'''Assumptions:

1) CW/CCW in terms pf moving forward or backward.'''

def step(a,b):

STP = #Number of steps required to achieve one rotation

s=[(b[0]-a[0])\*(STP/0.008),(b[1]-a[1])\*(STP/0.008),0.2\*(STP/0.008)]

return s

def one\_step():

import RPi.GPIO as GPIO

import time

GPIO.setmode(GPIO.BCM)

CW=1

CCW=0

R={'Full':(0,0,0),'Half':(),'1/4':(0,1,0),'1/8':(1,1,0),'1/16':(0,0,1),'1/32':(1,0,1)}

'''PINS to be notified while Experimenting'''

axisx = [] # 2 GPIO

DIRx=

STEPx=

delayx= #Time gap (Mandatory)

GPIO.setup(DIRx,GPIO.OUT)

GPIO.setup(STEPx,GPIO.OUT)

axisy = [] # 5 GPIO

DIRy=

STEPy=

MODEy=() #For microstepping

GPIO.setup(DIRy,GPIO.OUT)

GPIO.setup(STEPy,GPIO.OUT)

GPIO.setup(MODEy,GPIO.OUT)

axisz = [] # 5 GPIO

DIRz=

STEPz=

MODEz=() #For microstepping

GPIO.setup(DIRz,GPIO.OUT)

GPIO.setup(STEPz,GPIO.OUT)

GPIO.setup(MODEy,GPIO.OUT)

try:

while True:

s=step(o\_list[i],p\_list[i]) #Output from a function

stepx = s[0]

stepsy = s[1]

stepsz = s[2]

if int(stepx)<0:

GPIO.output(DIRx,CCW)

for x in range(-int(stepx)):

GPIO.output(STEPx,GPIO.HIGH)

time.sleep(delayx)

GPIO.output(STEPx,GPIO.LOW)

time.sleep(delayx)

time.sleep(0.5)

else:

GPIO.output(DIRx,CW)

for x in range(int(stepx)):

GPIO.output(STEPx,GPIO.HIGH)

time.sleep(delayx)

GPIO.output(STEPx,GPIO.LOW)

time.sleep(delayx)

time.sleep(0.5)

'''We will use a correction factor due to presence of mode in y and z direction '''

if int(stepy)<0:

GPIO.output(DIRy,CCW)

for x in range(-int(stepy)\*cfy):

GPIO.output(STEPy,GPIO.HIGH)

time.sleep(delayx/cfy)

GPIO.output(STEPy,GPIO.LOW)

time.sleep(delayx/cfy)

time.sleep(0.5)

else:

GPIO.output(DIRy,CW)

for x in range(int(stepy)\*cfy):

GPIO.output(STEPy,GPIO.HIGH)

time.sleep(delayx/cfy)

GPIO.output(STEPy,GPIO.LOW)

time.sleep(delayx/cfy)

time.sleep(0.5)

if int(stepz)<0:

GPIO.output(DIRz,CCW)

for x in range(-int(stepz)\*cfz):

GPIO.output(STEPz,GPIO.HIGH)

time.sleep(delayx/cfz)

GPIO.output(STEPz,GPIO.LOW)

time.sleep(delayx/cfz)

time.sleep(0.5)

else:

GPIO.output(DIRz,CW)

for x in range(int(stepy)\*cfz):

GPIO.output(STEPz,GPIO.HIGH)

time.sleep(delayx/cfz)

GPIO.output(STEPz,GPIO.LOW)

time.sleep(delayx/cfz)

time.sleep(0.5)

except KeyboardInterrupt:

print("Quit")

GPIO.cleanup()