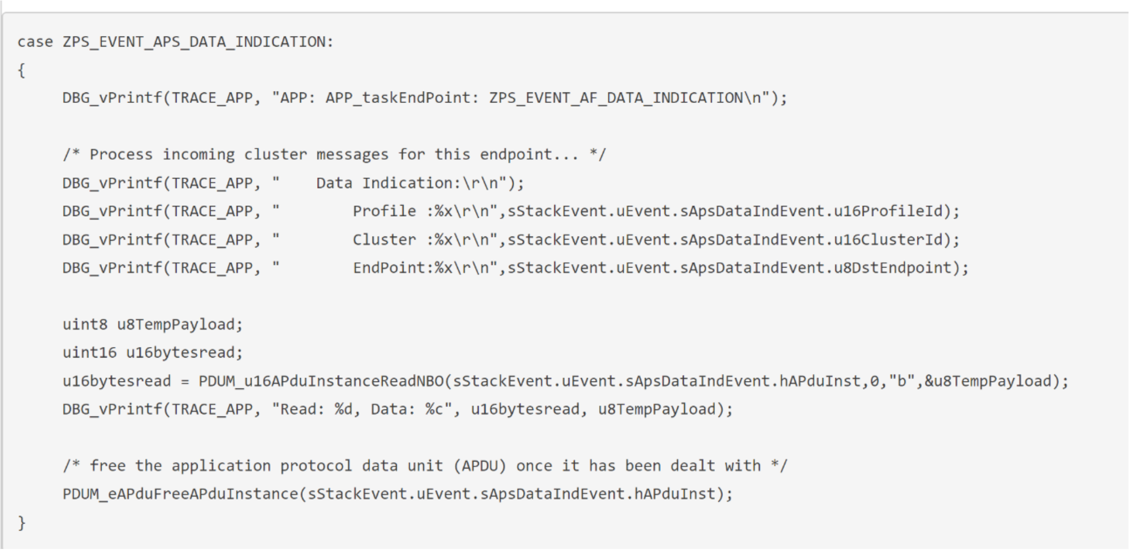
11/25

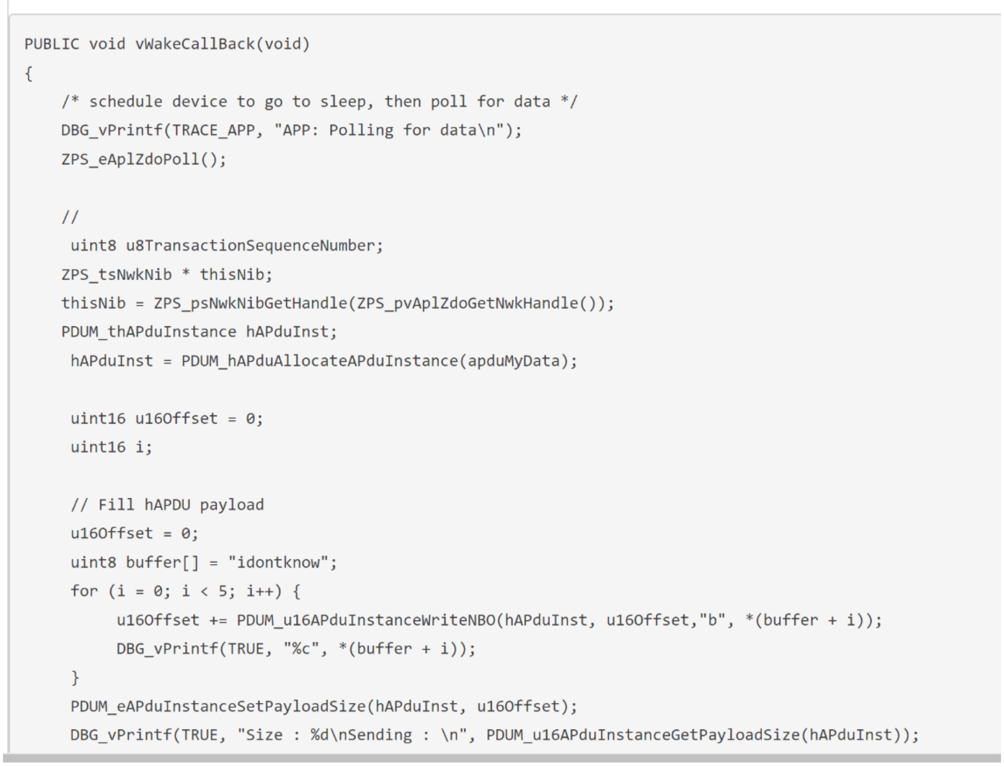
「NXP ZPS_eAplAfBroadcastDataReq」と検索
このサイトより、データ送信関数を調べた.
[已解决: Re: Simple Data transfer - NXP Community]

