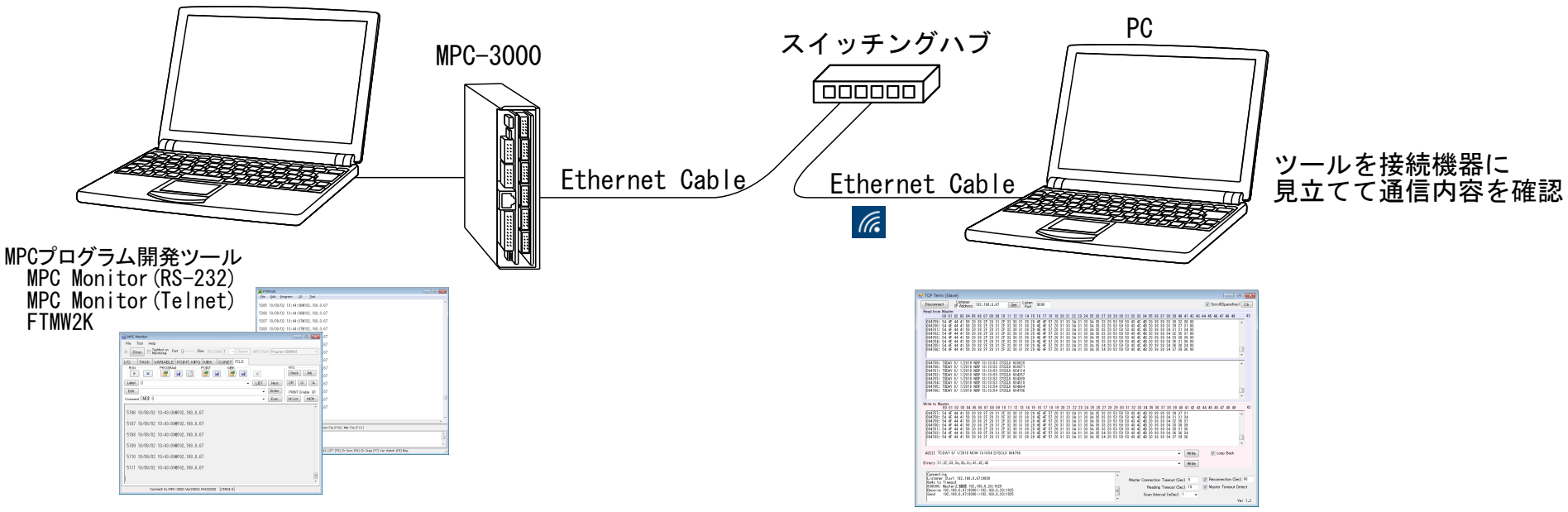
Ethernet開発支援ツール

ACCEL ti2k-180829 (LastUpdate Jul 31, 2019)

※改版により、実際のアプリケーションと画像が異なるものがあります。

■ 概要

- ターミナルツール

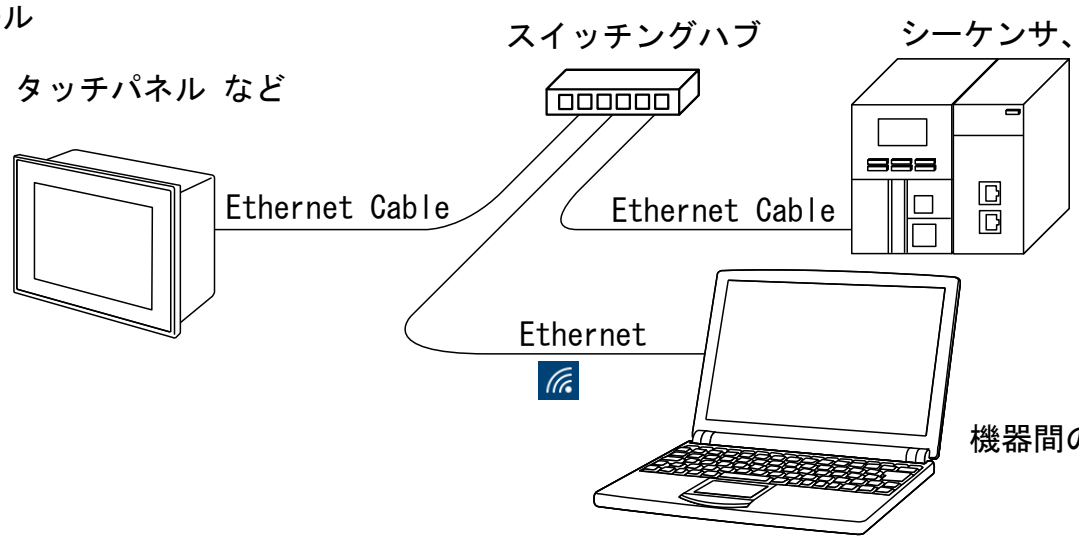


MPCの設定例

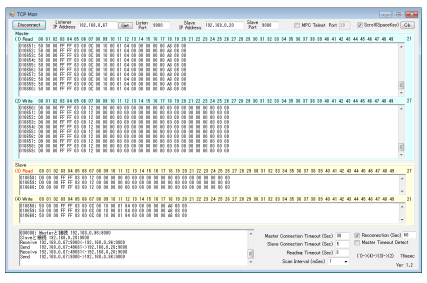
開発支援ツールの実行ファイル名

```
[Master] SET_IP TCP PACKET 9000 192 168 0 67  ⟸  TcpTermS.exe  TCPターミナル[Slave]
[Slave]  SET_IP TCP PACKET 9000               ⟹  TcpTermM.exe  TCPターミナル[Master]

[Master] SET_IP UDP PACKET 9000 192 168 0 67  ⟸  UdpTermS.exe  UDPターミナル[Slave]
[Slave]  SET_IP UDP PACKET 9000               ⟹  UdpTermM.exe  UDPターミナル[Master]
```

- モニターツール



機器間の通信内容をモニターします

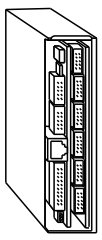


ツールの実行ファイル名

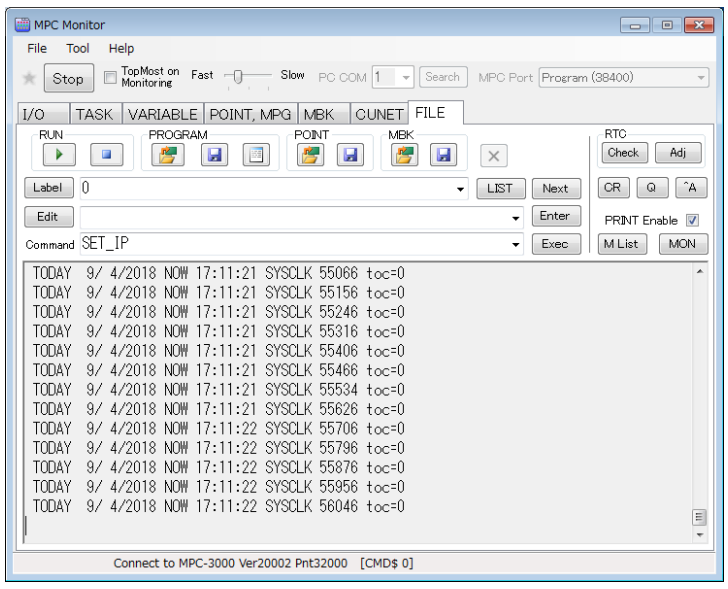
TcpMon.exe　TCPモニター
UdpMon.exe　UDPモニター

■ ターミナルツール(1) TcpTermS.exe
TCPでMPCがMaster、PCがSlave

[Master] MPC



```
#SET_IP                                  ← 設定内容を確認するコマンド
IPadrs = 192.168.0.20                    ← MPCのIP Addess
 MASK  = 255.255.255.0
Gateway= 192.168.0.248
OFF Delay ACK  baud=115200
MACadrs= 001EC08DAB6E

1 : /  TCP   Telnet          23      Slave  connected-> 192.168.0.53
2 : =  TCP   PACKET        9000    <TCP_XO$><TCP_RO$> Master 192.168.0.53 ref_time = 50
#
                          PCのPort    送信・受信コマンド        PCのIP Address
```



