

12/23

- ① “send”でデータ送信を制御することは、原因が分からないため、調査中
- ② Coordinator のアドレスを取得して、Enddevice から Coordinator の通信をユニキャスト通信でできるようにする。

ユニキャスト通信に変更して、Enddevice と coordinator の両方の terminal に “FromEnddevice”と表示されれば成功とする。

参考 URL: [Re: How to send Data from SED to Coordinator on JN... - NXP Community](#)

とブロードキャスト関数を参考にし、Enddevice に ZPS_eAplAfUnicastIeeeDataReq 関数を使用。

**ZigBee PRO Stack
User Guide**

ZPS_eAplAfUnicastIeeeDataReq

```
ZPS_teStatus ZPS_eAplAfUnicastIeeeDataReq(  
    PDUM_thAPduInstance hAPduInst,  
    uint16 u16ClusterId,  
    uint8 u8SrcEndpoint,  
    uint8 u8DstEndpoint,  
    uint64 u64DestAddr,  
    ZPS_teAplAfSecurityMode eSecurityMode,  
    uint8 u8Radius,  
    uint8 *pu8SeqNum);
```

第5引数以外はブロードキャストの関数と同じなので、再利用した。

Coordinator の 64bit の Mac アドレスを調べると、1bc501 と表示。

→これは Enddevice の Mac アドレスと同じになっている。

コンパイルを通るが, Enddevice の terminal に送信したことを意味する confirm が表示されない. また Coordinator の terminal には“FromEnddevice”と表示されなかった.

→送受信していない.

“send OK”も表示されるため, 実行はされている.

```
DBG_vPrintf(TRUE, "MyMacAddress=%x\n", mac);

if (hAPduInst == PDUM_INVALID_HANDLE)
{
    DBG_vPrintf(TRUE, "PDUM_INVALID_HANDLE\n");
} else {

    ZPS_teStatus eStatus;
    ZPS_teAplAfSecurityMode eSecurityMode = (ZPS_E_APL_AF_UNSECURE); //セキュリティ無効

    eStatus = ZPS_eAplAfUnicastIeeeDataReq(
        hAPduInst,
        0x1337,
        0x01,
        0x01,
        0x1bc501, //Dest: coordinator
        eSecurityMode,
        0,
        &u8TransactionSequenceNumber
    );

    DBG_vPrintf(TRUE, "send OK\n");
    /*
    eStatus = ZPS_eAplAfBroadcastDataReq(
        hAPduInst,
        0x1337,
        0x01,
        0x01,
        ZPS_E_BROADCAST_ZC_ZR, //Dest: All Coordinator & Routers
        eSecurityMode,
        0,
        &u8TransactionSequenceNumber
    );
    */
}
```

URL のサイトでは,

ZPS_eAplAfUnicastIeeeDataReq 関数の宛先は 64bit ではなく 16bit であったので,

```
uint16 mac16;
mac16 = ZPS_u64AplZdoGetIeeeAddr();
DBG_vPrintf(TRUE, "MyMacAddress=%x\n", mac16);
```

再度 Coordinator の 16bit の Mac アドレスを調べると 6bd5 であった.

なので, ユニキャスト通信の関数の引数を 0x6bd5 とした.

結果, 先ほどと変化なく, 送受信できていない.

ZPS_eAplAfUnicastDataReq 関数に変更した。

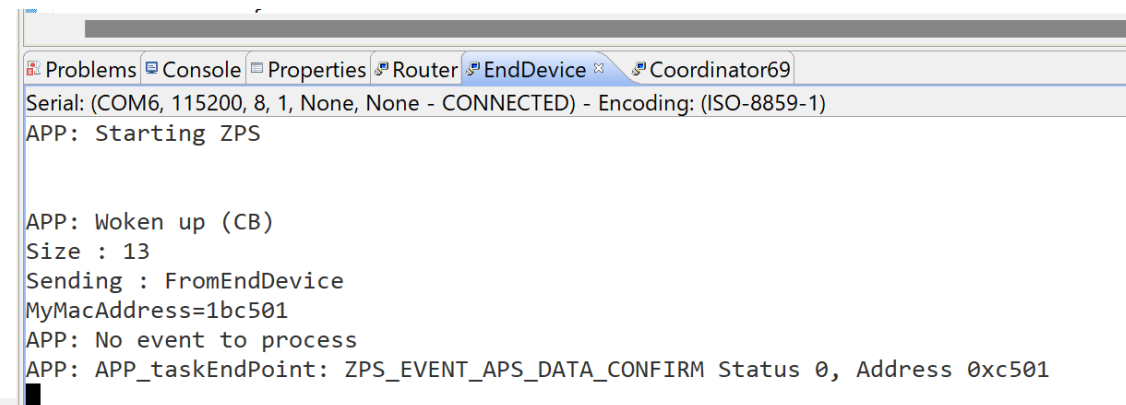
ZPS_eAplAfUnicastDataReq

```
ZPS_teStatus ZPS_eAplAfUnicastDataReq(  
    PDUM_thAPdulInstance hAPdulInst,  
    uint16 u16ClusterId,  
    uint8 u8SrcEndpoint,  
    uint8 u8DstEndpoint,  
    uint16 u16DestAddr,  
    ZPS_teAplAfSecurityMode eSecurityMode,  
    uint8 u8Radius,  
    uint8 *pu8SeqNum);
```

