

【Python】

センサから得たセンサ値を 16 進数に変更した結果の表示方法に誤りがあった。
以下が変更内容である。

```

56 def SendSensor(data_sensor):
57
58     dataBytes = bytearray()
59     uint8ToBytes(dataBytes, 170) # Begin of data 0xAA
60     uint16ToBytes(dataBytes, 187) # primitive type 0xBB
61     #uint8ToBytes(dataBytes, 3)
62     dataBytes_sensor = bytearray(struct.pack("d", data_sensor))
63     dataBytes.extend(dataBytes_sensor)
64     print([ "0x%02x" % b for b in dataBytes ])
65     Serial0bj.write(dataBytes)
66

```

また、大きいセンサ値を扱えるように 4byte の float 型ではなく、8byte の double 型にした。
以下がセンサから Enddevice への実行結果である。

```

ファイル(F) 編集(E) タブ(T) ヘルプ(H)
KYOHEI@KYOHEI:~ $ cd Desktop
KYOHEI@KYOHEI:~/Desktop $ python al.py
受信データ : 388.0
電圧 : 0.776
['0xaa', '0xbb', '0x00', '0xa2', '0x45', '0xb6', '0xf3', '0xfd', '0xd4', '0xe8', '0x3f']
b'\x00\x00In start state\r\n'
受信データ : 401.0
電圧 : 0.802
['0xaa', '0xbb', '0x00', '0xa4', '0x8b', '0x6c', '0xe7', '0xfb', '0xa9', '0xe9', '0x3f']
b'In primitive type1 state \r\n'
受信データ : 394.0
電圧 : 0.788
['0xaa', '0xbb', '0x00', '0x9e', '0xef', '0xa7', '0xc6', '0x4b', '0x37', '0xe9', '0x3f']
b'In primitive type2 state \r\n'
受信データ : 403.0
電圧 : 0.806
['0xaa', '0xbb', '0x00', '0x98', '0xe6', '0x12', '0x83', '0xc0', '0xca', '0xe9', '0x3f']
b'data: 0 \r\n'
受信データ : 404.0
電圧 : 0.808
['0xaa', '0xbb', '0x00', '0xa2', '0x60', '0xe5', '0xd0', '0x22', '0xdb', '0xe9', '0x3f']
b'data: a2 \r\n'
受信データ : 398.0
電圧 : 0.796
['0xaa', '0xbb', '0x00', '0xa6', '0xb6', '0xf3', '0xfd', '0xd4', '0x78', '0xe9', '0x3f']
b'data: 45 \r\n'
受信データ : 365.0
電圧 : 0.730
['0xaa', '0xbb', '0x00', '0x5c', '0x8f', '0xc2', '0xf5', '0x28', '0x5c', '0xe7', '0x3f']
b'data: b6 \r\n'
受信データ : 395.0
電圧 : 0.790
['0xaa', '0xbb', '0x00', '0xa8', '0xe1', '0x7a', '0x14', '0xae', '0x47', '0xe9', '0x3f']
b'data: f3 \r\n'
受信データ : 389.0
電圧 : 0.778
['0xaa', '0xbb', '0x00', '0x4c', '0x37', '0x89', '0x41', '0x60', '0xe5', '0xe8', '0x3f']
b'data: fd \r\n'
受信データ : 359.0
電圧 : 0.718
['0xaa', '0xbb', '0x00', '0x60', '0xe5', '0xd0', '0x22', '0xdb', '0xf9', '0xe6', '0x3f']
b'data: d4 \r\n'
受信データ : 401.0
電圧 : 0.802
['0xaa', '0xbb', '0x00', '0xa4', '0x8b', '0x6c', '0xe7', '0xfb', '0xa9', '0xe9', '0x3f']
b'In start state\r\n'
受信データ : 397.0
電圧 : 0.794
['0xaa', '0xbb', '0x00', '0x9c', '0xc4', '0x20', '0xb0', '0x72', '0x68', '0xe9', '0x3f']
b'In primitive type1 state \r\n'
受信データ : 398.0
電圧 : 0.796
['0xaa', '0xbb', '0x00', '0xa6', '0xb6', '0xf3', '0xfd', '0xd4', '0x78', '0xe9', '0x3f']
b'Command : 3\r\n'
^Z
[1]+ 停止 python al.py