(MPC Monitor (RS-232))

■ MPCプログラム例

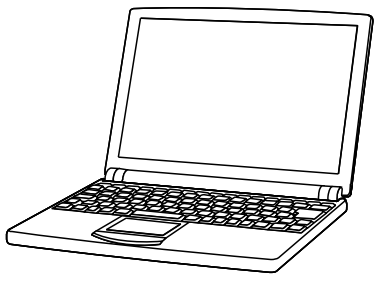
```
SET_IP TCP Telnet
SET_IP 192 168 0 20 255 255 255 0 192 168 0 248
SET_IP TCP PACKET 9000 192 168 0 53       /* Define as the Master

toc=0
SYSCLK=0
DO
  TCP_RO$=""
  TCP_XO$="TODAY "+DATE$(1)+" NOW "+TIME$(1)+" SYSCLK "+STR$(SYSCLK)+" toc="+STR$(toc)
  timer_=20
  WAIT (TCP_RO$<>"")|(timer_==0)
  IF timer_==0 THEN
    TCP_XO$="Timeout"
    PR "Timeout" toc
    toc=toc+1
  END_IF
  st$=TCP_RO$
  PR st$
  TIME 10
LOOP
```

■実行例

PC→MPCループバック文字列

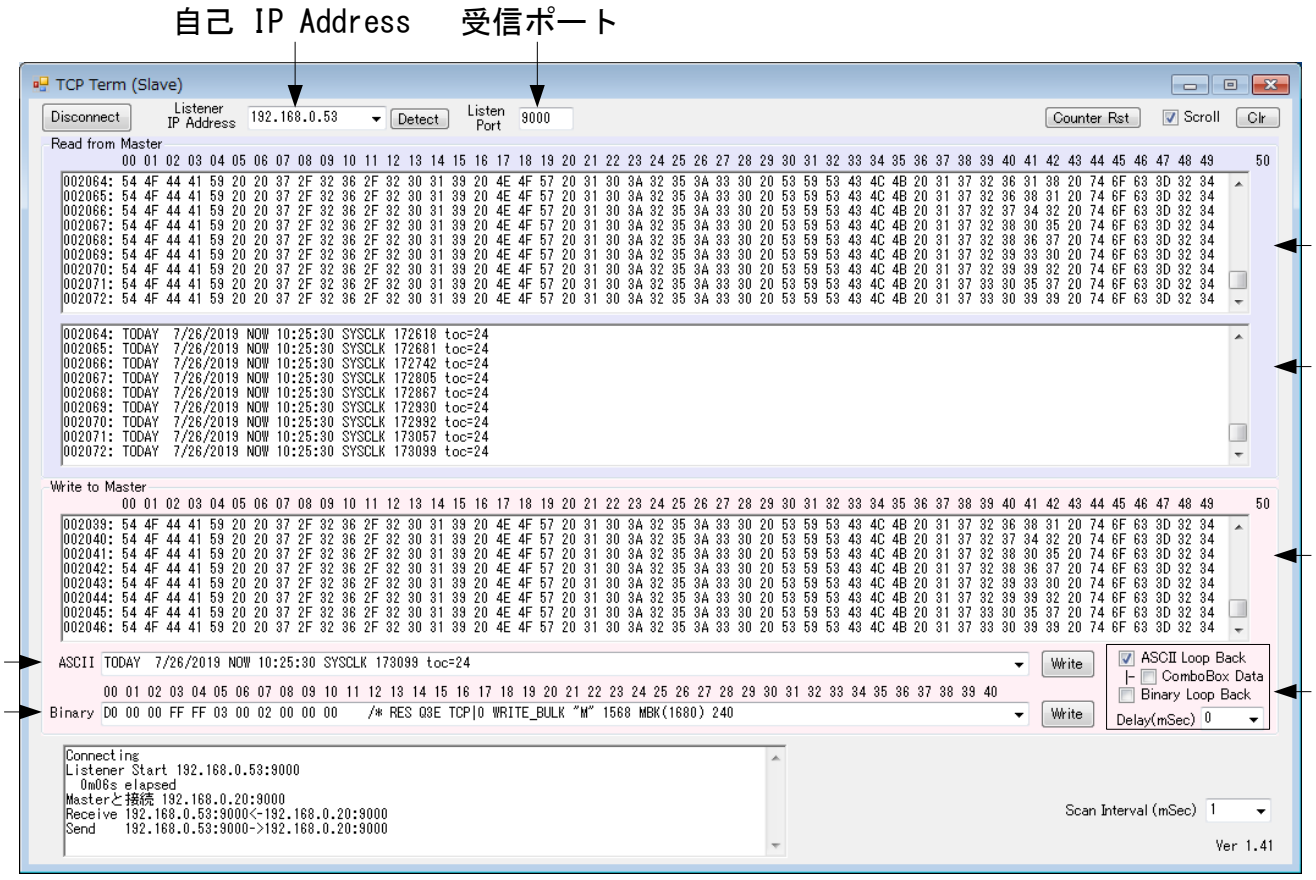
```
 TODAY  9/ 4/2018 NOW 17:12:40 SYSCLK 133866 toc=0
 TODAY  9/ 4/2018 NOW 17:12:40 SYSCLK 133956 toc=0
 TODAY  9/ 4/2018 NOW 17:12:40 SYSCLK 134016 toc=0
 TODAY  9/ 4/2018 NOW 17:12:40 SYSCLK 134090 toc=0
 TODAY  9/ 4/2018 NOW 17:12:40 SYSCLK 134176 toc=0
```

[Slave] PC



TcpTermS.exe
IP Address 192.168.0.53
Port 9000

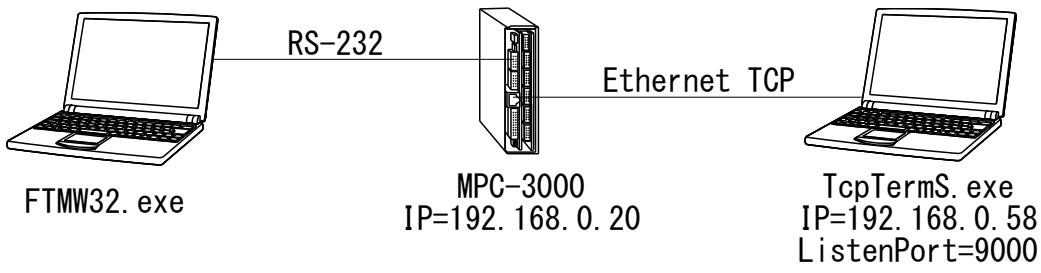
自己 IP Address　受信ポート



MPCから送信された文字列
Binary表示
ASCII表示
MPCへ送信した文字列
送信する文字列
ASCII
Binary
受信文字列をMPCへループバック
ComboBox Dataをチェックすると
左のComboBoxのTextを返信