Coordinator にこのコードを追加.コンパイルはエラーなく成功した.



```
case ZPS_EVENT_APS_DATA_INDICATION:
{
    DBG_vPrintf(TRACE_APP, "APP: APP_taskEndPoint: ZPS_EVENT_AF_DATA_INDICATION\n");

    /* Process incoming cluster messages for this endpoint... */
    DBG_vPrintf(TRACE_APP, "    Data Indication:\r\n");
    DBG_vPrintf(TRACE_APP, "        Profile :%x\r\n",sStackEvent.uEvent.sApsDataIndEvent.u16ProfileId);
    DBG_vPrintf(TRACE_APP, "        Cluster :%x\r\n",sStackEvent.uEvent.sApsDataIndEvent.u16ClusterId);
    DBG_vPrintf(TRACE_APP, "        EndPoint:%x\r\n",sStackEvent.uEvent.sApsDataIndEvent.u8DstEndpoint);

    uint8 u8TempPayload;
    uint16 u16bytesread;
    u16bytesread = PDUM_u16APduInstanceReadNBO(sStackEvent.uEvent.sApsDataIndEvent.hAPduInst,0,"b",&u8TempPayload);
    DBG_vPrintf(TRACE_APP, "Read: %d, Data: %c", u16bytesread, u8TempPayload);

    /* free the application protocol data unit (APDU) once it has been dealt with */
    PDUM_eAPduFreeAPduInstance(sStackEvent.uEvent.sApsDataIndEvent.hAPduInst);
}
```

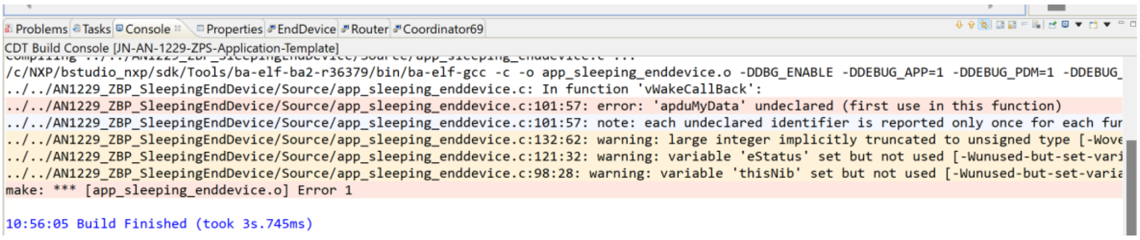
下記のコードを enddevice に追加した.



```
PUBLIC void vWakeCallBack(void)
{
    /* schedule device to go to sleep, then poll for data */
    DBG_vPrintf(TRACE_APP, "APP: Polling for data\n");
    ZPS_eAplZdoPoll();

    //
     uint8 u8TransactionSequenceNumber;
    ZPS_tsNwkNib * thisNib;
    thisNib = ZPS_psNwkNibGetHandle(ZPS_pvAplZdoGetNwkHandle());
    PDUM_thAPduInstance hAPduInst;
     hAPduInst = PDUM_hAPduAllocateAPduInstance(apduMyData);

     uint16 u16Offset = 0;
     uint16 i;

     // Fill hAPDU payload
     u16Offset = 0;
     uint8 buffer[] = "idontknow";
     for (i = 0; i < 5; i++) {
          u16Offset += PDUM_u16APduInstanceWriteNBO(hAPduInst, u16Offset,"b", *(buffer + i));
          DBG_vPrintf(TRUE, "%c", *(buffer + i));
     }
     PDUM_eAPduInstanceSetPayloadSize(hAPduInst, u16Offset);
     DBG_vPrintf(TRUE, "Size : %d\nSending : \n", PDUM_u16APduInstanceGetPayloadSize(hAPduInst));
```

