

[1]Enddevice から受信した 16 進数データが, Coordinator では途中までしか表示されないという問題について

以下の関数の第 3 引数を変更することで解決した.

Enddevice 側

```
u16Offset += PDUM_u16APduInstanceWriteNBO(hAPduInst, u16Offset, "a\x10", RxByte); //16進数センサデータ
```

Coordinator 側

```
u16bytesread = PDUM_u16APduInstanceReadNBO(sStackEvent.uEvent.sApsDataIndEvent.hAPduInst,0,"a\x10",&Rxbyte);
```

a¥x08 → a¥x10 に変更した. 数値は byte 数を意味している.

[2]約 100byte ほどのデータを送受信 (Enddevice から Coordinator 間) できるようにする事について

現在はラズパイから Enddevice では 16 進数データを 1byte ずつ送信している. そして Enddevice でセンサの 1 つデータを表す 9byte ためて, Coordinator へ送信している.

ただし, 毎回データの順番を表すカウント数 1byte も付与して送信するため, 実質 10byte である.

Enddevice でためておくデータを 10byte(配列の要素数を 10)以上にすると, Coordinator で受信した際に, データが正しくなかったり, 決まった無意味の数値がデータの最後に付与されたりしてしまう問題が発生している.

#### 【今後の課題】

- ・実際のセンサでは必要ないが, Enddevice から Coordinator まで一気に約 100byte ほど送信できることを目標にする.

>1 つの案として, 卒論時点で実装していた文字列でのマルチホップを再度採用して, 100 文字送信し, 結果を確認することも試す予定.

[3]辻先生が所有するセンサ（水質計、温湿度計、分光計）について

【計測から通信の手順】

1. MONOSTICK とセンサを USB で接続(1920bps)
2. MONOSTICK (NXP) のプログラムでセンサからデータを取得するために、コマンドを入力送信する.
3. コマンドの認識後は, センサから MONOSTICK へデータが送信される.
4. 取得したセンサデータを Coordinator である MONOSTIC に送信し、Coordinator 側で変換する.

センサデータは CSV で送信される. そのため, 実際にセンサを扱って実験できないため, プログラム(ラズパイの Python)にカンマ区切りのデータを用意し, それを Enddevice で受信し, Coordinator へ送信, Coordinator で受信できるように実装した.

【今後の課題】

・現在はセンサの各データを float から int(8byte)に変換し, バイト数が冗長である. 浮動小数点のデータ数を減らすために, CO2 温湿度計では 2byte, 分光計では 4byte で送信できるようにする. そして各センサの仕様書を参考に, センサデータを扱うプログラムを構築しておく.

#### [4]2024/04/10 の養殖場での実証実験について

前回の1回目の実証実験 2024/02/27 では, Router を経由した無線マルチホップネットワークでの通信が確認できた。しかし, 水槽の少なさなどにより, Enddevice と Coordinator が直接接続できる状態もあった。また Enddevice と Coordinator は別の部屋にも関わらず, 通信できていたのも確認できたため, 壁も貫通していた可能性も判明した。

##### 【今回の2回目の実証実験 2024/04/10 で判明したこと】

- ・1回目と同様に, 水槽以外の自由空間の割合が大きい場合, Router を介さずに Coordinator と Enddevice が直接接続されること。
- ・センサからのデータ取得速度が 0.1s と 0.01s では異常なく通信可能であったこと。
- ・0.001s ではパケロスが発生したこと。以下の画像はパケロスが発生した際のデータの一部である。受信した割合は,  $30 / 65 (01 \sim 41) = 0.461 \quad 46\%$

```
02. [MAC: 0xe7de][Data16: 0x06736891ed7c3fed3f]
04. [MAC: 0xe7de][Data16: 0x060c022b8716d9ee3f]
06. [MAC: 0xe7de][Data16: 0x06000000000000f03f]
07. [MAC: 0xe7de][Data16: 0x0621b0726891edec3f]
0a. [MAC: 0xe7de][Data16: 0x065839b4c876beef3f]
0c. [MAC: 0xe7de][Data16: 0x0608ac1c5a643bef3f]
0d. [MAC: 0xe7de][Data16: 0x06736891ed7c3fed3f]
10. [MAC: 0xe7de][Data16: 0x06b0726891ed7cef3f]
14. [MAC: 0xe7de][Data16: 0x06653bdf4f8dedec3f]
15. [MAC: 0xe7de][Data16: 0x061f85eb51b81eed3f]
18. [MAC: 0xe7de][Data16: 0x06c976be9f1a2fed3f]
1a. [MAC: 0xe7de][Data16: 0x06c976be9f1a2fed3f]
1c. [MAC: 0xe7de][Data16: 0x06b0726891ed7cef3f]
1f. [MAC: 0xe7de][Data16: 0x0662105839b4c8ee3f]
22. [MAC: 0xe7de][Data16: 0x0604560e2db29def3f]
24. [MAC: 0xe7de][Data16: 0x0604560e2db29def3f]
26. [MAC: 0xe7de][Data16: 0x06ae47e17a14aeef3f]
29. [MAC: 0xe7de][Data16: 0x061f85eb51b81eed3f]
2a. [MAC: 0xe7de][Data16: 0x06068195438b6cef3f]
2c. [MAC: 0xe7de][Data16: 0x0677be9f1a2fddec3f]
2e. [MAC: 0xe7de][Data16: 0x066f1283c0caa1ed3f]
30. [MAC: 0xe7de][Data16: 0x06022b8716d9ceef3f]
33. [MAC: 0xe7de][Data16: 0x06d122dbf97e6aec3f]
35. [MAC: 0xe7de][Data16: 0x065839b4c876beef3f]
37. [MAC: 0xe7de][Data16: 0x061f85eb51b81eed3f]
39. [MAC: 0xe7de][Data16: 0x06c74b37894160ed3f]
3c. [MAC: 0xe7de][Data16: 0x06d578e9263108f03f]
3e. [MAC: 0xe7de][Data16: 0x06cff753e3a59bec3f]
3f. [MAC: 0xe7de][Data16: 0x06cff753e3a59bec3f]
41. [MAC: 0xe7de][Data16: 0x060c022b8716d9ee3f]
```

##### 【今後の方針】

- ・MONOSTICK を水中に入れて, 通信状況を確認する。
- 水槽の内壁に電波発信部分を水中に, USB 部分を水面から出ている状態にし, 強制的に電波を伝えづらくし, マルチホップの環境を構築する。
- ・100byte 相当のデータを扱えるようにし, 実験を行う。
  - ・実際に養殖場で使用されているセンサを用いて, 通信状況を確認する。