12/16　　JN-AN-1174 を参考

・coordinatorTerminal で“send” を打ち込むと“FromCoordinator”が両方の Terminal で表示できるようにする.

・Enddevice も同様に“FromEnddevice” と表示できるようにする.

なので, Terminal では接続後, 何も表示されずに command(send)入力を待機しているようにしておく.

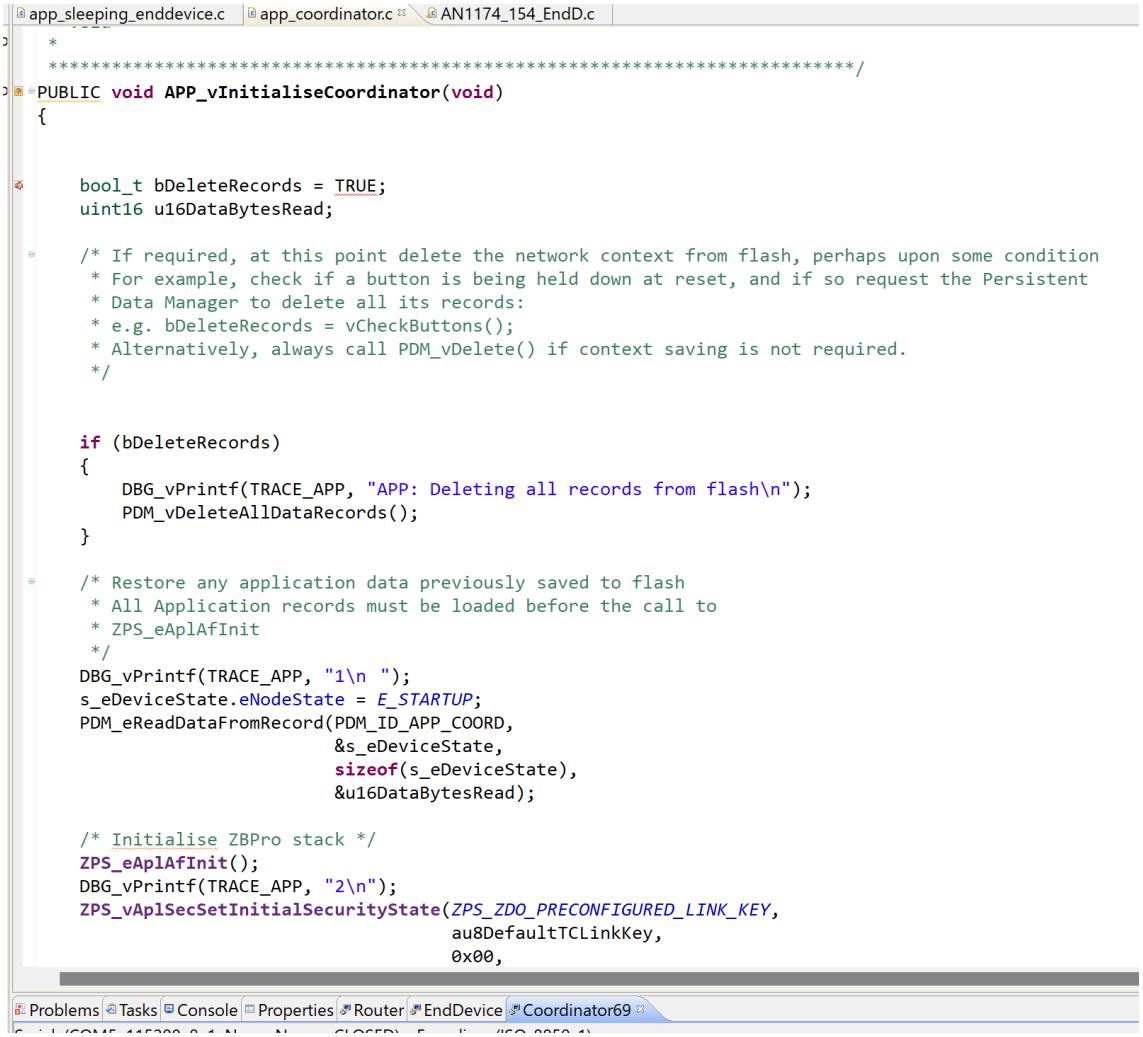
○Coordinator のコードを追加・変更を行う.