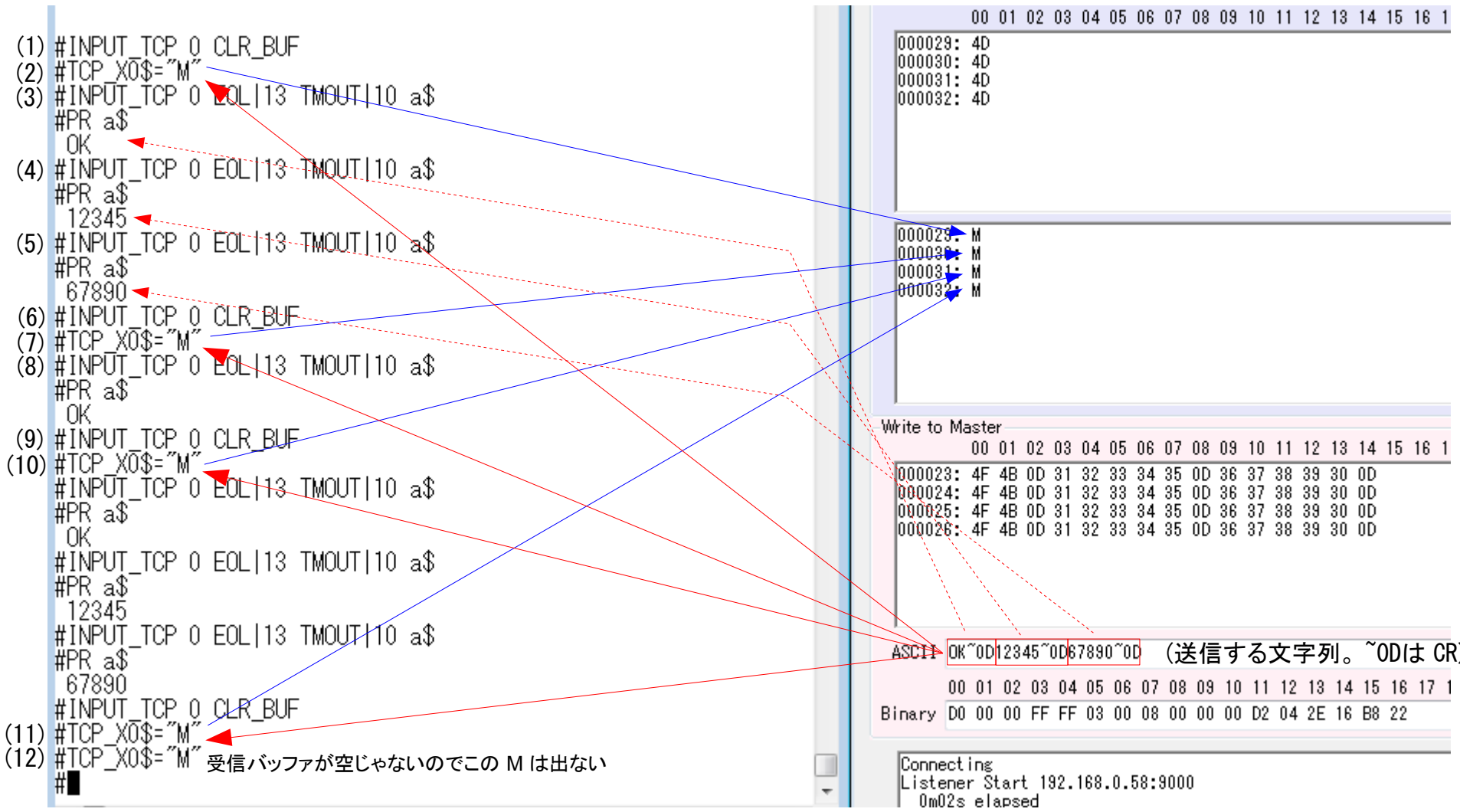
• TcpTermS.exe 使用例

TcpTermSを画像処理装置に見立て、MPC-3000のINPUT_TCPコマンドの動作確認をしています。
この装置は、 MPCからMコマンド受信 → MPCにOK送信 → 計測 → MPCに結果送信(CR区切りで複数ある) という動作をします。
OKから結果送信までインターロックは無くタイミングも不定です。また結果が複数のパケットになることもあります。
(確認:MPC-3000(SH2A) BL/I 2.01_23 2019/02/28)



MPC-3000の設定
SET_IP 192 168 0 20 255 255 255 0 192 168 0 248
SET_IP TCP PACKET 9000 PC 192 168 0 58

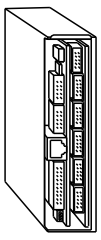
FTMW2K画面



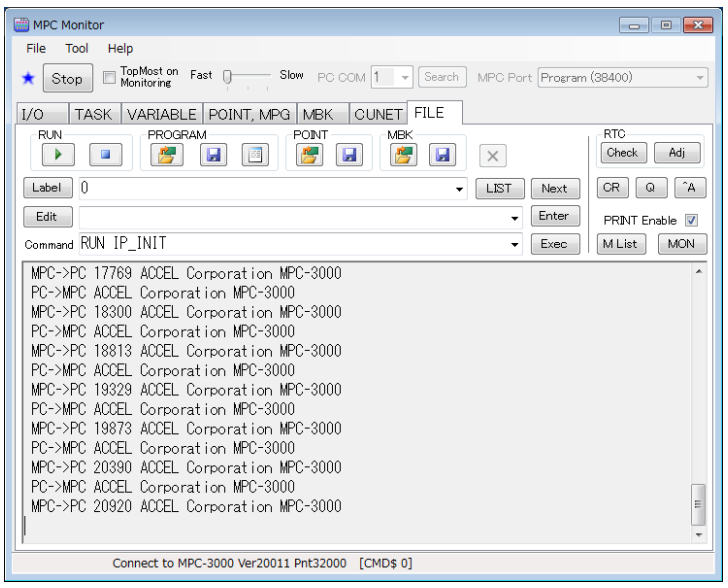
(1) 受信バッファクリア
(2) M送信。TcpTermSからMを受信後 OK<CR>12345<CR>567890<CR> と送信します。
(3) 既に受信バッファに文字列が入っているのでINPUT_TCPは抜けてa$にOKが入ります(<CR>は入りません)。
(4) 直ぐに抜けてa$に12345が入ります(<CR>は入りません)。
(5) 直ぐに抜けてa$に56789が入ります(<CR>は入りません)。
(6)(7)(8)は上と同様の手順
(9) まだ受信バッファに2文字列残っていますがクリアします。
(10) Mが送信されます。
(11)(12) バッファに残留物があると送信できません。

■ ターミナルツール(2) TcpTermM.exe
TCPでMPCがSlave、PCがMaster

[Slave] MPC



```
#SET_IP
IPadrs = 192.168.0.18
 MASK  = 255.255.255.0
Gateway= 192.168.0.248
OFF Delay ACK  baud=115200
MACadrs= 001EC08DAB6E

1 : /  TCP   Telnet          23      Slave
2 : =  TCP   PACKET        9000    <TCP_XO$><TCP_RO$> Slave  connected-> 192.168.0.53
#                        MPCのPort    送信・受信コマンド                  PCのIP Address
```



(MPC Monitor (RS-232))

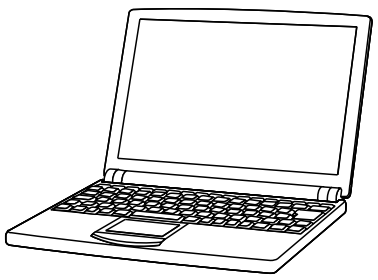
■ MPCプログラム例

```
SET_IP 192 168 0 18 255 255 255 0 192 168 0 248
SET_IP TCP Telnet
SET_IP TCP PACKET 9000                /* Define as the Slave

TCP_XO$=""
TCP_RO$=""
DO
  WAIT TCP_RO$<>""
  st$=TCP_RO$
  PR "PC->MPC" st$
  TCP_RO$=""
  st$=STR$(SYSCLK)+" "+st$
  PR "MPC->PC" st$
  TCP_XO$=st$                         /* Reply to the Master
  WAIT TCP_XO$==""
LOOP
```