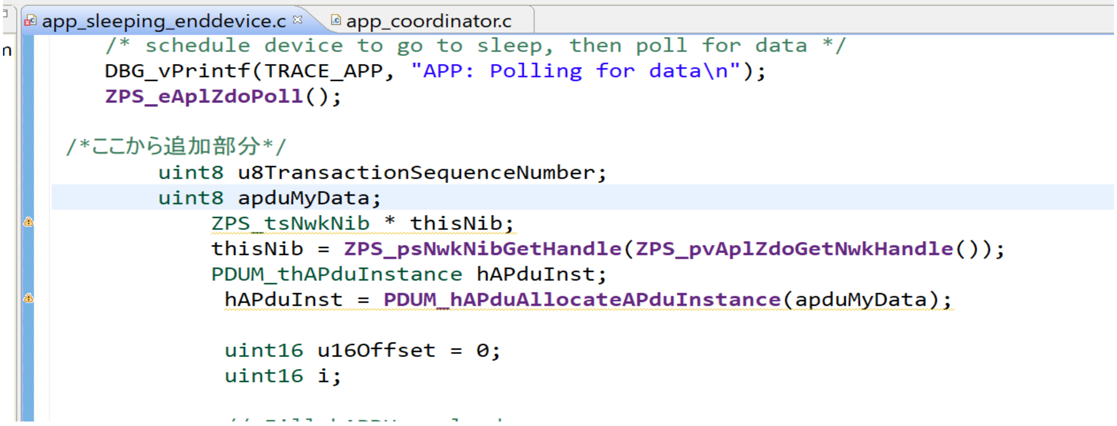
```
        }
        PDUM_eAPduInstanceSetPayloadSize(hAPduInst, u16Offset);
        DBG_vPrintf(TRUE, "Size : %d\nSending : \n", PDUM_u16APduInstanceGetPayloadSize(hAPduInst));


        if (hAPduInst == PDUM_INVALID_HANDLE)
        {
            DBG_vPrintf(TRUE, "PDUM_INVALID_HANDLE\n");
        } else {
            ZPS_teStatus eStatus;
            ZPS_teAplAfSecurityMode  eSecurityMode = (ZPS_E_APL_AF_SECURE_NWK);

            eStatus = ZPS_eAplAfBroadcastDataReq(     hAPduInst,
                                                       0x1337,
                                                       0x01,
                                                       0x01,
                                                       ZPS_E_BROADCAST_ZC_ZR,              //     Dest: All Coordinator & Routers
                                                       eSecurityMode,
                                                       0,
                                                       &u8TransactionSequenceNumber
                                                       );

        }
}
```

コンパイルするとエラーが発生した



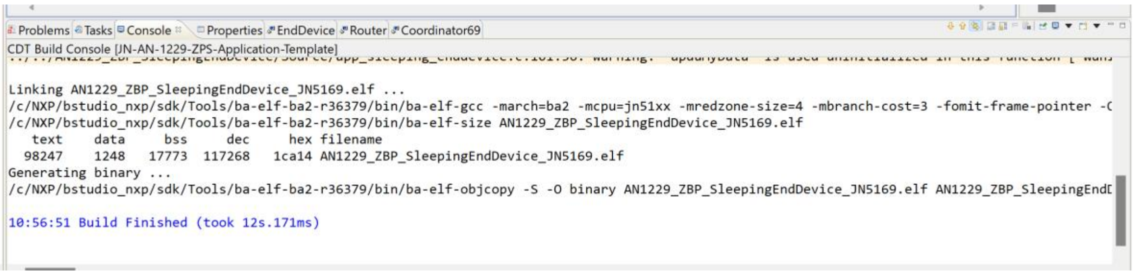
```
CDT Build Console [JN-AN-1229-ZPS-Application-Template]
Compiling ../../AN1229_ZBP_SleepingEndDevice/Source/app_sleeping_enddevice.c ...
/c/NXP/bstudio_nxp/sdk/Tools/ba-elf-ba2-r36379/bin/ba-elf-gcc -c -o app_sleeping_enddevice.o -DDBG_ENABLE -DDEBUG_APP=1 -DDEBUG_PDM=1 -DDEBUG_
../../AN1229_ZBP_SleepingEndDevice/Source/app_sleeping_enddevice.c: In function 'vWakeCallBack':
../../AN1229_ZBP_SleepingEndDevice/Source/app_sleeping_enddevice.c:101:57: error: 'apduMyData' undeclared (first use in this function)
../../AN1229_ZBP_SleepingEndDevice/Source/app_sleeping_enddevice.c:101:57: note: each undeclared identifier is reported only once for each fur
../../AN1229_ZBP_SleepingEndDevice/Source/app_sleeping_enddevice.c:132:62: warning: large integer implicitly truncated to unsigned type [-Wove
../../AN1229_ZBP_SleepingEndDevice/Source/app_sleeping_enddevice.c:121:32: warning: variable 'eStatus' set but not used [-Wunused-but-set-vari
../../AN1229_ZBP_SleepingEndDevice/Source/app_sleeping_enddevice.c:98:28: warning: variable 'thisNib' set but not used [-Wunused-but-set-varia
make: *** [app_sleeping_enddevice.o] Error 1

10:56:05 Build Finished (took 3s.745ms)
```