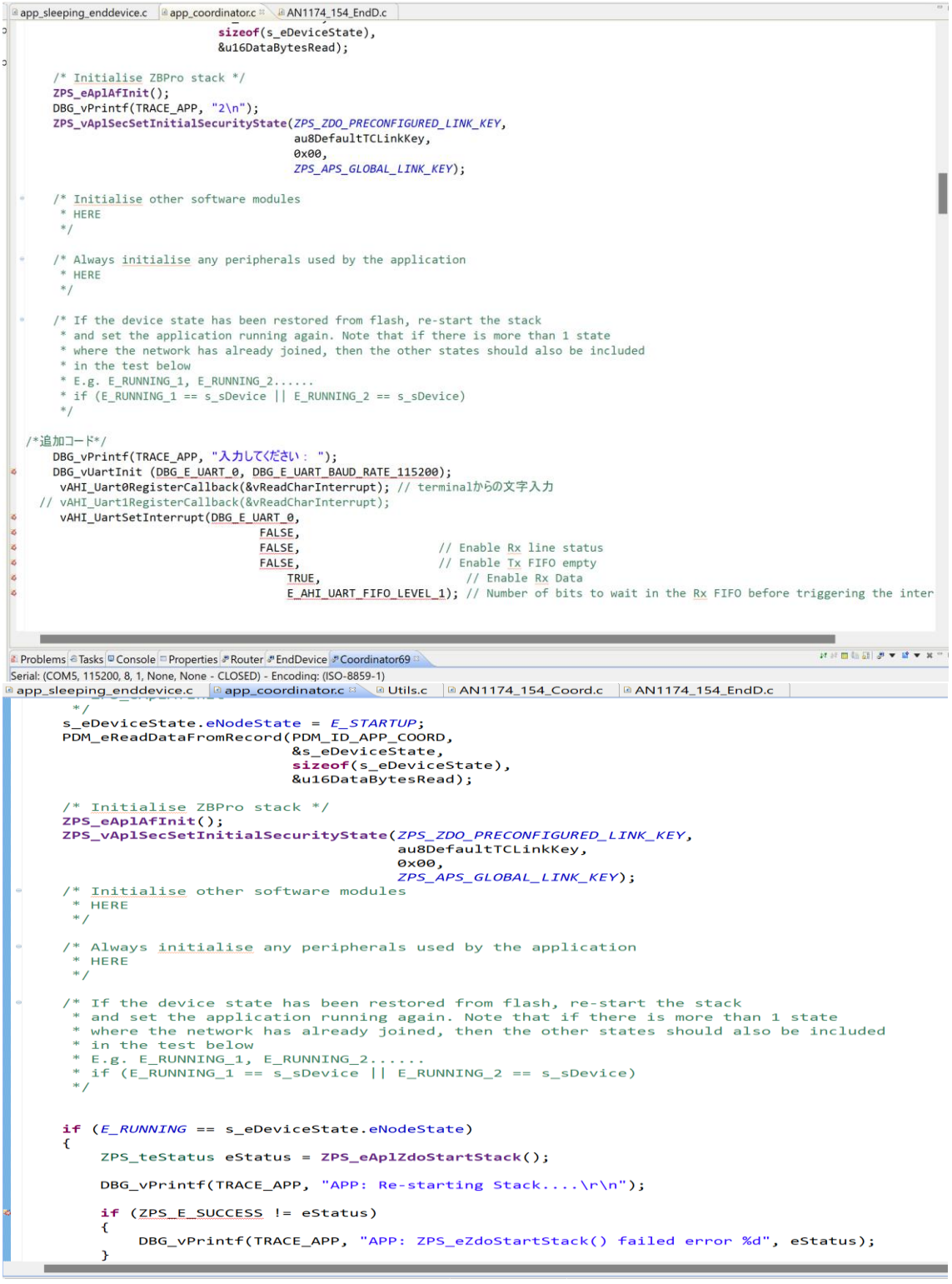
初期化を行う関数に terminal からの文字入力を読み取るコードを追加.

