import board

import time

import analogio

import adafruit\_thermistor

from adafruit\_ble import BLERadio

from adafruit\_ble.advertising.standard import ProvideServicesAdvertisement

from adafruit\_ble.services.nordic import UARTService

import adafruit\_lis3dh

import busio

import digitalio

import math

####Setup blue tooth

ble = BLERadio()

uart\_server = UARTService()

advertisement = ProvideServicesAdvertisement(uart\_server)

print('Bluetooth Setup')

##Setup thermistor and light sensor

thermistor = adafruit\_thermistor.Thermistor(board.TEMPERATURE, 10000, 10000, 25, 3950)

light = analogio.AnalogIn(board.LIGHT)

print('Thermistor and Light Sensor Setup')

##Accelerometer is hooked up to SDA/SCL which is I2C or just some kind of protocol

i2c = busio.I2C(board.ACCELEROMETER\_SCL, board.ACCELEROMETER\_SDA)

\_int1 = digitalio.DigitalInOut(board.ACCELEROMETER\_INTERRUPT)

lis3dh = adafruit\_lis3dh.LIS3DH\_I2C(i2c, address=0x19, int1=\_int1)

lis3dh.range = adafruit\_lis3dh.RANGE\_8\_G

print('Accelerometer Setup')

##This function just converts the digital bit value to a range from 0 to 50

def scale(value):

"""Scale the light sensor values from 0-65535 (AnalogIn range)

to 0-50 (arbitrarily chosen to plot well with temperature)"""

return value / 65535 \* 50

while True:

# Advertise when not connected.

print('Not connected')

ble.start\_advertising(advertisement)

##Keep looping until connection established

while not ble.connected:

pass

#Stop advertising once connected

print('Connected')

ble.stop\_advertising()

##Once connected

while ble.connected:

#Print the light values and thermistor to serial

#l = light.value

t = thermistor.temperature

starttime = time.monotonic()

starttime = 0

T = time.monotonic()

time.sleep(0.1)

#Accelerometer

x,y,z = lis3dh.acceleration

#theta = math.atan(x/z)

time.sleep(0.1)

#print(scale(l), t)

print((T,x,y,z,t))

#print((T,x,y,z))

#print((T,t))

#print((theta,))

#And send them over uart (which is basically serial) but this is \_server

#which means it's wireless

uart\_server.write('{}\n'.format(t))

uart\_server.write('{},{},{}\n'.format(x,y,z))

uart\_server.write('{}\n'.format(T))