■実行例

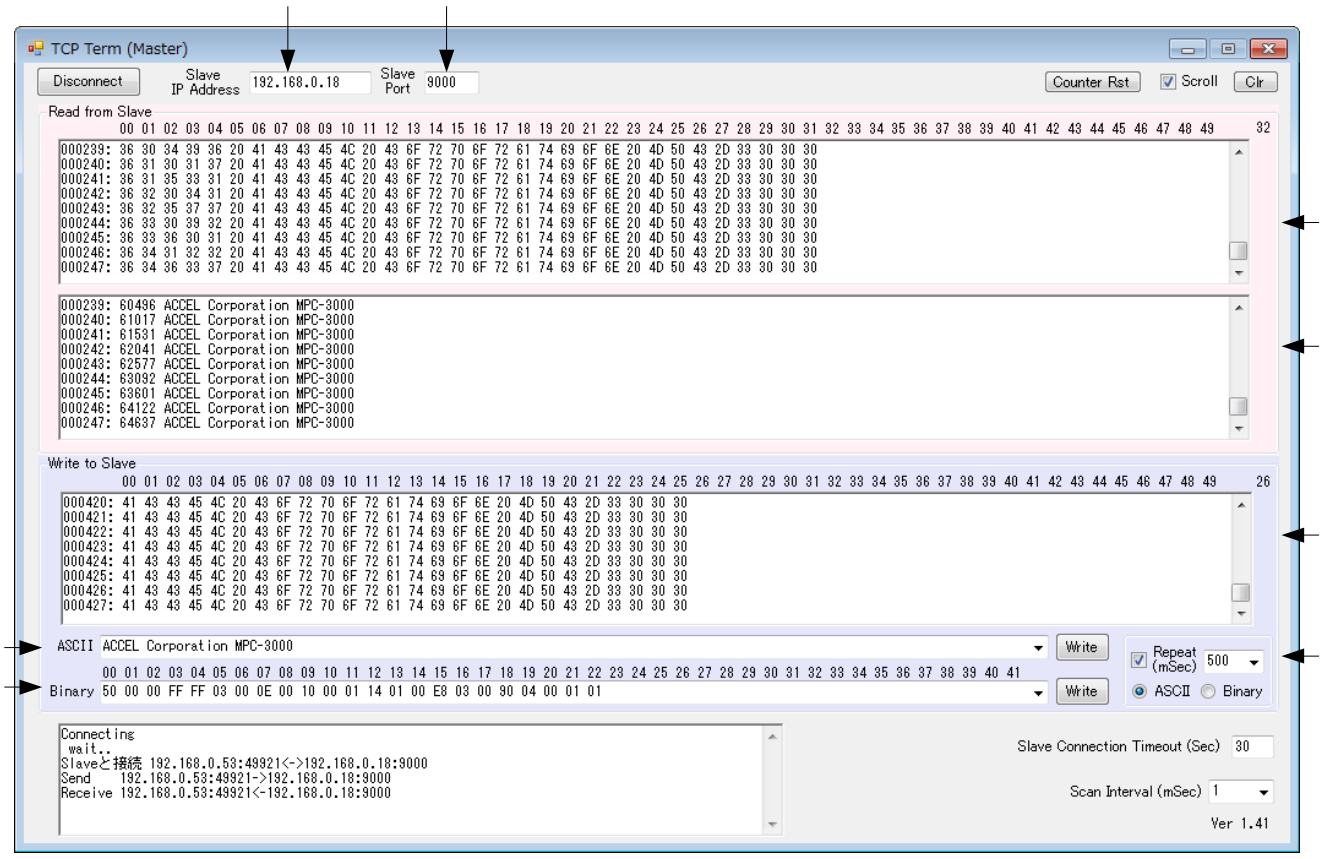
```
PC->MPC ACCEL Corporation MPC-3000
MPC->PC 5605667 ACCEL Corporation MPC-3000
PC->MPC ACCEL Corporation MPC-3000
MPC->PC 5606183 ACCEL Corporation MPC-3000
```