```

表示内容から、Enddevice はセンサからのデータを 16 進数で受信できている。

「b'data:」が Enddevice の表示である。

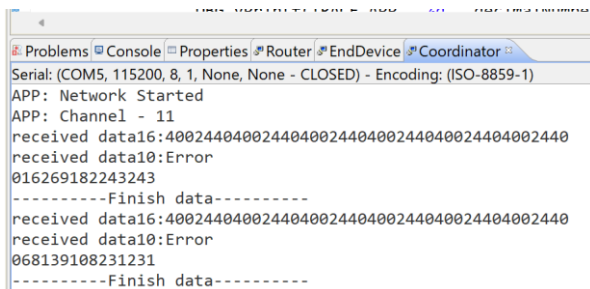
【NXP】

UART で受信した 8 byte のセンサ値を配列に格納できるように、ループ数を調整した。

Enddevice から Coordinator へ 16 進数のセンサ値を送信、そして 10 進数に直す動作をする。
現在は Enddevice から Coordinator へ 16 進数のセンサ値を送信を行えるよう取り組む。
以下は Coordinator のコードで、受信した 16 進数のデータが格納している tmpString を表示するコードである。

```
DBG_vPrintf(TRACE_APP, "received data16:");  
  
for(i = 0; i < 14; i++){ //受信した16進数  
    u16bytesread = PDUM_u16APduInstanceReadNBO(sStackEvent.uEvent.sApsDataIndEvent.hAPduInst,i,"b",&tmpString);  
    DBG_vPrintf(TRACE_APP, "%x", tmpString[i]);  
}
```

Coordinator の表示結果は



```
Serial: (COM5, 115200, 8, 1, None, None - CLOSED) - Encoding: (ISO-8859-1)  
APP: Network Started  
APP: Channel - 11  
received data16:4002440400244040024404002440400244040024404002440  
received data10:Error  
016269182243243  
-----Finish data-----  
received data16:4002440400244040024404002440400244040024404002440  
received data10:Error  
068139108231231  
-----Finish data-----
```

以下は PDUM_u16APduInstanceReadNBO 関数に関する説明である。

PDUM_u16APduInstanceReadNBO

```
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

<i>hAPduInst</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".

そこで PDUM_u16APduInstanceReadNBO 関数の第3引数に注目、以下のように変更をした。

[a%x01 に変更]

```
DBG_vPrintf(TRACE_APP, "received data16:");

for(i = 0; i < 14; i++){ //受信した16進数
    u16bytesread = PDUM_u16APduInstanceReadNBO(sStackEvent.uEvent.sApsDataIndEvent.hAPduInst,i,"a%x01",&tmpString);
    DBG_vPrintf(TRACE_APP, "%x", tmpString);
}

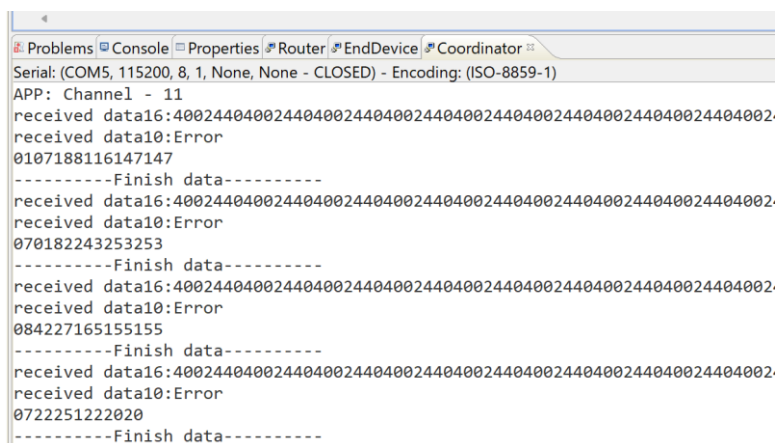
DBG_vPrintf(TRACE_APP, "\n");
```

[p%x01 に変更]

```
DBG_vPrintf(TRACE_APP, "received data16:");

for(i = 0; i < 14; i++){ //受信した16進数
    u16bytesread = PDUM_u16APduInstanceReadNBO(sStackEvent.uEvent.sApsDataIndEvent.hAPduInst,i,"p%x01",&tmpString);
    DBG_vPrintf(TRACE_APP, "%x", tmpString);
}
```

変更して動作してみたが、以下のように期待する結果にはならなかった。



The screenshot shows a serial console window with the following output:

```
Serial: (COM5, 115200, 8, 1, None, None - CLOSED) - Encoding: (ISO-8859-1)
APP: Channel - 11
received data16:4002440400244040024404002440400244040024404002440400244040024404002.
received data10:Error
0107188116147147
-----Finish data-----
received data16:400244040024404002440400244040024404002440400244040024404002.
received data10:Error
070182243253253
-----Finish data-----
received data16:400244040024404002440400244040024404002440400244040024404002.
received data10:Error
084227165155155
-----Finish data-----
received data16:400244040024404002440400244040024404002440400244040024404002.
received data10:Error
0722251222020
-----Finish data-----
```