import smbus #import SMBus module of I2C

from time import sleep #import

import time

import matplotlib.pyplot as plt

#some MPU6050 Registers and their Address

PWR\_MGMT\_1 = 0x6B

SMPLRT\_DIV = 0x19

CONFIG = 0x1A

GYRO\_CONFIG = 0x1B

INT\_ENABLE = 0x38

ACCEL\_XOUT\_H = 0x3B

ACCEL\_YOUT\_H = 0x3D

ACCEL\_ZOUT\_H = 0x3F

GYRO\_XOUT\_H = 0x43

GYRO\_YOUT\_H = 0x45

GYRO\_ZOUT\_H = 0x47

def MPU\_Init():

#write to sample rate register

bus.write\_byte\_data(Device\_Address, SMPLRT\_DIV, 7)

#Write to power management register

bus.write\_byte\_data(Device\_Address, PWR\_MGMT\_1, 1)

#Write to Configuration register

bus.write\_byte\_data(Device\_Address, CONFIG, 0)

#Write to Gyro configuration register

bus.write\_byte\_data(Device\_Address, GYRO\_CONFIG, 24)

#Write to interrupt enable register

bus.write\_byte\_data(Device\_Address, INT\_ENABLE, 1)

def read\_raw\_data(addr):

#Accelero and Gyro value are 16-bit

high = bus.read\_byte\_data(Device\_Address, addr)

low = bus.read\_byte\_data(Device\_Address, addr+1)

#concatenate higher and lower value

value = ((high << 8) | low)

#to get signed value from mpu6050

if(value > 32768):

value = value - 65536

return value

bus = smbus.SMBus(1) # or bus = smbus.SMBus(0) for older version boards

Device\_Address = 0x68 # MPU6050 device address

MPU\_Init()

print (" Reading Data of Gyroscope and Accelerometer")

stepCount = 0

possibleStep = 0

topThreshold = 11.1

lowThreshold = 10.9

array = []

past = time.time()

restingAverage = 0

num = 1

while stepCount < 50:

#Read Accelerometer raw value

acc\_x = read\_raw\_data(ACCEL\_XOUT\_H)

acc\_y = read\_raw\_data(ACCEL\_YOUT\_H)

acc\_z = read\_raw\_data(ACCEL\_ZOUT\_H)

#Read Gyroscope raw value

gyro\_x = read\_raw\_data(GYRO\_XOUT\_H)

gyro\_y = read\_raw\_data(GYRO\_YOUT\_H)

gyro\_z = read\_raw\_data(GYRO\_ZOUT\_H)

#Full scale range +/- 250 degree/C as per sensitivity scale factor

Ax = acc\_x/16384.0

Ay = acc\_y/16384.0

Az = acc\_z/16384.0

Gx = gyro\_x/131.0

Gy = gyro\_y/131.0

Gz = gyro\_z/131.0

#print ("Gx=%.2f" %Gx, u'\u00b0'+ "/s", "\tGy=%.2f" %Gy, u'\u00b0'+ "/s", "\tGz=%.2f" %Gz, u'\u00b0'+ "/s", "\tAx=%.2f g" %Ax, "\tAy=%.2f g" %Ay, "\tAz=%.2f g" %Az)

#print("\tAx=%.2f g" %acc\_x, "\tAy=%.2f g" %acc\_y, "\tAz=%.2f g" %acc\_z)

if(Az > (restingAverage/num) + 0.04 and possibleStep == 0 and num > 20):

possibleStep += 1

if(Az < (restingAverage/num) - 0.04 and possibleStep == 1 and time.time() - past > 0.7):

stepCount += 1

possibleStep = 0

print("Steps=%d" %stepCount, "\tAcc\_Z: %.2f" %Az)

past = time.time()

else:

restingAverage += Az

num += 1

array.append(Az \* 9.8)

sleep(0.15)

plt.plot(array)

plt.ylabel("Acceleration")

plt.show()