[Master] PC



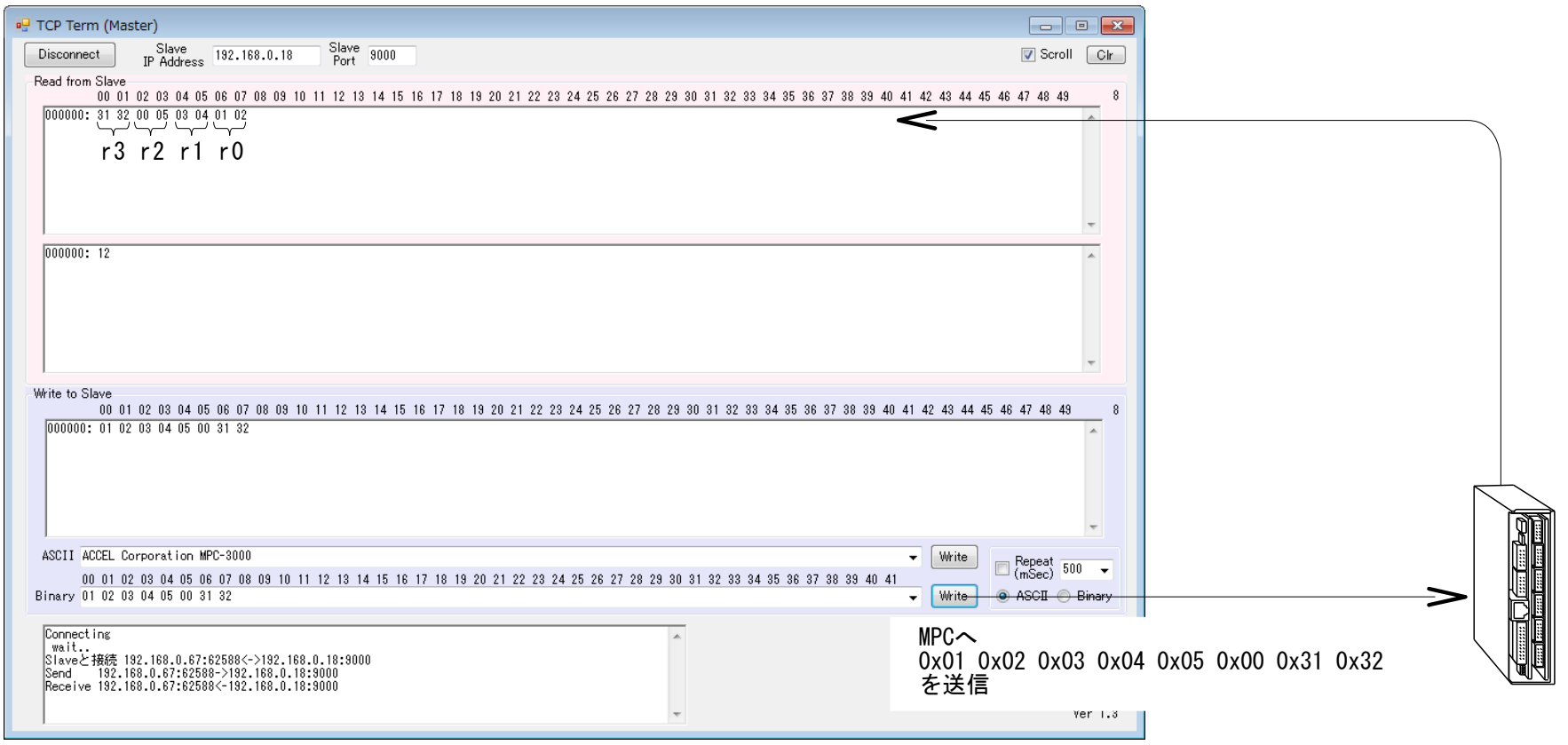
TcpTermM.exe
IP Address 自動
Port 自動

MPCの IP Address　MPCのポート



MPCから送信された文字列
Binary表示
ASCII表示
MPCへ送信した文字列
繰り返し送信
送信する文字列
ASCII
Binary

• TcpTermM.exe 使用例1
PCからMPCへバイナリ送信 → MPCはバイナリ受信処理 → PCへPACKET送信



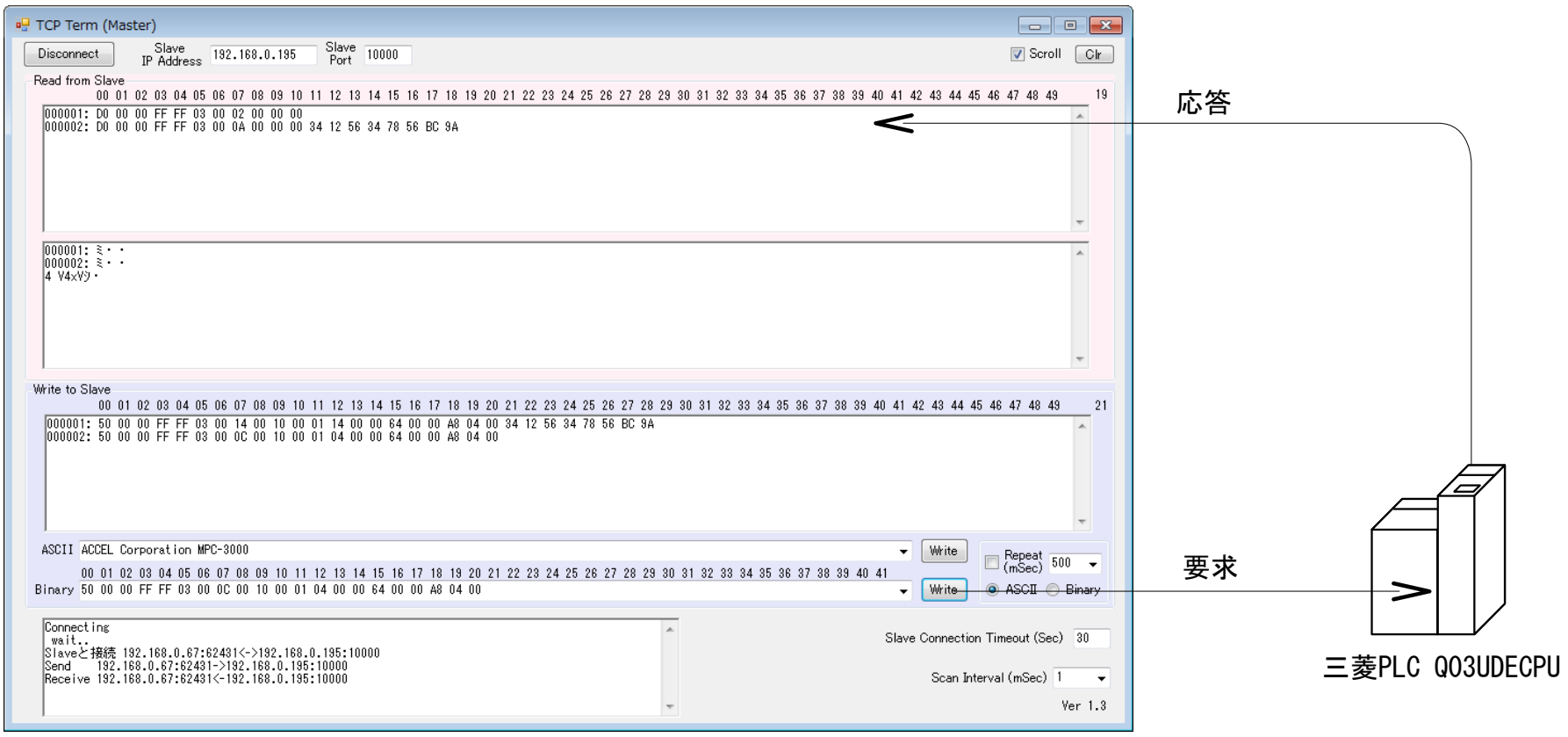
■ MPCプログラム

```
/* PCから文字列を受信したらバイナリデータとして変数に入れ、PCに送り返す

SET_IP 192 168 0 18 255 255 255 0 192 168 0 248
SET_IP TCP PACKET 9000              /* Master(PC)  -> Slave MPC:9000

DO
  TCP_RO$=""
  WAIT TCP_RO$<>""                  /* WAIT文:文字列先頭が0x00はNG
  ptr_=TCP_RO$                      /* prt_は受信文字列のポインター
  r0=PTR(2,0)                       /* 0番目(先頭)から2バイト(ワード)取得
  PR "r0=" HEX$(r0) r0
  r1=PTR(4,0)                       /* 0番目(先頭)から4バイト(ロング)取得
  PR "r1=" HEX$(r1) r1
  r2=PTR(-2,4)                      /* 4番目から2バイト(ワード)取得、上位下位スワップ
  PR "r2=" HEX$(r2) r2
  r3=PTR(2,6)                       /* 6番目から2バイト(ワード)取得
  PR "r3=" HEX$(r3) r3
  PACKET TCP_XO$ r3 r2 r1 r0  /* 送信するのは下位2バイト
LOOP
```

■ 実行結果

```
 r0= 00000102 258        ←HEX DEC
 r1= 01020304 16909060
 r2= 00000005 5
 r3= 00003132 12594
```

• TcpTermM.exe 使用例2
PC→PLC MCプロトコル



■ PLCのD100~D103に 0x1234, 0x3456 0x5678 0x9ABC を書き込む

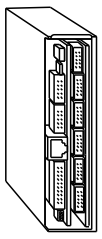
要求: 50 00 00 FF FF 03 00 14 00 10 00 01 14 00 00 64 00 00 A8 04 00 34 12 56 34 78 56 BC 9A
応答: D0 00 00 FF FF 03 00 02 00 00 00

■ PLCのD100~D103を読み出す

要求: 50 00 00 FF FF 03 00 0C 00 10 00 01 04 00 00 64 00 00 A8 04 00
応答: D0 00 00 FF FF 03 00 0A 00 00 00 34 12 56 34 78 56 BC 9A

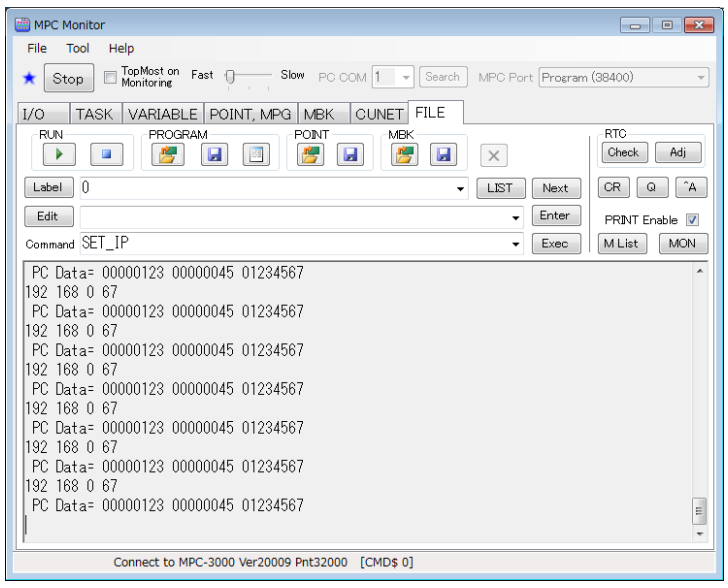
■ ターミナルツール(3) UdpTermS.exe
UDPでMPCがMaster、PCがSlave

