**ProgramCode:**

import smbus

from gpiozero import LED, Button

import time

import board

import busio

import math

buz = LED(21)

sw = Button(16)

buz.off()

bus = smbus.SMBus(1)

bus.write\_byte\_data (0x53, 0x2C, 0x0B)

value = bus.read\_byte\_data(0x53, 0x31)

value &= ~0x0F;

value |= 0x0B;

value |= 0x08;

bus.write\_byte\_data(0x53, 0x31, value)

bus.write\_byte\_data(0x53, 0x2D, 0x08)

import requests

import serial

ser=serial.Serial('/dev/ttyUSB0',9600)

TOKEN = "BBFF-ovy1lu1PFvyYTTBZKK7Ers0z7agzfl"

DEVICE\_LABEL = "blackbox"

time.sleep(2)

stat = 1

def convert\_to\_degrees(raw\_value):

decimal\_value = raw\_value/100.00

degrees = int(decimal\_value)

mm\_mmmm = (decimal\_value -int(decimal\_value))/0.6

position = degrees + mm\_mmmm

position = "%.4f" %(position)

return position

while True:

if sw.is\_pressed:

buz.on()

print ("Bumper SwitchActivated")

url = "http://industrial.api.ubidots.com/api/v1.6/devices/blackbox"

headers = {"X-Auth-Token":"BBFF-ovy1lu1PFvyYTTBZKK7Ers0z7agzfl", "Content-Type": "application/json"}

req = requests.post(url=url, headers=headers, json={"bumper": 1})

status = req.status\_code

print (status)

time.sleep(1)

if status==200:

print ("Updated")

else:

print ("Failed")

else:

print ("Bumper Switch Normal")

buz.off()

url = "http://industrial.api.ubidots.com/api/v1.6/devices/blackbox"

headers = {"X-Auth-Token":"BBFF-ovy1lu1PFvyYTTBZKK7Ers0z7agzfl", "Content-Type": "application/json"}

req = requests.post(url=url, headers=headers, json={"bumper": 0})

status = req.status\_code

print (status)

bytes = bus.read\_i2c\_block\_data(0x53, 0x32, 6)

x = bytes[0] | (bytes[1] << 8)

if(x & (1 << 16 -1)):

x = x -(1<<16)

y = bytes[2] | (bytes[3] << 8)

if(y & (1 << 16 -1)):

y = y -(1<<16)

z = bytes[4] | (bytes[5] << 8)

if(z & (1 << 16 -1)):

z = z -(1<<16)

x = x \* 0.04

y = y \* 0.04

z = z \* 0.04

x = round(x, 4)

y = round(y, 4)

z = round(z, 4)

print(" x = %.3f ms2" %x)

print(" y = %.3f ms2" %y)

print(" z = %.3f ms2" %z)

print("\n\n")

if(((x<-3) or (x > 3)) or ((y<-3) or (y>3))):

print ("MEMS Alert")

buz.on()

url = "http://industrial.api.ubidots.com/api/v1.6/devices/blackbox"

headers = {"X-Auth-Token":"BBFF-ovy1lu1PFvyYTTBZKK7Ers0z7agzfl", "Content-Type": "application/json"}

req = requests.post(url=url, headers=headers, json={"accident": 1})

status = req.status\_code

print (status)

time.sleep(1)

if status==200:

print ("Updated")

else:

print ("Failed")

else:

print ("MEMS Normal")

buz.off()

url = "http://industrial.api.ubidots.com/api/v1.6/devices/blackbox"

headers = {"X-Auth-Token":"BBFF-ovy1lu1PFvyYTTBZKK7Ers0z7agzfl", "Content-Type": "application/json"}

req = requests.post(url=url, headers=headers, json={"accident": 0})

status = req.status\_code

print (status)

time.sleep(1)

if status==200:

print ("Updated")

else:

print ("Failed")

global la

global lo

# ser.flushInput()

ser.flushInput()

l=1

while(l==1):

print ("gps")

ch=0

while ch != '$':

ch=ser.read(1).decode()

ch=ser.read(1).decode()

if ch == 'G':

ch=ser.read(1).decode()

else:

l=1

if ch=='N':

ch=ser.read(1).decode()

else:

l=1

if ch=='G':

ch=ser.read(1).decode()

else:

l=1

if ch=='G':

ch=ser.read(1).decode()

else:

l=1

if ch=='A':

u=ser.read(11).decode()

la=ser.read(10).decode()

k = ser.read(3)

lo=ser.read(11).decode()

print (la)

print (lo)

time.sleep(5)

lat = float(la) #convert string into float for calculation

longi = float(lo) #convertr string into float for calculation

lat\_in\_degrees = convert\_to\_degrees(lat)

long\_in\_degrees = convert\_to\_degrees(longi)

url = "http://industrial.api.ubidots.com/api/v1.6/devices/blackbox"

headers = {"X-Auth-Token":"BBFF-ovy1lu1PFvyYTTBZKK7Ers0z7agzfl", "Content-Type": "application/json"}

req = requests.post(url=url, headers=headers, json={"gps":{"lat": lat\_in\_degrees, "lng": long\_in\_degrees}})

status = req.status\_code

print (status)

time.sleep(1)

if status==200:

print ("Updated")

else:

print ("Failed")

l=0

else:

l=1

time.sleep(1)