入力文字を terminal から読み込んで MONOSTICK(Coordinator)に送信する経路を意味する UART は起動して最初に実行される初期化関数 initialize に記述する.

Loop 関数と平行して UART は動作する.

文字を入力すると UART のブァッファに保存され, send+enter キーが入力されると, Loop 関数にある Command で処理される.





```
app_sleeping_enddevice.c | app_coordinator.c | AN1174_154_EndD.c

                            sizeof(s_eDeviceState),
                            &u16DataBytesRead);

    /* Initialise ZBPro stack */
    ZPS_eAplAfInit();
    DBG_vPrintf(TRACE_APP, "2\n");
    ZPS_vAplSecSetInitialSecurityState(ZPS_ZDO_PRECONFIGURED_LINK_KEY,
                                       au8DefaultTCLinkKey,
                                       0x00,
                                       ZPS_APS_GLOBAL_LINK_KEY);

    /* Initialise other software modules
     * HERE
     */

    /* Always initialise any peripherals used by the application
     * HERE
     */

    /* If the device state has been restored from flash, re-start the stack
     * and set the application running again. Note that if there is more than 1 state
     * where the network has already joined, then the other states should also be included
     * in the test below
     * E.g. E_RUNNING_1, E_RUNNING_2......
     * if (E_RUNNING_1 == s_sDevice || E_RUNNING_2 == s_sDevice)
     */

/*追加コード*/
    DBG_vPrintf(TRACE_APP, "入力してください： ");
    DBG_vUartInit (DBG_E_UART_0, DBG_E_UART_BAUD_RATE_115200);
     vAHI_Uart0RegisterCallback(&vReadCharInterrupt); // terminalからの文字入力
  // vAHI_Uart1RegisterCallback(&vReadCharInterrupt);
    vAHI_UartSetInterrupt(DBG_E_UART_0,
                          FALSE,
                          FALSE,                    // Enable Rx line status
                          FALSE,                    // Enable Tx FIFO empty
                              TRUE,                     // Enable Rx Data
                          E_AHI_UART_FIFO_LEVEL_1); // Number of bits to wait in the Rx FIFO before triggering the inter
```

Problems | Tasks | Console | Properties | Router | EndDevice | Coordinator69

Serial: (COM5, 115200, 8, 1, None, None - CLOSED) - Encoding: (ISO-8859-1)

app_sleeping_enddevice.c | app_coordinator.c | Utils.c | AN1174_154_Coord.c | AN1174_154_EndD.c

```
     */
    s_eDeviceState.eNodeState = E_STARTUP;
    PDM_eReadDataFromRecord(PDM_ID_APP_COORD,
                            &s_eDeviceState,
                            sizeof(s_eDeviceState),
                            &u16DataBytesRead);

    /* Initialise ZBPro stack */
    ZPS_eAplAfInit();
    ZPS_vAplSecSetInitialSecurityState(ZPS_ZDO_PRECONFIGURED_LINK_KEY,
                                       au8DefaultTCLinkKey,
                                       0x00,
                                       ZPS_APS_GLOBAL_LINK_KEY);
    /* Initialise other software modules
     * HERE
     */

    /* Always initialise any peripherals used by the application
     * HERE
     */

    /* If the device state has been restored from flash, re-start the stack
     * and set the application running again. Note that if there is more than 1 state
     * where the network has already joined, then the other states should also be included
     * in the test below
     * E.g. E_RUNNING_1, E_RUNNING_2......
     * if (E_RUNNING_1 == s_sDevice || E_RUNNING_2 == s_sDevice)
     */


    if (E_RUNNING == s_eDeviceState.eNodeState)
    {
        ZPS_teStatus eStatus = ZPS_eAplZdoStartStack();

        DBG_vPrintf(TRACE_APP, "APP: Re-starting Stack....\r\n");

        if (ZPS_E_SUCCESS != eStatus)
        {
            DBG_vPrintf(TRACE_APP, "APP: ZPS_eZdoStartStack() failed error %d", eStatus);
        }
```

app_sleeping_enddevice.c | app_coordinator.c | Utils.c | AN1174_154_Coord.c | AN1174_154_EndD.c