[Master] MPC



```
#SET_IP
IPadrs = 192.168.0.20
 MASK  = 255.255.255.0
Gateway= 192.168.0.248
OFF Delay ACK
MACadrs= 001EC08DAB6E

1 : /  UDP    PACKET      61440    <UDP_X0$><UDP_R0$>
```

PCのPort　送受信に用いる予約文字列変数



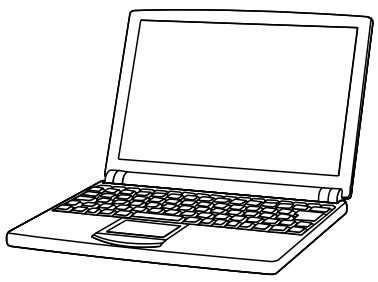
■ MPCプログラム例

```
SET_IP 192 168 0 20 255 255 255 0 192 168 0 248 /* 自己IP設定
SET_IP UDP PACKET 61440                 /* UDP Port番号

IP_CONV 192 168 0 53 PcIp               /* 変数 PcIp にPCのIP 192.168.0.53 を入れる
SET_DEST 0ppdIp UDP_X0$                 /* 宛先のIPと送信文字列変数 UDP_X0$ を紐付け。
DO

  DO
    UDP_R0$=""
    SET_DEST PcIp UDP_X0$               /* 宛先設定
    PACKET UDP_X0$ &h0123 &h4567        /* 送信
    timer_=10
    WAIT (IPC(UDP_R0$)!=0)|(timer_==0)  /* 応答待ち
    IF timer_==0 THEN
      PR "TIME OUT PC"
    ELSE
      BREAK
    END_IF
  LOOP
  IP_CONV IPA(UDP_R0$)                  /* 送信元IP表示
  ptr_=UDP_R0$                          /* PC から [0]0x01 [1]0x23 [2]0x45 [3]0x67 と来る
  a1=PTR(2,0)                           /* [0]番目から Word 読み込み
  a2=PTR(1,2)                           /* [2]番目から Byte 読み込み
  a3=PTR(4,0)                           /* [0]番目から Long 読み込み
  PRINT "PC Data=" HEX$(a1) HEX$(a2) HEX$(a3)

LOOP
```

■実行例

```
192 168 0 53
 PC Data= 00000123 00000045 01234567
192 168 0 53
 PC Data= 00000123 00000045 01234567
```

[Slave] PC



UdpTermS.exe
IP Address 192.168.0.53
Port 61440

自己 IP Address　受信ポート

MPCから送信された文字列

Binary表示

ASCII表示

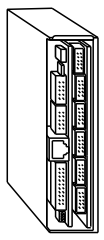
MPCへ送信した文字列

送信する文字列
ASCII
Binary

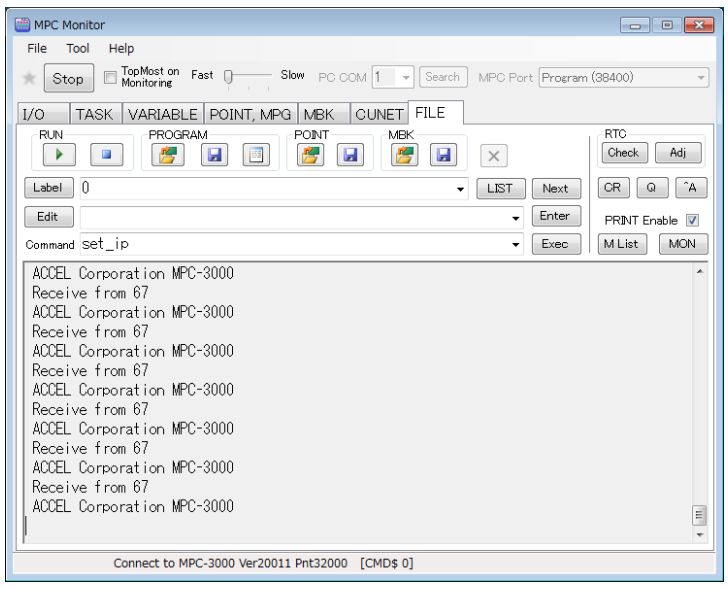
受信文字列をMPCへループバック

■ ターミナルツール(4) UdpTermM.exe
UDPでMPCがSlave、PCがMaster

[Slave] MPC



```
#SET_IP
IPadrs = 192.168.0.20
 MASK  = 255.255.255.0
Gateway= 192.168.0.248
OFF Delay ACK  baud=115200
MACadrs= 001EC08DAB6E

1 : /  UDP    PACKET        61440      <UDP_XO$><UDP_RO$>
                                  MPCのPort    送受信に用いる予約文字列変数
```



(MPC Monitor (RS-232))

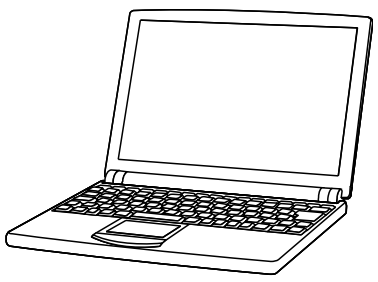
■ MPCプログラム例

```
/* 受信だけなら※行は不要

 SET_IP 192 168 0 20 255 255 255 0 192 168 0 248 /* 自己IP設定
 SET_IP UDP PACKET 61440                  /* UDP Port番号
 IP_CONV 192 168 0 53 PcIp                /* 変数 PcIp にPCのIP 192.168.0.53 を入れる ※
 SET_DEST PcIp UDP_XO$                    /* 宛先設定 ※

 DO
   UDP_RO$=""
   WAIT IPC(UDP_RO$)!=0                   /* 受信待ち
   ipa=IPA(UDP_RO$)&&hFF                  /* 送信元IPアドレスの最下位
   PR "Receive from" ipa
   len=IPC(UDP_RO$)                       /* 受信文字列長
   ptr_=UDP_RO$
   rcv$=PTR$(len)
   PR rcv$
   UDP_XO$=STR$(SYSCLK)+" "+rcv$          /*SYSCLK値と受信文字列をPCに送信 ※
 LOOP
```