エラーより apduMyData が宣言されていないことだったので，下記のコードを追加．



```
    /* schedule device to go to sleep, then poll for data */
    DBG_vPrintf(TRACE_APP, "APP: Polling for data\n");
    ZPS_eAplZdoPoll();

 /*ここから追加部分*/
        uint8 u8TransactionSequenceNumber;
        uint8 apduMyData;
            ZPS_tsNwkNib * thisNib;
            thisNib = ZPS_psNwkNibGetHandle(ZPS_pvAplZdoGetNwkHandle());
            PDUM_thAPduInstance hAPduInst;
             hAPduInst = PDUM_hAPduAllocateAPduInstance(apduMyData);

             uint16 u16Offset = 0;
             uint16 i;
```

エラーなくコンパイルに成功した．

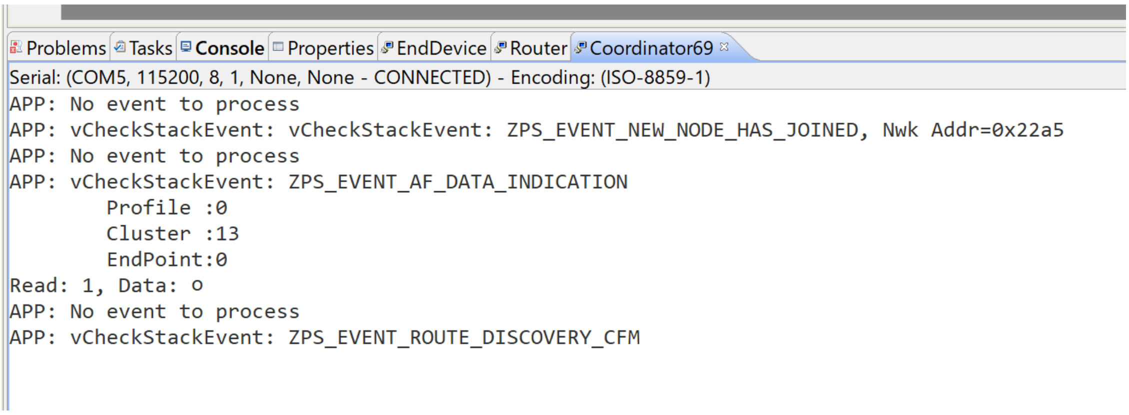


```
CDT Build Console [JN-AN-1229-ZPS-Application-Template]
../../AN1229_ZBP_SleepingEndDevice/Source/app_sleeping_enddevice.c:101:56: warning: 'apduMyData' is used uninitialized in this function [-Wuni

Linking AN1229_ZBP_SleepingEndDevice_JN5169.elf ...
/c/NXP/bstudio_nxp/sdk/Tools/ba-elf-ba2-r36379/bin/ba-elf-gcc -march=ba2 -mcpu=jn51xx -mredzone-size=4 -mbranch-cost=3 -fomit-frame-pointer -C
/c/NXP/bstudio_nxp/sdk/Tools/ba-elf-ba2-r36379/bin/ba-elf-size AN1229_ZBP_SleepingEndDevice_JN5169.elf
   text    data     bss     dec     hex filename
  98247    1248   17773  117268   1ca14 AN1229_ZBP_SleepingEndDevice_JN5169.elf
Generating binary ...
/c/NXP/bstudio_nxp/sdk/Tools/ba-elf-ba2-r36379/bin/ba-elf-objcopy -S -O binary AN1229_ZBP_SleepingEndDevice_JN5169.elf AN1229_ZBP_SleepingEndD

10:56:51 Build Finished (took 12s.171ms)
```

勝手に apduMyData　を uint8　にしたが, uint16 でも良かったのか

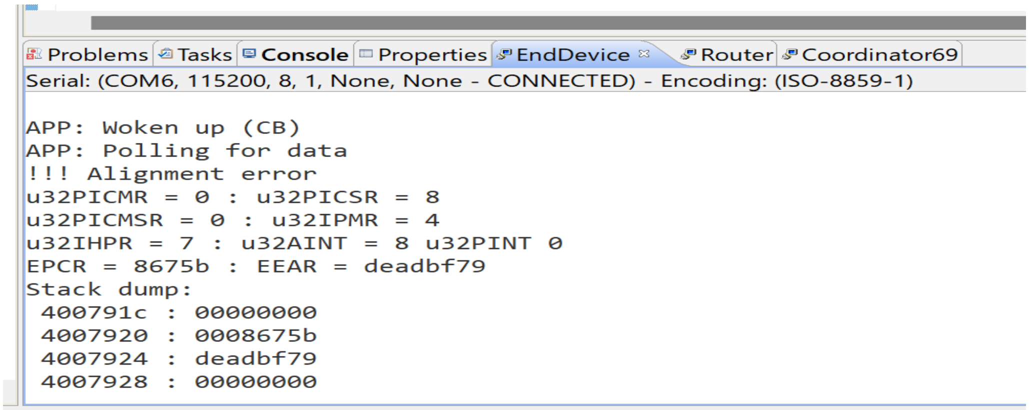
次に monostick にそれぞれのコードを入れて確認した.