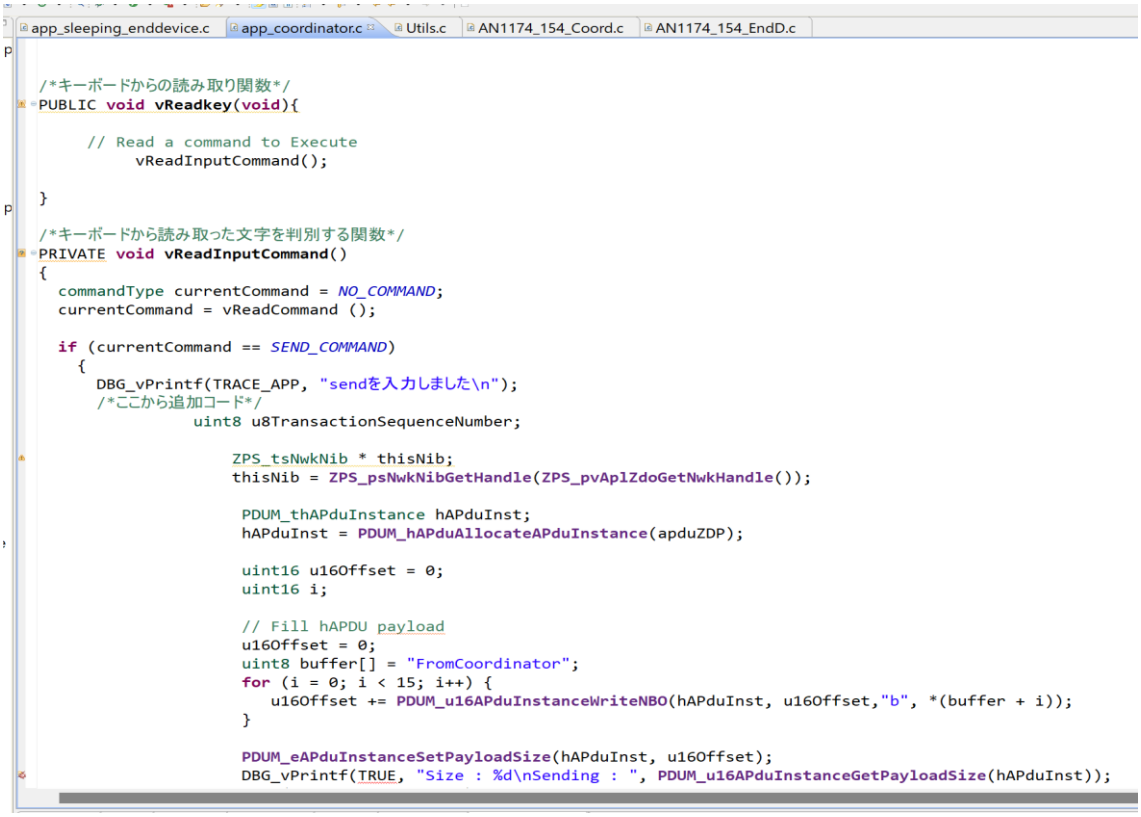
```
    if (E_RUNNING == s_eDeviceState.eNodeState)
    {
        ZPS_teStatus eStatus = ZPS_eAplZdoStartStack();

        DBG_vPrintf(TRACE_APP, "APP: Re-starting Stack....\r\n");

        if (ZPS_E_SUCCESS != eStatus)
        {
            DBG_vPrintf(TRACE_APP, "APP: ZPS_eZdoStartStack() failed error %d", eStatus);
        }

        /* Turn on joining */
        ZPS_eAplZdoPermitJoining(0xff);


        /* Re-start any other application software modules
         * HERE
         */
    }
    else /* perform any actions require on initial start-up */
    {
        /* Return the device to the start-up start if it was reset during the network formation stage */
        s_eDeviceState.eNodeState = E_STARTUP;
    }
}

/****************************************************************************
 *
 * NAME: APP_vtaskCoordinator
 *
 * DESCRIPTION:
 * Main state machine
 *
 * RETURNS:
 * void
 *
 ****************************************************************************/
PUBLIC void APP_vtaskCoordinator ( void )
{
```