■実行例

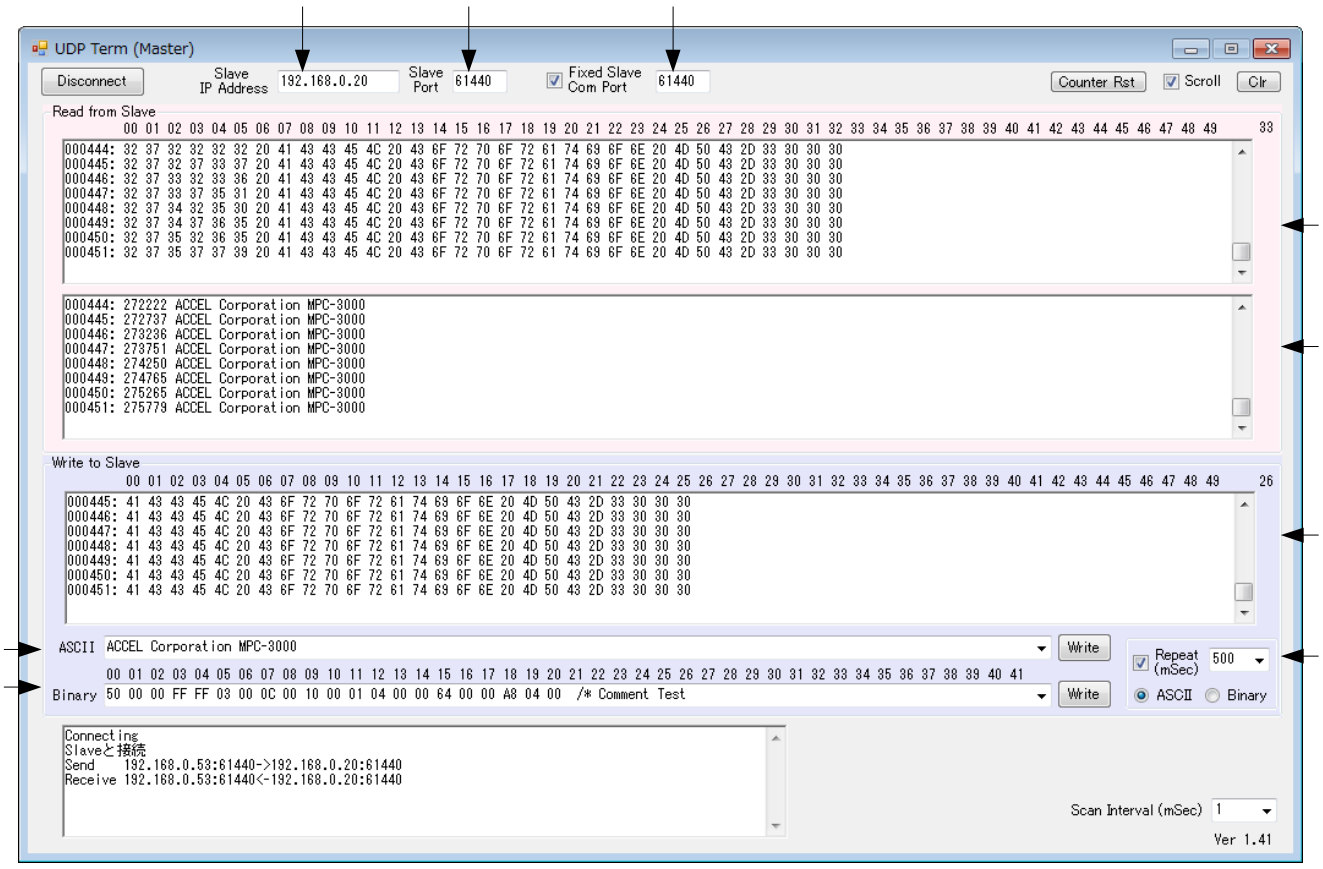
```
Receive from 53              ← PCのIP Address 最下位
ACCEL Corporation MPC-3000   ← 受信した文字列
```

[Master] PC



UdpTermM.exe
IP Address 自動
Port 自動(固定)

MPCの IP Address　MPCのポート MPCと通信するポートを固定

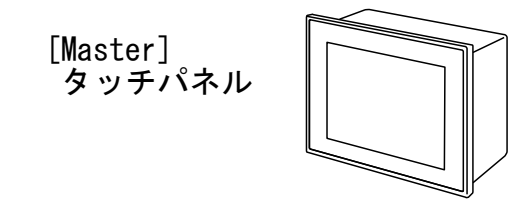


MPCから送信された文字列
← Binary表示
← ASCII表示
← MPCへ送信した文字列
← 繰り返し送信

送信する文字列
ASCII →
Binary →

■ モニターツール(1) TcpMon.exe
Master・Slave TCP通信のモニタ

Proface GP4301TW
本体 イーサネット設定
IPアドレス 192.168.0.96、サブネットマスク 255.255.255.0、ポート 8000
デフォルトゲートウェイ 192.168.0.248
周辺機器 ＞ 接続機器設定
接続機器
三菱電機(株) Q/QnAシリーズイーサネット TCP
通信設定
ポート番号 固定/自動、タイムアウト 5sec、リトライ 0、送信ウェイト 0
機器設定
接続機器名 PLC1、IPアドレス 192.168.0.67、ポート番号 9000、データコード バイナリ



