

0201

○現在の目的は,

Enddevice からの 16 進数のデータを, Coordinator が受信して 16 進数のまま表示させる.

以下は Enddevice から送信する際使用する Write 関数である.

PDUM_u16APdulInstanceWriteNBO

```
uint16 PDUM_u16APdulInstanceWriteNBO(  
    PDUM_thAPdulInstance hAPdulInst,  
    uint16 u16Pos,  
    const char *szFormat, ...);
```

Description

This function writes the specified data values into the specified APDU instance. The byte position of the start of the data (least significant byte) in the APDU instance must be specified, as well as the format of the data.

The data values are written into the APDU instance at the specified position in packed network byte order (little-endian). The input data values should be in host byte order (big-endian for the JN51xx device).

Parameters

<i>hAPdulInst</i>	Handle of the APDU instance to write the data into
<i>u32Pos</i>	The starting position (least significant byte) of the data within the APDU instance
<i>*szFormat</i>	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\xnnnn (hex) bytes of packing
...	Variable list of data values described by the format string

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".

以下は Coordinator が受信する際使用する Read 関数である.

PDUM_u16APdulInstanceReadNBO

```
uint16 PDUM_u16APdulInstanceReadNBO(  
    PDUM_thAPdulInstance hAPdulInst,  
    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

<i>hAPdulInst</i>	Handle of APDU instance to read the data from
<i>u32Pos</i>	The starting position (least significant byte) of the data within the APDU
<i>*szFormat</i>	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
<i>*pvStruct</i>	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".

[Enddevice]

```
u16Offset += PDUM_u16APduInstanceWriteNBO(hAPduInst, u16Offset, "a\x08", TxByte); 16進数センサデータ
```

TxByte[]には 16 進数のセンサデータが格納されてある。
これはラズパイの方で表示させ、確認済み。

[Coordinator]

```
uint32_t Rxbyte[32];
uint32_t *RxbytePointer;
RxbytePointer = &Rxbyte;

DBG_vPrintf(TRACE_APP, "received data16:");

//受信した16進数
u16bytesread = PDUM_u16APduInstanceReadNBO(sStackEvent.uEvent.sApsDataIndEvent.hAPduInst, 0, "a\x08", RxbytePointer);
DBG_vPrintf(TRACE_APP, "%02x", RxbytePointer);
```

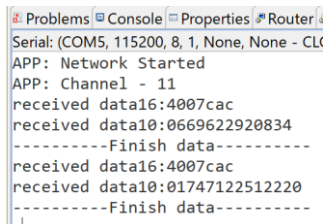
PDUM_u16APduInstanceReadNBO 関数の第 3 引数は 8byte の array のため、a¥x08

第 4 引数はポインタにして実装した。

しかし、ここでの Rxbyte[]には何も格納されてないプログラムのようにになっている。そのため、この Rxbyte[]に受信データを格納したいが、

PDUM_u16APduInstanceReadNBO 関数はポインタの仕様であるため、Rxbyte[]と直に格納できない。

[結果]



```
Serial: (COM5, 115200, 8, 1, None, None - CL
APP: Network Started
APP: Channel - 11
received data16:4007cac
received data10:0669622920834
-----Finish data-----
received data16:4007cac
received data10:01747122512220
-----Finish data-----
|
```

○センサデータの array を送受信するのではなく、1 byte ずつ送信するよう実装をする。

[Enddevice]

```
u16Offset += PDUM_u16APduInstanceWriteNBO(hAPduInst, u16Offset, "b", 0x01); //1byte

PDUM_eAPduInstanceSetPayloadSize(hAPduInst, u16Offset);
DBG_vPrintf(TRUE, "Size : %d\nSending : ", PDUM_u16APduInstanceGetPayloadSize(hAPduInst));
```

PDUM_u16APduInstanceWriteNBO 関数の第 3 引数を b にし、0x01 を送信する。

[Coordinator]

```
u16bytesread = PDUM_u16APduInstanceReadNBO(sStackEvent.uEvent.sApsDataIndEvent.hAPduInst, 0, "b", RxbytePointer); //1byte
DBG_vPrintf(TRACE_APP, "%02x", RxbytePointer);
```

[結果]

```
Serial: (COM5, 115200, 8, 1, None, None - CL
APP: Network Started
APP: Channel - 11
received data16:4007ca4
received data10:Error
111111
-----Finish data-----
received data16:4007ca4
received data10:Error
111111
-----Finish data-----
```

「1」を確認できた。

○次に 4byte ずつ送信できるよう実装した。

1. [Enddevice]

```
uint32 buffer = 1023;

//u16Offset += PDUM_u16APduInstanceWriteNB0(hAPduInst, u16Offset, "a\x08", TxByte); 16進数センサデータ
u16Offset += PDUM_u16APduInstanceWriteNB0(hAPduInst, u16Offset, "w", buffer);
```

[結果]

```
Serial: (COM5, 115200, 8, 1, None, None - CLOSED)
APP: Network Started
APP: Channel - 11
received data16:4007d74
received data10:Error
25530000
-----Finish data-----
received data16:4007d74
received data10:Error
25530000
-----Finish data-----
```

2. [Enddevice]

```
uint32_t buffer = 111222333;

//u16Offset += PDUM_u16APduInstanceWriteNB0(hAPduInst, u16Offset, "a\x08", TxByte); 16進数センサデータ
u16Offset += PDUM_u16APduInstanceWriteNB0(hAPduInst, u16Offset, "w", buffer);
```

[結果]

```
Serial: (COM5, 115200, 8, 1, None, None - CLOSE
APP: Network Started
APP: Channel - 11
received data16:4007d74
received data10:Error
6130161666
-----Finish data-----
```

○スケジュール

～2/15 16進数でセンサデータ表示と10進数変換

～2/22 10台のマルチホップで実験

2/27 牡蠣の養殖場で実証実験