入力文字を解析するコードは E_RUNNING に追加した.

app_sleeping_enddevice.c | app_coordinator.c | Utils.c | AN1174_154_Coord.c | AN1174_154_EndD.c

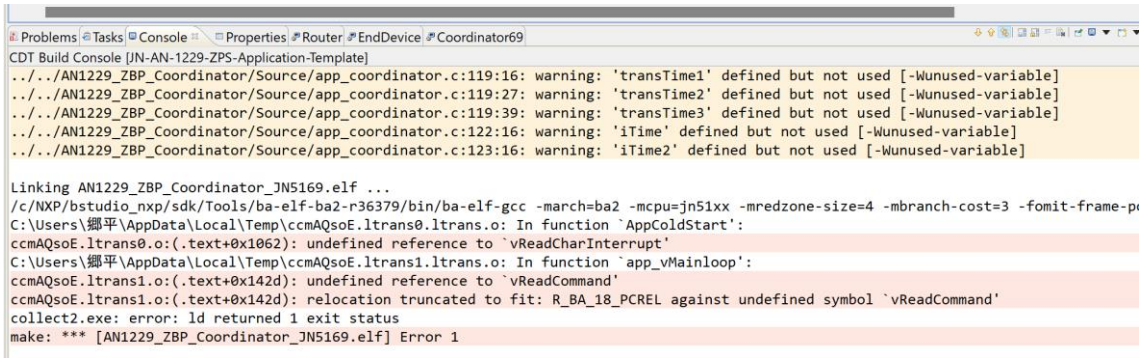
```
PUBLIC void APP_vtaskCoordinator ( void )
{

    ZPS_tsAfEvent sStackEvent;
    sStackEvent.eType = ZPS_EVENT_NONE;

    /* check if any messages to collect */
    if ( ZQ_bQueueReceive(&APP_msgZpsEvents, &sStackEvent) )
    {
        //DBG_vPrintf(TRACE_APP, "APP: No event to process\n"); コメントアウト
    }

    if ( ZTIMER_eGetState(u8App_tmr1sec) == E_ZTIMER_STATE_EXPIRED )
    {
        ZTIMER_eStart (u8App_tmr1sec, ZTIMER_TIME_SEC(1) );
    }

    switch (s_eDeviceState.eNodeState)
    {
        case E_STARTUP:
        {
            vStartup();
        }
        break;

        case E_NETWORK_FORMATION:
        {
            vWaitForNetworkFormation(sStackEvent);
        }
        break;

        case E_RUNNING:
        {

            vHandleStackEvent(sStackEvent);
            vReadkey(); //キーボードからの入力を読み取る
        }
        break;

        default:
```

Utils.c



```
/*キーボードからの読み取り関数*/
PUBLIC void vReadkey(void){

    // Read a command to Execute
        vReadInputCommand();

}
/*キーボードから読み取った文字を判別する関数*/
PRIVATE void vReadInputCommand()
{
  commandType currentCommand = NO_COMMAND;
  currentCommand = vReadCommand ();

  if (currentCommand == SEND_COMMAND)
    {
      DBG_vPrintf(TRACE_APP, "sendを入力しました\n");
      /*ここから追加コード*/
                uint8 u8TransactionSequenceNumber;

                    ZPS_tsNwkNib * thisNib;
                    thisNib = ZPS_psNwkNibGetHandle(ZPS_pvAplZdoGetNwkHandle());

                     PDUM_thAPduInstance hAPduInst;
                     hAPduInst = PDUM_hAPduAllocateAPduInstance(apduZDP);

                     uint16 u16Offset = 0;
                     uint16 i;

                     // Fill hAPDU payload
                     u16Offset = 0;
                     uint8 buffer[] = "FromCoordinator";
                     for (i = 0; i < 15; i++) {
                        u16Offset += PDUM_u16APduInstanceWriteNBO(hAPduInst, u16Offset,"b", *(buffer + i));
                     }

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

以下がエラー内容である.



