```
from tkinter import *
import time
import os
import math
import motors as motor

# variables globales



joyXprev,joyYprev = 0,0
modulo = 0
angulo = 0
mouse_inscreen = 0
joyX = 0
joyY = 0

fwd = 0

#motor.rightMotor(100,1)
#motor.leftMotor(100,1)



'''
############################### MOTORES ##################################################################
GPIO.setmode (GPIO.BOARD)

RM_A = 10
RM_B = 8
LM_A = 3
LM_B = 5

pin_pwm_L = 7
pin_pwm_R = 12
GPIO.setup(RM_A ,GPIO.OUT)
GPIO.setup(RM_B,GPIO.OUT)
GPIO.setup(LM_A, GPIO.OUT)
GPIO.setup(LM_B, GPIO.OUT)

GPIO.setup(pin_pwm_L,GPIO.OUT)
GPIO.setup(pin_pwm_R,GPIO.OUT)

pwm_L = GPIO.PWM(pin_pwm_L,100)
pwm_R = GPIO.PWM(pin_pwm_R,100)

pwm_L.start(100)
pwm_R.start(100)



############################ FUNCIONES VARIAS###################################################################
def leftMotor (velL):
    pwm_L.ChangeDutyCycle(velL)
    if fwd:
        GPIO.output(LM_A,True)
        GPIO.output(LM_B,False)

    else:
        GPIO.output(LM_A,False)
        GPIO.output(LM_B,True)

    print(velL)




def rightMotor (velR):
```

```python
    pwm_R.ChangeDutyCycle(velR)
    if fwd:
        GPIO.output(RM_A,True)
        GPIO.output(RM_A,False)
    else:
        GPIO.output(RM_A,False)
        GPIO.output(RM_A,True)
    print (velR)

'''

def calculate_module (x,y):
    module = x*x + y*y
    module = math.sqrt(module)
    return module

def create_circle(x, y, r, canvasName): #center coordinates, radius
    x0 = x - r
    y0 = y - r
    x1 = x + r
    y1 = y + r
    return canvasName.create_oval(x0, y0, x1, y1)

def algulo ():
    if joyX == 0:
        return 90
    else:
        angulo = math.atan(joyY/joyX)
    return angulo

def joystick (vector): # joystick
    fwd = 0
    joyX = vector.x*(2/3)-100
    joyY = vector.y*(-2/3)+100
    mouse_inscreen = 1
    if joyX > 100 or joyX < -100 :
        joyX = 0
        joyY = 0
        mouse_inscreen = 0

    if joyY > 100 or joyY < -100 :
        joyX = 0
        joyY = 0
        mouse_inscreen = 0

    if mouse_inscreen == 1 and calculate_module(joyX,joyY) < 100:
        modulo = calculate_module(joyX,joyY)
        canvas.delete(ALL)
        control_screen.delete(ALL)
        canvas.create_line(150,150,vector.x, vector.y,width=3,fill='blue')
        canvas.create_text(vector.x,vector.y+10, text = ("x = ",int (joyX),"y = ",int(joyY),"modulo = ",int(calculate_module(int (joyX),int(joyY))) ), fill = 'green')
        create_circle(153,153,149,canvas)
        control_screen.create_text(130,15, text = ("x = ",int (joyX),"y = ",int(joyY),"modulo = ",int(calculate_module(int (joyX),int(joyY))) ), fill = 'green')

        leftV = 0
        rightV = 0
        angulo = 0

        if joyX != 0:
            angulo = math.atan(joyY/joyX)
        else :
                    if joyY > 0:
                            leftV = int(modulo)
                            rightV = int(modulo)
                            fwd = 1
                    else:
                            leftV = int(modulo)
                            rightV = int(modulo)
                            fwd = 0


        if joyX > 0 and joyY >= 0: # cuadrante 1 giro alante derecha

                leftV = int(modulo)
                rightV = int((100 - math.cos(angulo)*100) * modulo / 100)
                fwd = 1


        if joyX < 0 and joyY >= 0:# cuadrante 2 alante izquierda
            leftV = int((100 - math.cos(angulo)*100) * modulo / 100)
            rightV = int(modulo)
            fwd = 1


        if joyX < 0 and joyY <= 0: #cuadrante 3
            leftV = int((100 - math.cos(angulo)*100) * modulo/100)
            rightV = int(modulo)
            fwd = 0


        if joyX > 0 and joyY <= 0: # cuadrante 4
            leftV = int(modulo)
            rightV = int((100 - math.cos(angulo)*100) * modulo / 100)
            fwd = 0

        motor.rightMotor(rightV,fwd)
        motor.leftMotor(leftV,fwd)
        #print (leftV,rightV)



    #print ("click en:",int(joyX),int(joyY))
    #print (mouse_inscreen)

def stop ():
    joyX = 0
    joyY = 0
    module = 0
    canvas.delete(ALL)




############################## PANTALLA ###################################################################

root = Tk()
root.title("DeltaDynamics'robot")
root.geometry("600x600")
root.config(bg="#3DC3D8")
root.config(bd=10)
root.config(relief="groove")


canvas = Canvas(width=300, height=300, bg='white')
canvas.place(x = 100, y =10 )
canvas.bind("<B1-Motion>", joystick)
canvas.config(relief="groove")
canvas.config(bd=1)
create_circle(153,153,149,canvas)
canvas.pack()

botonera = Frame()
botonera.place(x=0, y=350)
botonera.config(width=600, height=100)
botonera.config(bg ="#A53EF1")
botonera.config(relief="groove")

control_screen = Canvas(width = 300, height=30, bg='white')
control_screen.pack(pady = 5)

#botones

button_stop = Button(botonera,padx=50, pady=20, text="STOP", command = stop)
button_stop.pack(side = LEFT)

button_stop = Button(botonera,padx=50, pady=20, text="esto", command = stop)
button_stop.pack(side = LEFT)

button_stop = Button(botonera,padx=50, pady=20, text="dos cosas", command = stop)
button_stop.pack(side = LEFT)


root.mainloop()

```