

0621

【センサデータの表示】

ラズパイ (python) --- Enddevice(NXP) ==wireless==> Coordinator(NXP)---PC(python)

今までは Coordinator での NXP でセンサデータを表示していたが、別で python を用意し、表示される。

【ラズパイの python コード】

```
al.py  sensor.py
30 SerialObj.bytesize = 8 # Number of data bits = 8
31 SerialObj.parity = 'N' # No parity
32 SerialObj.stopbits = 1 # Number of Stop bits = 1
33 SerialObj.timeout = None # Setting timeouts here No timeouts, waits for
34
35 def uint8ToBytes (dataArray, decimalValue):
36     bytes = decimalValue.to_bytes(1, byteorder="little")
37     dataArray.extend(bytes)
38 def uint16ToBytes (dataArray, decimalValue):
39     bytes = decimalValue.to_bytes(2, byteorder="little")
40     dataArray.extend(bytes)
41 def uint32ToBytes (dataArray, decimalValue):
42     bytes = decimalValue.to_bytes(4, byteorder="little")
43     dataArray.extend(bytes)
44 def uint64ToBytes (dataArray, decimalValue):
45     bytes = decimalValue.to_bytes(8, byteorder="little")
46     dataArray.extend(bytes)
47
48 def SendSensor(data_sensor, i):
49
50     dataBytes = bytearray()
51     uint8ToBytes(dataBytes, 170) # Begin of data 0xAA
52     uint16ToBytes(dataBytes, i) # primitive type 0xBB(187)
53     dataBytes_sensor = bytearray(struct.pack("d", data_sensor))
54     dataBytes.extend(dataBytes_sensor)
55     print([ "0x%02x" % b for b in dataBytes ])
56     SerialObj.write(dataBytes)
57
58
59
60 while True:
61     data = 0
62     sum_data = 0
63     volt = 0
64     sum_volt = 0
65     count = 0
66     avg_data = 0
67     avg_volt = 0
68
69     #ADS1015からデータを取得
70     data = ads.read_adc(ads1015_pin, gain=GAIN)
71     sum_data += data
72     volt = data * UNIT
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
```

【Coordinator でのコード】

```
u16bytesread = PDU_u16APduInstanceReadNB0(sStackEvent.uEvent.sApsDataIndEvent.hAPduInst, 0, "a\x10", &Rxbyte);
SendBytes(Rxbyte[1], 1); //受信したデータ1目をUart通信で送信する関数
SendBytes(Rxbyte[2], 1); //受信したデータ2目をUart通信で送信する関数
```

[PC での python コード]

```
sensor.py
1 import serial
2 import time
3
4 def main():
5     try:
6         ser = serial.Serial(
7             port='COM5',      # デバイスポート番号
8             baudrate=115200,  # ボーレート
9             parity=serial.PARITY_NONE,
10            stopbits=serial.STOPBITS_ONE,
11            bytesize=serial.EIGHTBITS,
12            timeout=1
13        )
14        if ser.is_open: # シリアルポートがオープンされているかチェック
15            print("Serial port opened successfully.")
16        while True: # データ受信
17            print("Waiting for data...")
18            if ser.in_waiting > 0: # 受信バッファにデータがあるか確認
19                print("Data in buffer")
20                high_data = ser.read(1)
21                low_data = ser.read(1)
22                val = (ord(high_data) << 8) + ord(low_data)
23                voltage = val * (5.0 / 1024)
24                print(f"Voltage: {voltage:.2f} V")
25                time.sleep(1) # 1秒待機
26
27    except serial.SerialException as e:
28        print("Error opening or operating the serial port:", e)
29    finally:
30        if ser.is_open:
31            ser.close()
32            print("Serial port closed.")
33
34 if __name__ == "__main__":
35     main()
36
```

[ラズパイの表示結果]

```
受信データ : 429.0
電圧       : 0.858
['0xaa', '0x08', '0x00', '0xdb', '0xf9', '0x7e', '0x6a', '0xbc', '0x74', '0xeb', '0x3f']
b'APP: Starting ZPS\r\n'
受信データ : 430.0
電圧       : 0.860
['0xaa', '0x09', '0x00', '0x85', '0xeb', '0x51', '0xb8', '0x1e', '0x85', '0xeb', '0x3f']
b'APP: No event to process!\r\n'
受信データ : 429.0
電圧       : 0.858
['0xaa', '0x0a', '0x00', '0xdb', '0xf9', '0x7e', '0x6a', '0xbc', '0x74', '0xeb', '0x3f']
```

[PC での表示結果]

```
C:\Users\郷平\Desktop>python sensor.py
Serial port opened successfully.
Waiting for data...
Waiting for data...
Data in buffer
Voltage: 278.58 V
Waiting for data...
Data in buffer
Voltage: 278.58 V
Waiting for data...
Data in buffer
Voltage: 278.58 V
Waiting for data...
Data in buffer
Voltage: 278.58 V
Waiting for data...
Data in buffer
Voltage: 278.58 V
```

明らかに、PC 上での結果数値がおかしい。

また、ラズパイでのデータ値も正しい値なのかを調査し、調整する予定である。

【ラズパイを使用した実験方法について】

前回の養殖場では、ラズパイを操作する PC デスクトップを持ち歩いていた。そのため、今後はラズパイをコンセントに接続した際に、プログラムを起動させるようにする。

こうすることで、スムーズな実験を行うことができる。

実際にマルチホップでの実証実験に、実装する予定である。

参考 URL

[Raspberry Pi4 起動時に指定したプログラムを実行させる | パソコン工房 NEXMAG \(pc-koubou.jp\)](https://nexamag.jp/pc-koubou.jp/)

【LQIを確認できるアプリ PAL_ARIA_Viewer】

ダウンロード済み。

今後の通信実験で使用する予定。

【M2 輪講.英語発表の質疑応答について】

MONOSTICK とケーブルとの価格比較

Q. 水族館にあるような巨大な水槽が複数個ある環境で、Router として MONOSTICK を数台設置し迂回させる場合と、長距離のケーブル 1.2 本で Coordinator と Enddevice を繋げる場合と、どちらが安価なのか？ MONOSTICK1 台が 3000 円前後ならば、ケーブルの場合の方が安価なのでは？

A. 想定環境が異なる。私の研究場合は、タンクの大きさの水槽が複数個ある環境や、工場にある冷凍室の環境などの密閉空間や電波の届かない空間で得たセンサデータをマルチホップで外にデータを運んで Coordinator に送信する、といった環境を想定している。そのため、水族館にあるような巨大な水槽が複数個ある環境では、ケーブルを使用すれば良い。