Coordinator

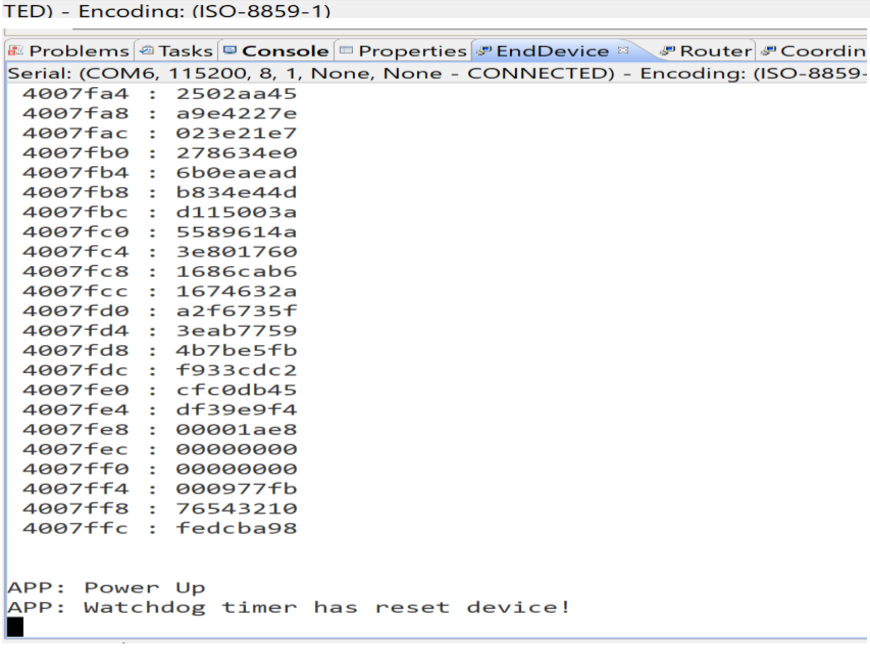


```
Serial: (COM5, 115200, 8, 1, None, None - CONNECTED) - Encoding: (ISO-8859-1)
APP: No event to process
APP: vCheckStackEvent: vCheckStackEvent: ZPS_EVENT_NEW_NODE_HAS_JOINED, Nwk Addr=0x22a5
APP: No event to process
APP: vCheckStackEvent: ZPS_EVENT_AF_DATA_INDICATION
        Profile :0
        Cluster :13
        EndPoint:0
Read: 1, Data: o
APP: No event to process
APP: vCheckStackEvent: ZPS_EVENT_ROUTE_DISCOVERY_CFM
```

Enddevice



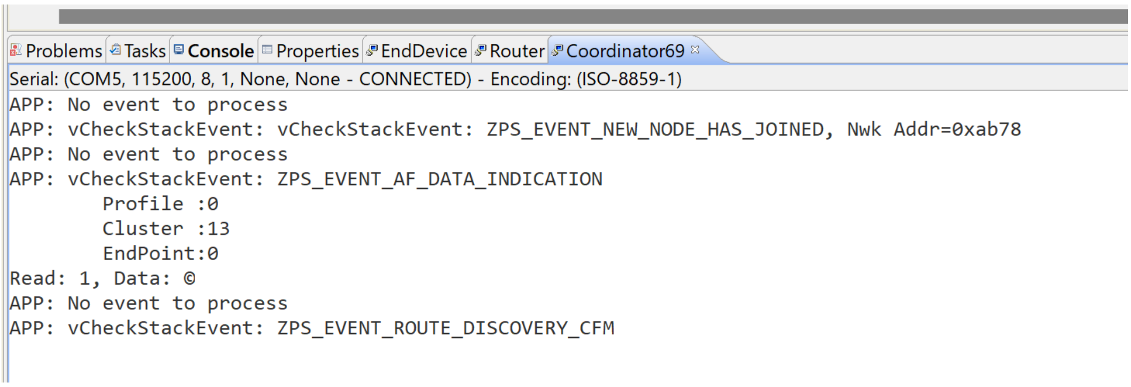
```
Serial: (COM6, 115200, 8, 1, None, None - CONNECTED) - Encoding: (ISO-8859-1)

APP: Woken up (CB)
APP: Polling for data
!!! Alignment error
u32PICMR = 0 : u32PICSR = 8
u32PICMSR = 0 : u32IPMR = 4
u32IHPR = 7 : u32AINT = 8 u32PINT 0
EPCR = 8675b : EEAR = deadbf79
Stack dump:
 400791c : 00000000
 4007920 : 0008675b
 4007924 : deadbf79
 4007928 : 00000000
```