```
Problems Tasks Console Properties Router EndDevice Coordinator69
CDT Build Console [JN-AN-1229-ZPS-Application-Template]
../../AN1229_ZBP_Coordinator/Source/app_coordinator.c:119:16: warning: 'transTime1' defined but not used [-Wunused-variable]
../../AN1229_ZBP_Coordinator/Source/app_coordinator.c:119:27: warning: 'transTime2' defined but not used [-Wunused-variable]
../../AN1229_ZBP_Coordinator/Source/app_coordinator.c:119:39: warning: 'transTime3' defined but not used [-Wunused-variable]
../../AN1229_ZBP_Coordinator/Source/app_coordinator.c:122:16: warning: 'iTime' defined but not used [-Wunused-variable]
../../AN1229_ZBP_Coordinator/Source/app_coordinator.c:123:16: warning: 'iTime2' defined but not used [-Wunused-variable]

Linking AN1229_ZBP_Coordinator_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-p
C:\Users\郷平\AppData\Local\Temp\ccmAQsoE.ltrans0.ltrans.o: In function `AppColdStart':
ccmAQsoE.ltrans0.o:(.text+0x1062): undefined reference to `vReadCharInterrupt'
C:\Users\郷平\AppData\Local\Temp\ccmAQsoE.ltrans1.ltrans.o: In function `app_vMainloop':
ccmAQsoE.ltrans1.o:(.text+0x142d): undefined reference to `vReadCommand'
ccmAQsoE.ltrans1.o:(.text+0x142d): relocation truncated to fit: R_BA_18_PCREL against undefined symbol `vReadCommand'
collect2.exe: error: ld returned 1 exit status
make: *** [AN1229_ZBP_Coordinator_JN5169.elf] Error 1
```

Makefile に Utils.c を追加することで，エラーは解消した.
DBG_vPrintf(“1 ¥n”); など入れても, terminal に何も表示されない.

```
vHandleStackEvent(sStackEvent);
DBG_vPrintf(TRACE_APP, "vReadkey¥n");
vReadkey(); //キーボードからの入力を読み取る
```

このようにすると， vReadkey が表示されたので，キーボードからの文字を読み取っていないか，入力された文字を判別できていないか.

今まではブロードキャスト通信だったので，シリアル通信をするためにアドレスを取得する関数を使って，Enddevice のアドレス取得をした．
参考にしたサイト

Re: Using BeyondStudio for NXP JN5169 I can't seem... - NXP Community

ZPS_u64AplZdoGetIeeeAddr

```
uint64 ZPS_u64AplZdoGetIeeeAddr(void);
```

Description

This function obtains the 64-bit IEEE (MAC) address of the local node.

Parameters

None

Returns

64-bit IEEE/MAC address obtained

Enddevice に追加したコード

```
uint64 mac;
mac = ZPS_u64AplZdoGetIeeeAddr();
DBG_vPrintf(TRUE, "0x%04x¥n",mac);
```

実行結果では，0x1bc501 が表示された．

```
uint64 mac64=ZPS_u64AplZdoGetIeeeAddr();
uint32 mac32h=(uint32)(mac64>>32);
uint32 mac32l=(uint32)(mac64<<32);
DBG_vPrintf(TRUE, "mac64=0x%04X%04X¥n", mac32h, mac32l);
```

実行結果は，mac63=0xXX が表示された．

今後の予定

① Coordinator と Enddevice のアドレスを調べて，シリアル通信を構築する．

② Router を追加してマルチホップ通信にする.この時もシリアル通信で行う．

1）Coordinator から Router に, send コマンドと Enddevice のアドレスを送信

2）Router が宛先である Enddevice に送信する．

3）Enddevice は受信後，送信元の Coordinator(source)のアドレスをデータと共に得ているので, Router にそのアドレスと ACK データを送信する．

4）Router はそのアドレスを見て, Coordinator に送信する．