Parameters

<i>hAPdulInst</i>	Handle of APDU instance to be sent
<i>u16ClusterId</i>	Identifier of relevant output cluster on source endpoint
<i>u8SrcEndpoint</i>	Source endpoint number (1-240) on local node
<i>u8DstEndpoint</i>	Destination endpoint number (1-240) on remote node
<i>u16DestAddr</i>	Network address of destination node

以下が結果で, confirm が表示された。



The screenshot shows a serial terminal window with the following text:

```
Serial: (COM6, 115200, 8, 1, None, None - CONNECTED) - Encoding: (ISO-8859-1)  
APP: Starting ZPS  
  
APP: Woken up (CB)  
Size : 13  
Sending : FromEndDevice  
MyMacAddress=1bc501  
APP: No event to process  
APP: APP_taskEndPoint: ZPS_EVENT_APS_DATA_CONFIRM Status 0, Address 0xc501
```

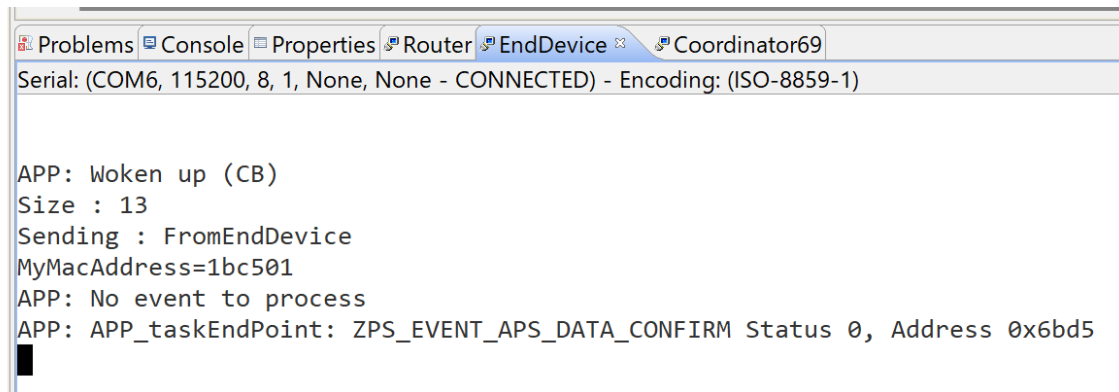
以下が Confirm が表示されているコード

```
case ZPS_EVENT_APS_DATA_CONFIRM: //何かしら他端末にデータを送信したことを自分に報告  
{  
    DBG_vPrintf	TRACE_APP, "APP: APP_taskEndPoint: ZPS_EVENT_APS_DATA_CONFIRM Status %d, Address 0x%04x\n",  
        sStackEvent.uEvent.sApsDataConfirmEvent.u8Status,  
        sStackEvent.uEvent.sApsDataConfirmEvent.uDstAddr.u16Addr);  
}  
break;
```

なので、宛先を示す引数の値を 0x1bc501→0xc501 と変更.

結果は Coordinator の terminal は変化なし.

6bd5 に変更して、再度接続した.



```
Problems Console Properties Router EndDevice Coordinator69
Serial: (COM6, 115200, 8, 1, None, None - CONNECTED) - Encoding: (ISO-8859-1)

APP: Woken up (CB)
Size : 13
Sending : FromEndDevice
MyMacAddress=1bc501
APP: No event to process
APP: APP_taskEndPoint: ZPS_EVENT_APS_DATA_CONFIRM Status 0, Address 0x6bd5
```

Coordinator の表示には変化なし.

Confirm で表示される宛先のアドレスはユニキャスト通信の関数で指定したアドレスになることが分かった.

なので、正しい Coordinatoor のアドレスではない.

- ・第 5 引数の違い

ZPS_eAplAfUnicastDataReq 関数は 16bit のネットワークアドレス

ZPS_eAplAfUnicastIeeeDataReq 関数は 64bit の Mac アドレス

→64bit の Mac アドレスを使う ZPS_eAplAfUnicastIeeeDataReq 関数の方が正しいの考えるが, ZPS_eAplAfUnicastDataReq 関数は送信したことを示す confirm が表示されるので, 調査中

- ・ URL の PDUM_u16APduInstanceWriteNBO 関数がブロードキャスト通信の時に引数の数が違う. JenOS User guide.pdf では引数は ... になっている.

PDUM_u16APduInstanceWriteNBO

```
uint16 PDUM_u16APduInstanceWriteNBO(
    PDUM_thAPduInstance hAPduInst,
    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 JN516x device).

Parameters

<i>hAPduInst</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".