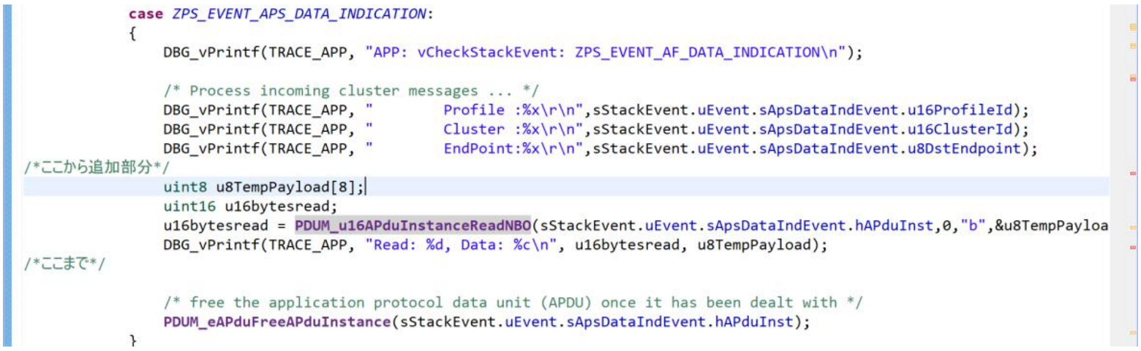
```
Serial: (COM6, 115200, 8, 1, None, None - CONNECTED) - Encoding: (ISO-8859-
 4007fa4 : 2502aa45
 4007fa8 : a9e4227e
 4007fac : 023e21e7
 4007fb0 : 278634e0
 4007fb4 : 6b0eaead
 4007fb8 : b834e44d
 4007fbc : d115003a
 4007fc0 : 5589614a
 4007fc4 : 3e801760
 4007fc8 : 1686cab6
 4007fcc : 1674632a
 4007fd0 : a2f6735f
 4007fd4 : 3eab7759
 4007fd8 : 4b7be5fb
 4007fdc : f933cdc2
 4007fe0 : cfc0db45
 4007fe4 : df39e9f4
 4007fe8 : 00001ae8
 4007fec : 00000000
 4007ff0 : 00000000
 4007ff4 : 000977fb
 4007ff8 : 76543210
 4007ffc : fedcba98


APP: Power Up
APP: Watchdog timer has reset device!
```

２回目では enddevice での表示内容は上記と同じ.

Coordinator では, Data 内容が変化した.



```
Serial: (COM5, 115200, 8, 1, None, None - CONNECTED) - Encoding: (ISO-8859-1)
APP: No event to process
APP: vCheckStackEvent: vCheckStackEvent: ZPS_EVENT_NEW_NODE_HAS_JOINED, Nwk Addr=0xab78
APP: No event to process
APP: vCheckStackEvent: ZPS_EVENT_AF_DATA_INDICATION
        Profile :0
        Cluster :13
        EndPoint:0
Read: 1, Data: ©
APP: No event to process
APP: vCheckStackEvent: ZPS_EVENT_ROUTE_DISCOVERY_CFM
```



```
            case ZPS_EVENT_APS_DATA_INDICATION:
            {
                DBG_vPrintf(TRACE_APP, "APP: vCheckStackEvent: ZPS_EVENT_AF_DATA_INDICATION\n");

                /* Process incoming cluster messages ... */
                DBG_vPrintf(TRACE_APP, "        Profile :%x\r\n",sStackEvent.uEvent.sApsDataIndEvent.u16ProfileId);
                DBG_vPrintf(TRACE_APP, "        Cluster :%x\r\n",sStackEvent.uEvent.sApsDataIndEvent.u16ClusterId);
                DBG_vPrintf(TRACE_APP, "        EndPoint:%x\r\n",sStackEvent.uEvent.sApsDataIndEvent.u8DstEndpoint);
/*ここから追加部分*/
                uint8 u8TempPayload[8];
                uint16 u16bytesread;
                u16bytesread = PDUM_u16APduInstanceReadNBO(sStackEvent.uEvent.sApsDataIndEvent.hAPduInst,0,"b",&u8TempPayloa
                DBG_vPrintf(TRACE_APP, "Read: %d, Data: %c\n", u16bytesread, u8TempPayload);
/*ここまで*/

                /* free the application protocol data unit (APDU) once it has been dealt with */
                PDUM_eAPduFreeAPduInstance(sStackEvent.uEvent.sApsDataIndEvent.hAPduInst);
            }
```

uint8 u8TempPayload; →uint8 u8TempPayload[8]; に変更した.

