(v) +"cm", "luz ambiental detectada" + str(lum) + "%") |
|  | print ("N° de giros de rueda: "+ str(rue),"Distancia recorida en: "+ str(cm) +"cm"+", "+ str(m) +"m"+", "+ str(ft) +"ft") |
|  | def gdatos(): ##Deficiondo la funcion de guardado de tados |
|  | archi=open('datos/dato.txt', 'a') |
|  | archi.write("Objeto detectado a: "+ str(v) +"cm"+"y"+ "luz ambiental detectada" + str(lum) + "%"+'\n') |
|  | archi.write("N° de giros de rueda: "+ str(rue)+"y"+"Distancia recorida en: "+ str(cm) +"cm"+", "+ str(m) +"m"+", "+ str(ft) +"ft"+'\n') |
|  | archi.close() |
|  | datos() |
|  | gdatos() |
|  |  |
|  | else: |
|  |  |
|  | c = cm |
|  |  |
|  | d = d + an |
|  | a = c\*mt.cos(mt.radians(d)) |
|  | b = c\*mt.sin(mt.radians(d)) |
|  |  |
|  | x =(j,(a + j)) |
|  | y =(k,(b + k)) |
|  | ## print (x,y) |
|  |  |
|  | plt.plot(x,y) |
|  | plt.savefig('Map\_of\_ev3.png') |
|  | j,k = (a + j,b + k) |
|  |  |
|  | an = s3.recv() |
|  | print("dis: "+ str(cm),"angulo: "+ str(an)) |
|  | an = float(an) |
|  |  |
|  | if an > 0: |
|  | print(an) |
|  | s3.send\_string(str(an) +'degrees to the right') |
|  | else: |
|  | print(an) |
|  | s3.send\_string(str(an) +'degrees to the left') |