0510

【Enddevice から Coordinator へ約 100byte のデータ通信】

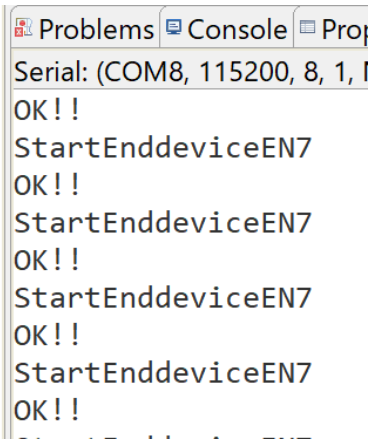
以下の Enddevice のコード画像のように“StartEnddeviceEND”（17byte） という文字列を Coordinator へ送信するようにした.
[Enddevice 側]

```
u16Offset += PDUM_u16APduInstanceWriteNBO(hAPduInst, u16Offset,"a\x10", "StartEnddeviceEND");
DBG_vPrintf(TRACE_APP, "\n");
```

[Coordinator 側]

```
uint16 u16bytesread;
uint8_t i;

字列実験用コード
char u8TempPayload[256];
for(i = 0; i < 20; i++){
u16bytesread = PDUM_u16APduInstanceReadNBO(sStackEvent.uEvent.sApsDataIndEvent.hAPduInst,i,"a\x10",&u8TempPayload);
DBG_vPrintf(TRACE_APP, "%c", u8TempPayload[i]);
}

DBG_vPrintf(TRACE_APP, "\n");
```

[Coordinator の受信結果]



“StartEnddeviceEND”の最後の D が文字化けしていた.

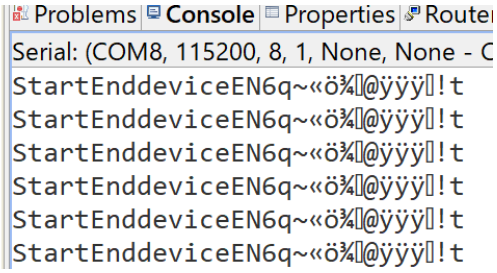
次に送信文字列の長さを 2 倍（34byte）にした. また，受信した文字列を格納する配列の要素数を増加した.
[Enddevice 側]

```
u16Offset += PDUM_u16APduInstanceWriteNBO(hAPduInst, u16Offset,"a\x10", "StartEnddeviceENDStartEnddeviceEND");
DBG_vPrintf(TRACE_APP, "\n");
```

[Coordinator 側]

```
F列実験用コード
   char u8TempPayload[256];
   u16bytesread = PDUM_u16APduInstanceReadNBO(sStackEvent.uEvent.sApsDataIndEvent.hAPduInst,i,"a\x10",&u8TempPayload);
   for(i = 0; i < 40; i++){
   DBG_vPrintf(TRACE_APP, "%c", u8TempPayload[i]);
   }

   DBG_vPrintf(TRACE_APP, "\n");
```

[Coordinator の受信結果]



上記の受信結果より，同様に D 以降の文字列が文字化けしていた．

次のように, Enddevice から送信する関数の第 3 引数を変更した．
以下は，関数の仕様説明である．

uint16 PDUM_u16APduInstanceReadNBO(
PDUM_thAPduInstance *hAPduInst*,
uint16 *u16Pos*,
const char **szFormat*,
void **pvStruct*);

Description

This function reads data from the specified APDU instance and inserts the data into a C structure. The byte position of the start (least significant byte) of the data in the APDU instance must be specified, as well as the format of the data.

Data is read from the APDU instance in packed network byte order (little-endian) and translated into unpacked host byte order for the C structure (big-endian for the JN51xx device).

Parameters

hAPduInst	Handle of APDU instance to read the data from
u32Pos	The starting position (least significant byte) of the data within the APDU
**szFormat*	Format string of the data:
	b 8-bit byte
	h 16-bit half-word (short integer)
	w 32-bit word
	l 64-bit long-word (long integer)
	a\xnn nn (hex) bytes of data (array)
	p\xnn nn (hex) bytes of packing
**pvStruct*	Pointer to C structure to receive the data

Note that the compiler will not correctly interpret the format string "a\xnnb" for a data array followed by a single byte, e.g. "a\x0ab". In this case, to ensure that the 'b' (for byte) is not interpreted as a hex value, use the format "a\xnn" "b", e.g. "a\x0a" "b".

Returns

Total number of data bytes read from the APDU instance

a¥x10 → a¥20 に変更

```
  char u8TempPayload[256];
  u16bytesread = PDUM_u16APduInstanceReadNBO(sStackEvent.uEvent.sApsDataIndEvent.hAPduInst,i,"a\x20",&u8TempPayload);
  for(i = 0; i < 40; i++){
  DBG_vPrintf(TRACE_APP, "%c", u8TempPayload[i]);
  }

  DBG_vPrintf(TRACE_APP, "\n");
```

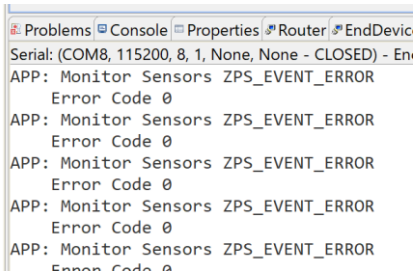
a¥x10 → a¥40 に変更

```
P列実験用コード
   char u8TempPayload[256];
   u16bytesread = PDUM_u16APduInstanceReadNBO(sStackEvent.uEvent.sApsDataIndEvent.hAPduInst,i,"a\x40",&u8TempPayload);
   for(i = 0; i < 40; i++){
   DBG_vPrintf(TRACE_APP, "%c", u8TempPayload[i]);
   }

   DBG_vPrintf(TRACE_APP, "\n");
```

a¥x10 → b に変更

```
char buffer[] = "StartEnddeviceEND";
      for (i = 0; i < 17; i++) {
            u16Offset += PDUM_u16APduInstanceWriteNBO(hAPduInst, u16Offset,"b", *(buffer + i));
            DBG_vPrintf(TRUE, "%c", *(buffer + i));
      }

      //u16Offset += PDUM_u16APduInstanceWriteNBO(hAPduInst, u16Offset,"a\x10", "StartEnddeviceEND");
      DBG_vPrintf(TRACE_APP, "\n");
```

[Coordinator の受信結果]



このエラーの意味は，パケットのペイロードが小さすぎる，というエラー文である．

C: >NXP>bstudio_nxp>sdk>JN-SW-4170>Components>ZPSAPL>Include>Zps_apl_ah.h

[今後の方針]

・MONOSTICK を水槽の内壁に設置し，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．

・センサ（Enddevice）が複数の場合を想定しての通信実験する．データ量が増え，送信間隔が短くなったときに，あえてホップさせて，問題なく通信できるかを確認する．