

【python から NXP1 にコマンドを送信して、NXP1 で受信すると受信したことを証明するために、ダミーデータを NXP2 送信する。そして NXP2 の方でデータ受信を確認できたら、pyhton から NXP1 に、コマンドを送信したことを示すことができる。】

Python(ラズパイ) → NXP1 → NXP2

現在の pyhton から、まとまった複数文字を送信していることが問題あるかもしれない。

そのため、失敗すれば、1

文字のコマンドを送信してテストしてみる。

補足として、キーボードからコマンド「Rx\_Data」もしくは「x」の入力は成功している。

#### ・送信コマンド「Rx\_Data」の場合

結果、NXP 2 で何も受信できていないため、python → NXP1 が失敗している。

#### ・送信コマンド「X」の場合（「x」も同じ）

以下のように python 側の送信コマンドを変更した。

結果は、NXP 2 で何も受信できていないため、失敗している。

```

19 print(line)
20 readSer.close()
21 print("==== Read Serial Complete ====\n")
22
23 # Write Serial
24 serialCommand = "X"
25 writeSer = serial.Serial(deviceName, baudrateNum, timeout=timeoutNum)
26 writeSer.write(serialCommand.encode())
27 writeSer.close()
28 print("==== Write Serial Complete ====\n")
29
30 print("==== End Program ====\n")
31
2 import serial
3 import struct
4 import binascii
5 import sys
6
7 use_port = '/dev/ttyUSB0'
8
9 _serial = serial.Serial(use_port)
10 _serial.baudrate = 9600
11 _serial.parity = serial.PARITY_NONE
12 _serial.bytesize = serial.EIGHTBITS
13 _serial.stopbits = serial.STOPBITS_ONE
14 _serial.timeout = 5 #sec
15
16 #commands = [ 0xB6, 0x01, 0x02, 0x00 ]
17
18 #for cmd in commands:
19 #    data = struct.pack("B", cmd)
20 #    print("tx: ", data)
21 #    _serial.write(data) #データ送信
22
23 serialCommand = "X"
24 #writeSer.write(serialCommand.encode())
25 _serial.write(serialCommand.encode()) #データ送信
26 _serial.flush() #データを送信するまで待機
27
28
29 rx = _serial.readline() #データ受信
30 print("rx: ", rx.hex()) #16進数文字列で表記 0x表記がなくなる
31
32 _serial.close()
33

```

・python の方で, シリアル通信速度を 9600bps であったため, 115200bps に変更した.  
変更理由は, NXP のプログラムを MONOSTICK に書き込む速度を 115200bps であったからである.

しかし, 結果は同じく, NXP2 で何も受信していない.

Python コードミスの可能性が高いが, NXP 側での受信に問題があるのではないかと考える.

```
void vReadCharInterrupt ()
{
    char inputChar = (char) u8AHI_UartReadData (DBG_E_UART_0);

    if (inputChar == 13)
    {
        // if inputchar is "Enter"
        memset(command,0,sizeof(command));
        strncpy(command,tmpString,64);
        memset(tmpString,0,sizeof(tmpString));
        indexCount = 0;
        DBG_vPrintf(TRUE,"\n");
        // DBG_vPrintf(TRUE,"\nCommand: %s \n",command);
    }
    else if (indexCount < 64)
    {
        //make sure we do not overflow
        tmpString[indexCount] = inputChar;
        indexCount++;
        //writes the input char on the terminal
        DBG_vPrintf(TRUE,"%c",inputChar);
    }
    else
    {
        // Clean temporal data string
        indexCount = 0;
        memset(tmpString,0,sizeof(tmpString));
    }
}
```

キーボード入力では, 打ち込んだ文字が u8AHI\_UartReadData(DBG\_E\_UART\_0)で読み込まれ, 変数 inputChar に代入される.

しかし, NXP の User guide では, 読み込む Uart 通信の関数はこの関数になっている.