Coordinator の terminal の結果は

Read: 1, Data: d　　他は同じ

app_coordinator.c の

u16bytesread = PDUM_u16ApduInstanceReadNBO(sStackEvent.uEvent.sApsDataIndEvent.hAPduInst,0, "bbbbbb",&u8TempPayload);

のように, “b”を“bbbbbb”と変更すると,

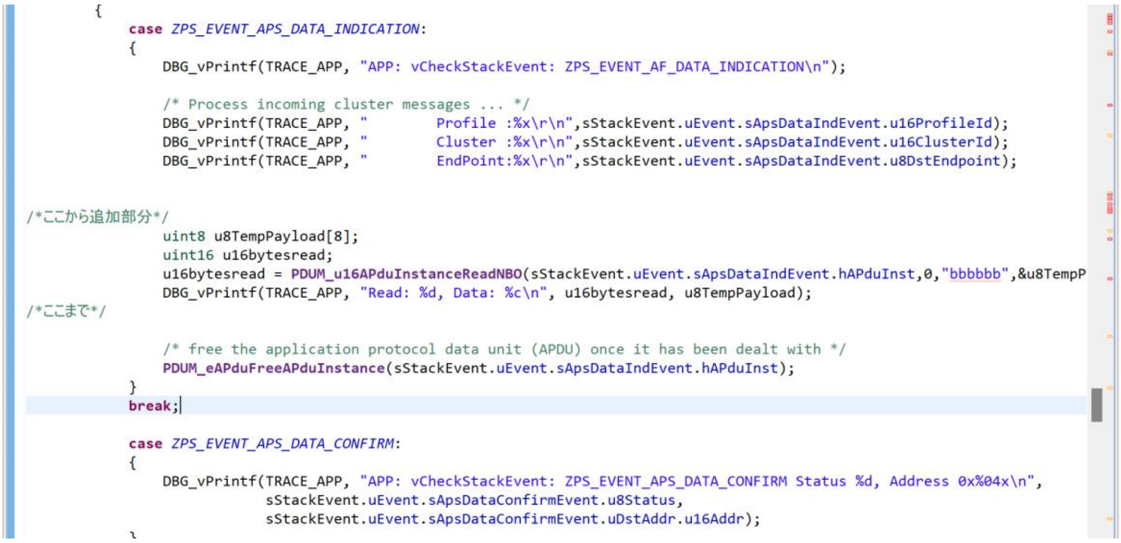
結果は

Read: 6, Data: d　　他は同じ

[已解决: Re: Simple Data transfer - 第 2 页 - NXP Community]

[How to send data from Coordinator to Sleeping End ... - NXP Community]

上記まではcoordinatorにしか画像のようなコードを入れていなかったが,
Enddeviceにもこのコードを入れて試した.
しかし、結果は同じだった.



```
        {
            case ZPS_EVENT_APS_DATA_INDICATION:
            {
                DBG_vPrintf(TRACE_APP, "APP: vCheckStackEvent: ZPS_EVENT_AF_DATA_INDICATION\n");

                /* Process incoming cluster messages ... */
                DBG_vPrintf(TRACE_APP, "        Profile :%x\r\n",sStackEvent.uEvent.sApsDataIndEvent.u16ProfileId);
                DBG_vPrintf(TRACE_APP, "        Cluster :%x\r\n",sStackEvent.uEvent.sApsDataIndEvent.u16ClusterId);
                DBG_vPrintf(TRACE_APP, "        EndPoint:%x\r\n",sStackEvent.uEvent.sApsDataIndEvent.u8DstEndpoint);


/*ここから追加部分*/
                uint8 u8TempPayload[8];
                uint16 u16bytesread;
                u16bytesread = PDUM_u16APduInstanceReadNBO(sStackEvent.uEvent.sApsDataIndEvent.hAPduInst,0,"bbbbbb",&u8TempP
                DBG_vPrintf(TRACE_APP, "Read: %d, Data: %c\n", u16bytesread, u8TempPayload);
/*ここまで*/

                /* free the application protocol data unit (APDU) once it has been dealt with */
                PDUM_eAPduFreeAPduInstance(sStackEvent.uEvent.sApsDataIndEvent.hAPduInst);
            }
            break;

            case ZPS_EVENT_APS_DATA_CONFIRM:
            {
                DBG_vPrintf(TRACE_APP, "APP: vCheckStackEvent: ZPS_EVENT_APS_DATA_CONFIRM Status %d, Address 0x%04x\n",
                            sStackEvent.uEvent.sApsDataConfirmEvent.u8Status,
                            sStackEvent.uEvent.sApsDataConfirmEvent.uDstAddr.u16Addr);
            }
```

