

\\athena\IPSA\home\isaori\Desktop\完成版\構成ファイル\omron_env_broadcast.py

```

1  from bluepy import btle
2  import struct
3
4  class ScanDelegate(btle.DefaultDelegate):
5      #コンストラクタ
6      def __init__(self):
7          btle.DefaultDelegate.__init__(self)
8          #センサデータ保持用変数
9          self.sensorValue = None
10
11     # スキャンハンドラー
12     def handleDiscovery(self, dev, isNewDev, isNewData):
13         # 新しいデバイスが見つかったら
14         if isNewDev or isNewData:
15             # アドバタイズデータを取り出し
16             for (adtype, desc, value) in dev.getScanData():
17                 #環境センサのとき、データ取り出しを実行
18                 if desc == 'Manufacturer' and value[0:4] == 'd502':
19                     #センサの種類 (EP or IM) を取り出し
20                     sensorType = dev.scanData[dev.SHORT_LOCAL_NAME].decode(encoding='utf-8'
21 )
22                     #EPのときのセンサデータ取り出し
23                     if sensorType == 'EP':
24                         self.decodeSensorData_EP(value)
25                     #IMのときのセンサデータ取り出し
26                     if sensorType == 'IM':
27                         self.decodeSensorData_IM(value)
28
29     # センサデータを取り出してdict形式に変換 (EPモード時)
30     def decodeSensorData_EP(self, valueStr):
31         #文字列からセンサデータ(6文字目以降)のみ取り出し、バイナリに変換
32         valueBinary = bytes.fromhex(valueStr[6:])
33         #バイナリ形式のセンサデータを整数型Tupleに変換
34         (temp, humid, light, uv, press, noise, discomf, wbgt, rfu, batt) = struct.unpack('<
35 hhhhhhhhB', valueBinary)
36         #単位変換した上でdict型に格納
37         self.sensorValue = {
38             'SensorType': 'EP',
39             'Temperature': temp / 100,
40             'Humidity': humid / 100,
41             'Light': light,
42             'UV': uv / 100,
43             'Pressure': press / 10,
44             'Noise': noise / 100,
45             'Discomfort': discomf / 100,
46             'WGBT': wbgt / 100,
47             'BatteryVoltage': (batt + 100) / 100
48         }
49
50     # センサデータを取り出してdict形式に変換 (IMモード時)
51     def decodeSensorData_IM(self, valueStr):
52         #文字列からセンサデータ(6文字目以降)のみ取り出し、バイナリに変換
53         valueBinary = bytes.fromhex(valueStr[6:])
54         #バイナリ形式のセンサデータを整数型Tupleに変換
55         (temp, humid, light, uv, press, noise, accelX, accelY, accelZ, batt) =
56 struct.unpack('<hhhhhhhhhB', valueBinary)
57         #単位変換した上でdict型に格納
58         self.sensorValue = {

```

```
56         'SensorType': 'IM',
57         'Temperature': temp / 100,
58         'Humidity': humid / 100,
59         'Light': light,
60         'UV': uv / 100,
61         'Pressure': press / 10,
62         'Noise': noise / 100,
63         'AccelerationX': accelX / 10,
64         'AccelerationY': accelY / 10,
65         'AccelerationZ': accelZ / 10,
66         'BatteryVoltage': (batt + 100) / 100
67     }
68
```