

キーボード入力の場合、コマンド+エンターキー(13)を入力することで送信できていた。
Python コードにはエンターキーである 13 をコマンドを入れ込んでいなかったため、シリアル通信の実行ができていなかったのではないか、という結果に前回ミーティングは至った。

- 1) serialCommand = "x, 13"
 - 2) serialCommand1 = "x"
serialCommand2 = "13"
 - 3) serialCommnad = "x" で送信して, serialCommand = "13"
- の3パターンでテストしたが、送信されなかった。

またコマンドを受信する NXP 側において、以下のコードの vReadCgarInterrupt 関数でコマンド処理がされる。

```
/*追加コード*/
DBG_vUartInit (DBG_E_UART_0, DBG_E_UART_BAUD_RATE_115200);

vAHI_Uart0RegisterCallback(&vReadCharInterrupt); // terminalからの文字入力
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 in

DBG_vPrintf(TRACE_APP, "\n\nAPP: Power Up\n");
#if JENNIC_CHIP_FAMILY == JN516x
```

vReadCgarInterrupt 関数は vAHI_Uart0RegisterCallback 関数で実行されるため、この関数の中にあることが問題ではないかと考え、以下のように関数外に出した。

```
/*追加コード*/
DBG_vUartInit (DBG_E_UART_0, DBG_E_UART_BAUD_RATE_115200);

vReadCharInterrupt();

vAHI_Uart0RegisterCallback(&vReadCharInterrupt); // terminalからの文字入力
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 interrupt
```

しかし、結果として変化なく、コマンドは送信されなかった。
キーボード入力も UART 通信も割り込みなので、上記のような関数のままでよい。

以下のように python の serial.write() を変更することで、成功した。
17~20 行目のコードである。

```
4
5 # Set Parameter
6 deviceName = '/dev/ttyUSB0'
7 baudrateNum = 115200
8 timeoutNum = 3
9 print("==== Set Parameter Complete ====\\n")
10
11
12 # Write Serial
13 #serialCommand = "X"
14 writeSer = serial.Serial(deviceName, baudrateNum, timeout=timeoutNum)
15 #writeSer.write(serialCommand.encode())
16
17 writeSer.write(b'x\\r')
18 writeSer.flush()
19 writeSer.write(b'13\\r')
20 writeSer.flush()
21 writeSer.close()
22 print("==== Write Serial Complete ====\\n")
```

Python を実行することで、ラズパイに繋がっている MONOSTICK1 にラズパイから
コマンド(x)が送信され、受信すると MONOSTICK2 に送信する。

今後の解決すべき点

- ① 正しくセンサ値を取得できる回路を実装する
(現状として、何かしらの数値は受け取り表示ができています)
- ② コマンド + センサ値を送信し、MONOSTICK1 で受信し、そのセンサ値を
MONOSTICK2 に送信し、確認しなければならない。
(現状として MONOSTICK 1 → MONOSTICK 2 は、事前に決めた文字を送信している)