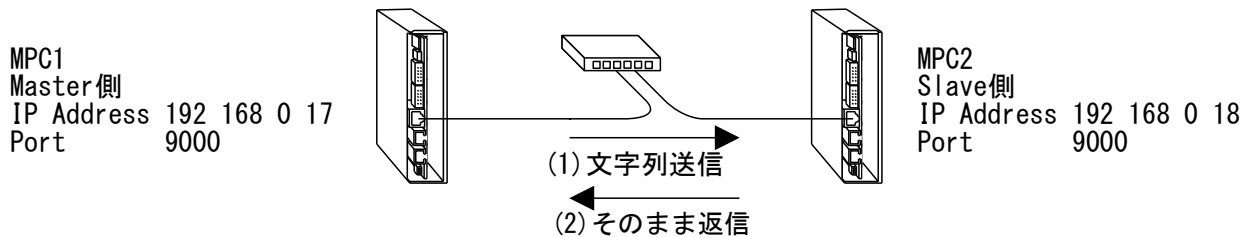


MPC間のUDP通信例

ti2k-221223 (Last update 221226)
MPC-3200(SH2A) BL/I 2.01_60 2022/12/21

簡単なUDP送受信例です。
Master側が送信した文字列をSlave側はそのまま返信します。
Master側は返信状態(タイムアウト、整合)をチェックします。



・ MPC1プログラム

```
SET_IP 192 168 0 17 255 255 255 0 192 168 0 248 /* 自己IP設定
SET_IP UDP PACKET 9000 /* UDP Port番号
IP_CONV 192 168 0 18 slv /* 変数 slv に相手のIPアドレスを入れる
SET_DEST slv UDP_XO$ /* 宛先設定

UDP_RO$="" /* 受信バッファクリア
c=0
tmout=0
ngcnt=0
DO
  snd$=STR$(c)+" "+TIME$(1)
  DO
    DO
      UDP_XO$=snd$ /* Slaveに文字列送信
      PR "snd$=" snd$
      timer_=10
      WAIT (UDP_RO$<>"") | (timer_==0)
      IF timer_>0 THEN
        BREAK /* Timeoutじゃない->受信文字列入力へ
      END_IF
      PR "Timeout"
      tmout=tmout+1
    LOOP /* Timeoutならリトライ
    rcvIP=IPA(UDP_RO$)&&hFF /* 送信元IPアドレス
    rcv$=UDP_RO$ /* 受信文字列
    UDP_RO$="" /* 受信バッファクリア
    PR "from IP=" rcvIP "rcv$=" rcv$
    IF snd$==rcv$ THEN
      PR "ok"
      BREAK
    END_IF
    PR "mismatch" snd$ rcv$
    ngcnt=ngcnt+1
  LOOP /* 不整合ならリトライ
  c=c+1
  TIME 100
LOOP
```

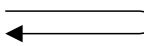
・ MPC2プログラム

```
SET_IP 192 168 0 18 255 255 255 0 192 168 0 248 /* 自己IP設定
SET_IP UDP PACKET 9000 /* UDP Port番号
UDP_RO$=""
DO
  WAIT UDP_RO$<>"" /* 受信待ち
  rcvIP=IPA(UDP_RO$)&&hFF /* 送信元IPアドレス
  rcv$=UDP_RO$ /* 受信文字列
  UDP_RO$="" /* 受信バッファクリア
  PR "from IP=" rcvIP "rcv$=" rcv$
  UDP_XO$=rcv$ /* 受信文字列をMasterに返信
LOOP
```

・ 実行結果

```
MPC1
snd$= 3500 10:12:20
from IP= 18 rcv$= 3500 10:12:20
ok
snd$= 3501 10:12:20
from IP= 18 rcv$= 3501 10:12:20
ok
snd$= 3502 10:12:20
from IP= 18 rcv$= 3502 10:12:20
ok
snd$= 3503 10:12:20
from IP= 18 rcv$= 3503 10:12:20
ok
snd$= 3504 10:12:21
from IP= 18 rcv$= 3504 10:12:21
ok
snd$= 3505 10:12:21
from IP= 18 rcv$= 3505 10:12:21
ok

MPC2
from IP= 17 rcv$= 3500 10:12:20
from IP= 17 rcv$= 3501 10:12:20
from IP= 17 rcv$= 3502 10:12:20
from IP= 17 rcv$= 3503 10:12:20
from IP= 17 rcv$= 3504 10:12:21
from IP= 17 rcv$= 3505 10:12:21
```



※社内LANに接続した環境で実行した結果、10万回で約40回のタイムアウトが発生しました。整合エラーは0でした。

[参考] MPC間のTCP、UDP通信について

TCP通信はリトライまで含めて信頼性の高い通信ということになっていますが以下の欠点があります。

- 1) 通信デバイス間でパワーオンオフが発生した場合、再びTCPコネクションの再確立に手間がかかる。
電源オフでなくてもデバイスのデバッグ中には同様のことが発生し、やはり回復に手間がかかる。
(コマンド操作では回復せず電源再投入が必要なこともある)
- 2) 通信経路でエラーが発生した場合にリトライロスが1秒程度必要となる。
このエラーは、1つのLANの中で接続されたデバイスが多いほど発生しやすくなる。

このためTCP/IPではクライアント/サーバーの役割が明確でスター状の接続の場合には比較的運用がしやすい。多くの場合サーバーはTCPのスレーブとなり、接続ルートごとにタスクを生成してクライアントサーバーシステムを構築します。また、サーバーはダウンしないように運用します。

MPCでは1つのポートに1つタスクのみとなるため、複数のマスターからの応答には複数のポートを用意します。また、サーバーとなるMPCは停止させないように運用する必要が生じ、稼働はかなり難しいものとなります。

従いまして、MPC間の相互通信にはUDPの方が安定したシステムを構築できます。もちろんUDPには、パケット喪失というリスクがともないますので、リトライ確認の機能を通信に埋め込む必要があります。