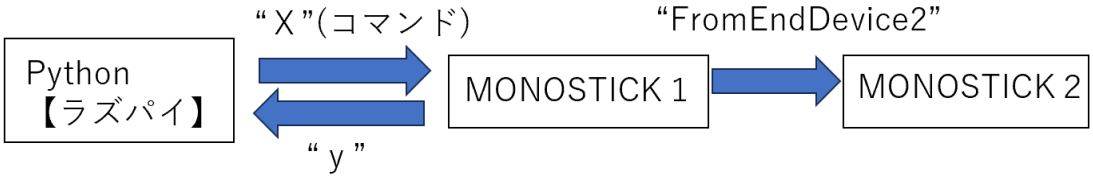
1102

○コマンド送信の確認を行う.




・MONOSTICK 1 (NXP)

```
///追加コマンド////
else if(strcmp(command, "x") == 0)
{
  //コマンドを受信したら確認として y を返す
    vAHI_UartWriteData (DBG_E_UART_0, 'y');
    vAHI_UartWriteData (DBG_E_UART_0, '\n');

    memset(command,0,sizeof(command));
    return RX_COMMAND;
}
//////////////
```

・ラズパイ(python)

```
import serial
import time

print("===== Start Program =====\n")

# Set Parameter
deviceName = '/dev/ttyUSB0'
baudrateNum = 115200
timeoutNum = 3
print("===== Set Parameter Complete =====\n")


writeSer = serial.Serial(deviceName, baudrateNum, timeout=timeoutNum)
writeSer.write(b'x\r\n')
writeSer.flush()
writeSer.write(b'13\r\n')
writeSer.flush()
writeSer.close()

print("===== Write Serial Complete =====\n")

readSer = serial.Serial(deviceName, baudrateNum, timeout=timeoutNum)
read_data = readSer.readline()
print(read_data)
readSer.close()

print("===== Read Serial Complete =====\n")

print("===== End Program =====\n")
```

・結果

```
KYOHEI@KYOHEI:~/Desktop $ python test.py
===== Start Program =====

===== Set Parameter Complete =====

===== Write Serial Complete =====

b'ding : FromEndDevice2\r\n'
===== Read Serial Complete =====

===== End Program =====
```

MONOSTICK 1 →MONOSTICK 2 に送信している文字列が読み取られた.
以下は sendData 関数で，x コマンドを受信すると， 文字列を送信する関数

```
PUBLIC void SendData(){

    /*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;
    u16Offset = 0;
    uint8 buffer[] = "FromEndDevice2";
    for (i = 0; i < 14; i++) {
        u16Offset += PDUM_u16APduInstanceWriteNBO(hAPduInst, u16Offset,"b", *(buffer + i));
    }
    PDUM_eAPduInstanceSetPayloadSize(hAPduInst, u16Offset);
    DBG_vPrintf(TRUE, "Size : %d\nSending : ", PDUM_u16APduInstanceGetPayloadSize(hAPduInst));
    for (i = 0; i < 14; i++) {
        DBG_vPrintf(TRUE, "%c", *(buffer + i));
    }
    DBG_vPrintf(TRUE, "\n");
```

予測する原因
MONOSTICK2（Coordinator）→MONOSTICK 1 (Enddevice)
はプーリング機能を解除しないとできない. 解除すると送信できるが, 今までの Enddeivce からのコマンドを送信できなくなる.(Enddeivce のスリープ機能より)

今まではアプリケーションの想定環境から Enddevice→Coordinaotr ができれば良いとしていた.

○センサ値を取得

センサ値の取得回数を 5 秒に 1 回にする.

```
31  ads1015_pin = 0 #ピン指定
32  startTime =0
33  nextTime =0
34
35  startTime = time.time()
36  while True:
37      data = 0
38      sum_data = 0
39      volt = 0
40      sum_volt = 0
41      count =0
42      avg_data = 0
43      avg_volt = 0
44
45
46      #ADS1015からデータを取得
47      data = ads.read_adc(ads1015_pin, gain=GAIN)
48      sum_data += data
49      volt = data * UNIT
50      sum_volt += volt
51      count = count + 1
52      nextTime = time.time() - startTime
53
54
55      if nextTime > 5:
56          avg_data = sum_data / count
57          avg_volt = sum_volt / count
58          print("受信データ：" + str(avg_data))
59          print("電圧　　：" + "{:.3f}".format(avg_volt))
60          startTime = time.time()
61          nextTime = 0
62          count = 0
63
64  #_serial.close()
```

```
KYOHEI@KYOHEI:~/Desktop $ python ADloopback.py
受信データ：1033.0
電圧　　：2.066
受信データ：989.0
電圧　　：1.978
受信データ：989.0
電圧　　：1.978
受信データ：1010.0
電圧　　：2.020
受信データ：1013.0
電圧　　：2.026
受信データ：962.0
電圧　　：1.924
受信データ：1001.0
電圧　　：2.002
受信データ：969.0
電圧　　：1.938
受信データ：975.0
電圧　　：1.950
受信データ：1013.0
電圧　　：2.026
^Z
[4]+  停止                 python ADloopback.py
```

光センサを使用しており，不正確の可能性があるが，手で覆うと受信データが 900 台になる．期待することは，極端に値が減少した表示になっていること．

○今後の予定

・正しい回路でセンサ値を取得する

→ x ＋センサ＋13　（ x は開始, 13 は終了）

→MONOSTICK で送受信できるようにする．