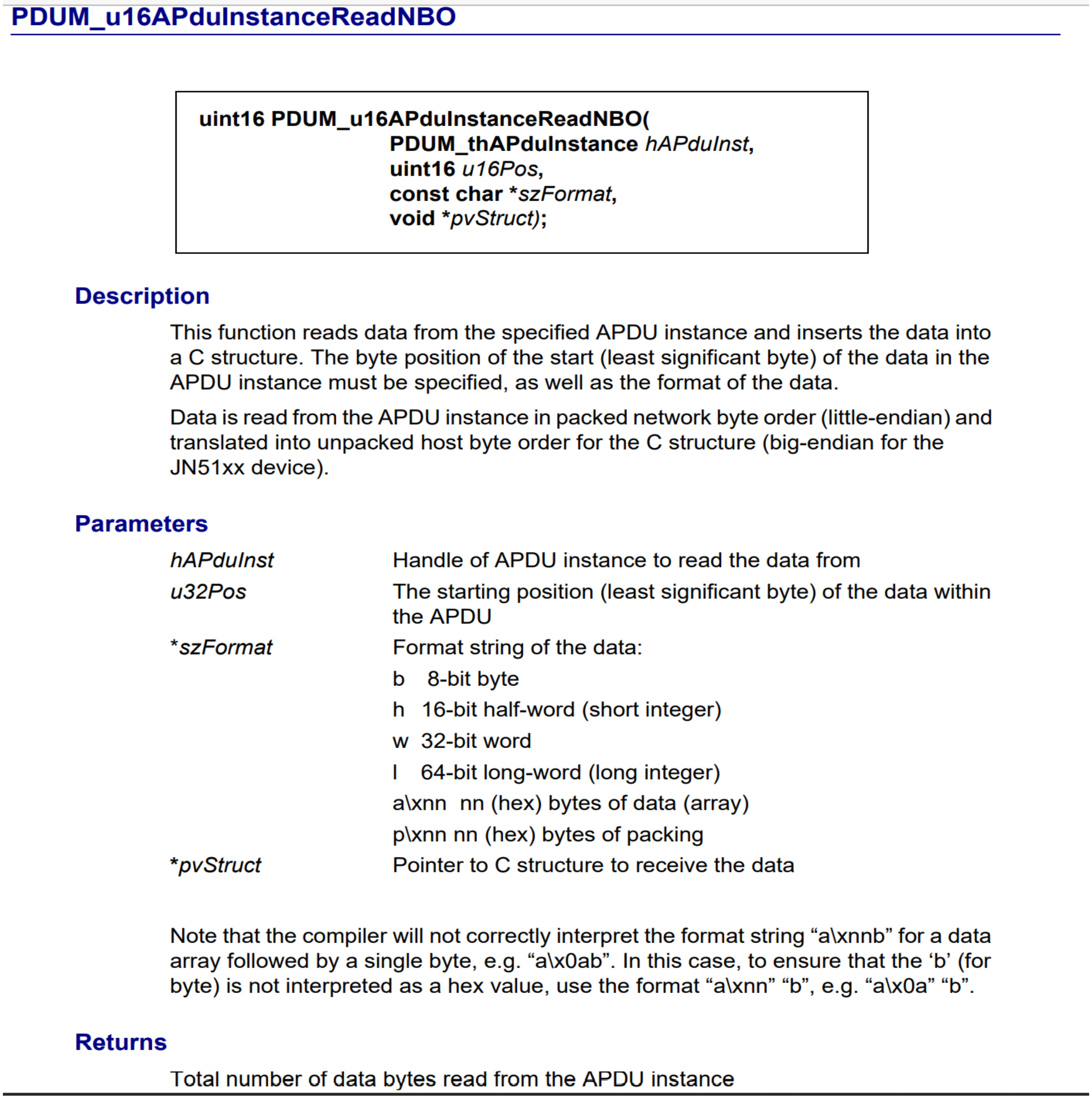
ZigBee_3.0_Stack_User_Guide_v1.5.pdfより

・PDUM_u16APduInstanceReadNBO()について
　Once a data packet has been collected from a message queue, the data can be extracted from the APDU instance using the PDUM function PDUM_u16APduInstanceReadNBO(). The APDU instance must then be released using the PDUM function PDUM_eAPduFreeAPduInstance()

・PDUM_eAPduFreeAPduInstance()について

If the request is not successfully sent (the send function does not return ZPS_E_SUCCESS) then the APDU instance will not be automatically de-allocated and the application should de-allocate it using the PDUM function PDUM_eAPduFreeAPduInstance().

When a response is subsequently received, the stack automatically allocates a local APDU instance and includes its handle in the notification event for the response. Once the response has been dealt with, the application must de-allocate the APDU instance using the function PDUM_eAPduFreeAPduInstance()



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

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