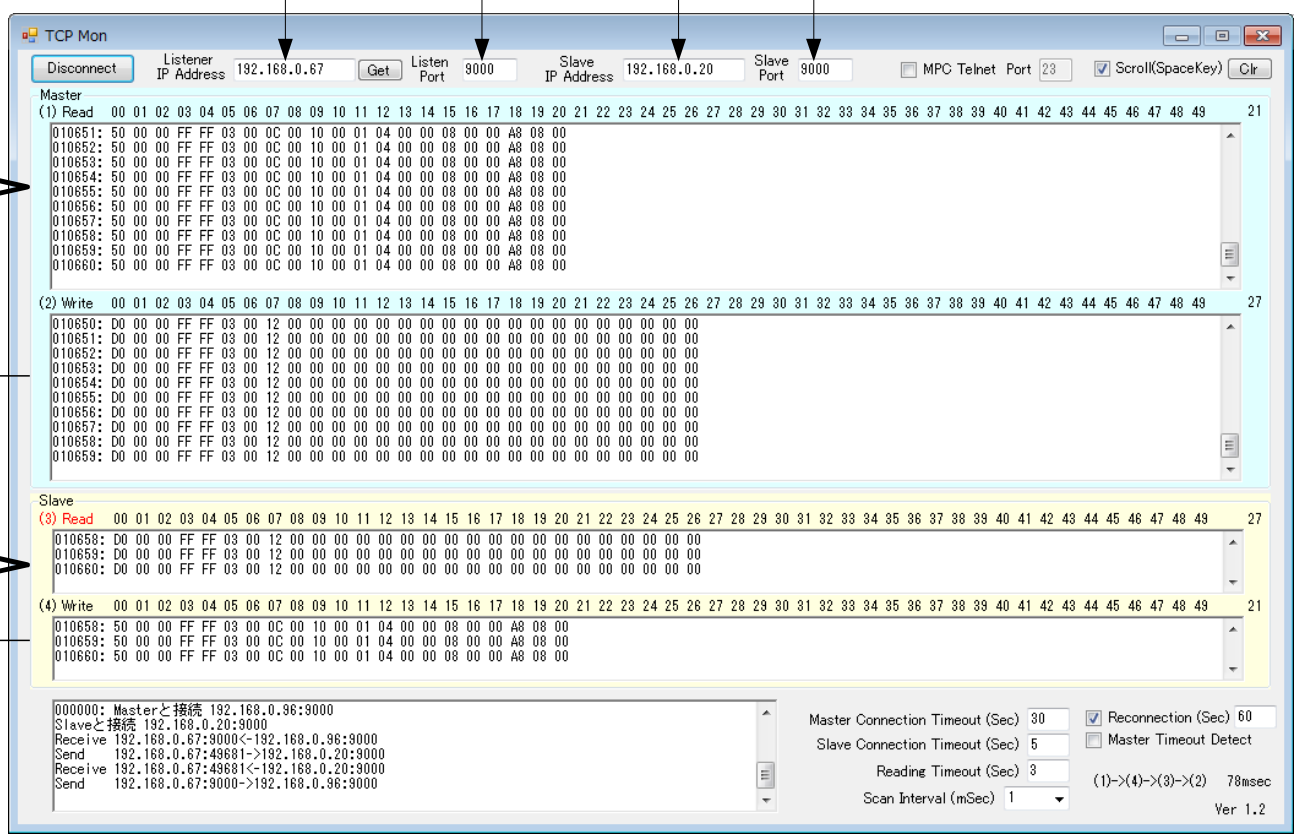
[Master]
タッチパネル

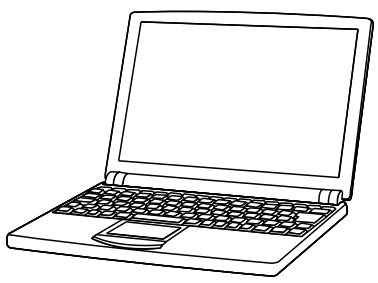
(4)
Write

(1)
Read

自己 IP Address, 受信ポート

MPCのIP Address, ポート

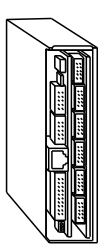




TcpMon.exe
IP Address 192.168.0.67
Port 9000

(3)
Read

(2)
Write



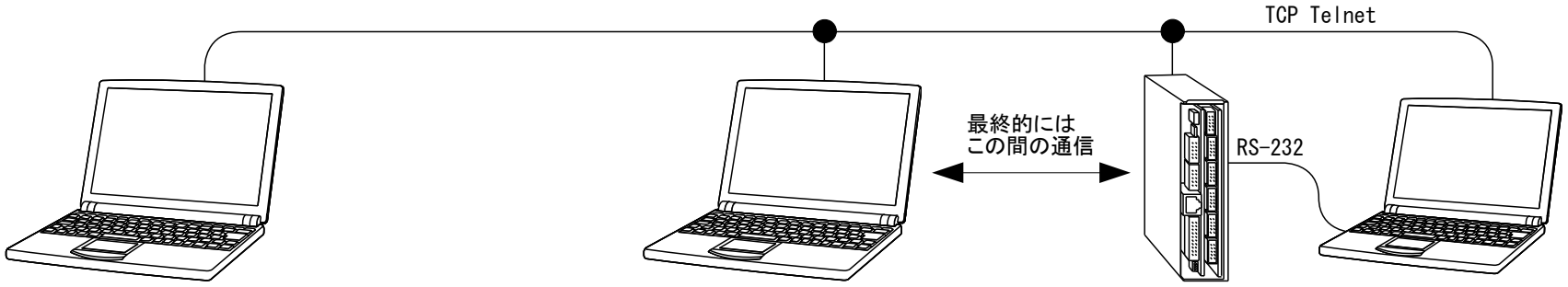
[Slave] MPC

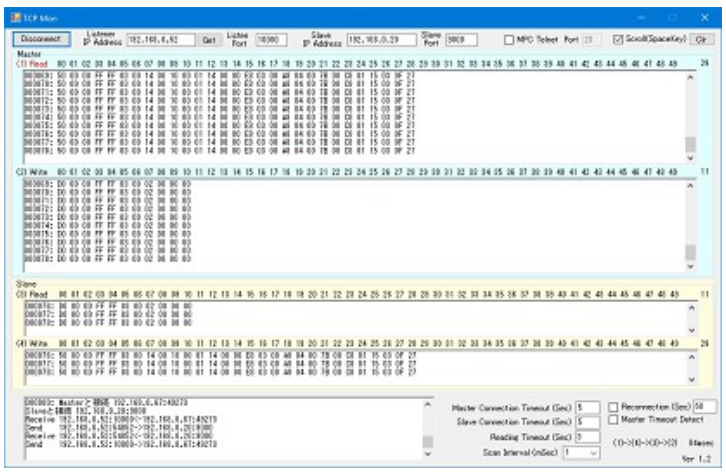
#SET_IP
IPadrs = 192.168.0.20
MASK = 255.255.255.0
Gateway= 192.168.0.248
OFF Delay ACK
MACadrs= 001EC08DAB6E

1 : = TCP MC_PROTOCOL 9000 Slave connected-> 192.168.0.67
MPCのPort PCのIP Address

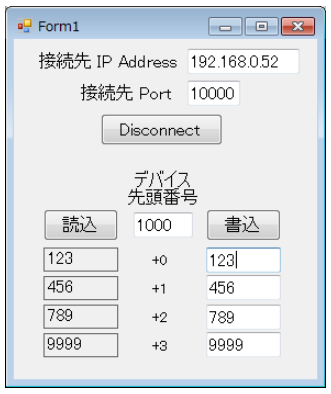
■設定コマンド
SET_IP TCP MC_PROTOCOL 9000

• 使用例1
通信内容を見ながらアプリケーションを開発できます。

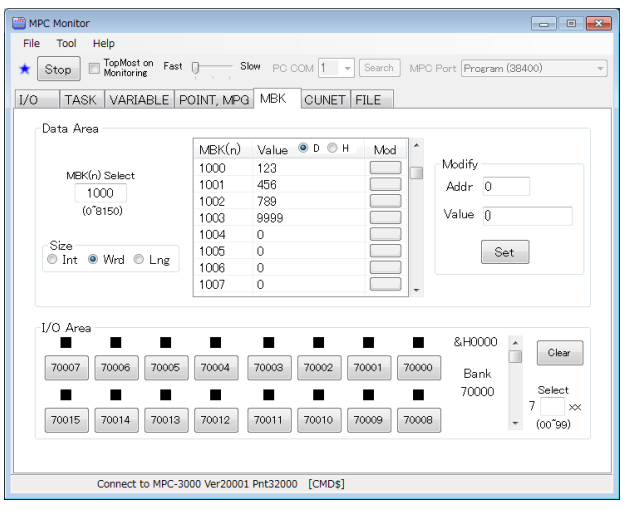




TcpMon.exe
TCPアプリとMPCの通信内容が見えます



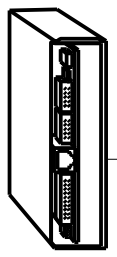
[Master]
ユーザー開発
TCPアプリケーション



[Slave]
MPCプログラム開発

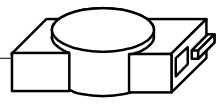
• 使用例2
MPC-3000とADAM-6052間のModbus/TCP通信をモニタしてみます。

[Master]
MPC-3000

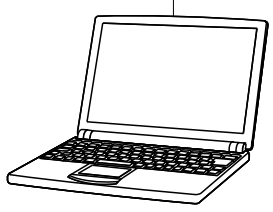


SET_IP 192 168 0 20 255 255 255 0 192 168 0 248
SET_IP TCP Modbus ADAM 192 168 0 67
(SET_IP TCP Modbus ADAM 192 168 0 196 とすればADAMと直接通信)

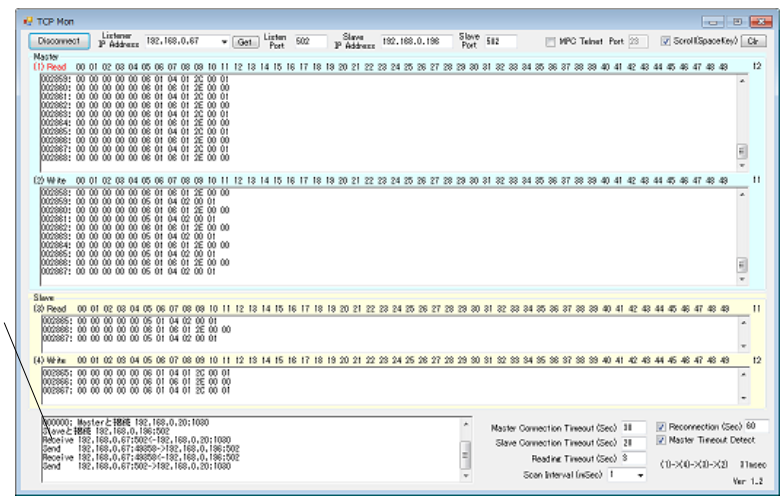
[Slave]
Advantech I/Oモジュール
ADAM-6052



IP Address 192.168.0.196
Port Number 502



TcpMon.exe

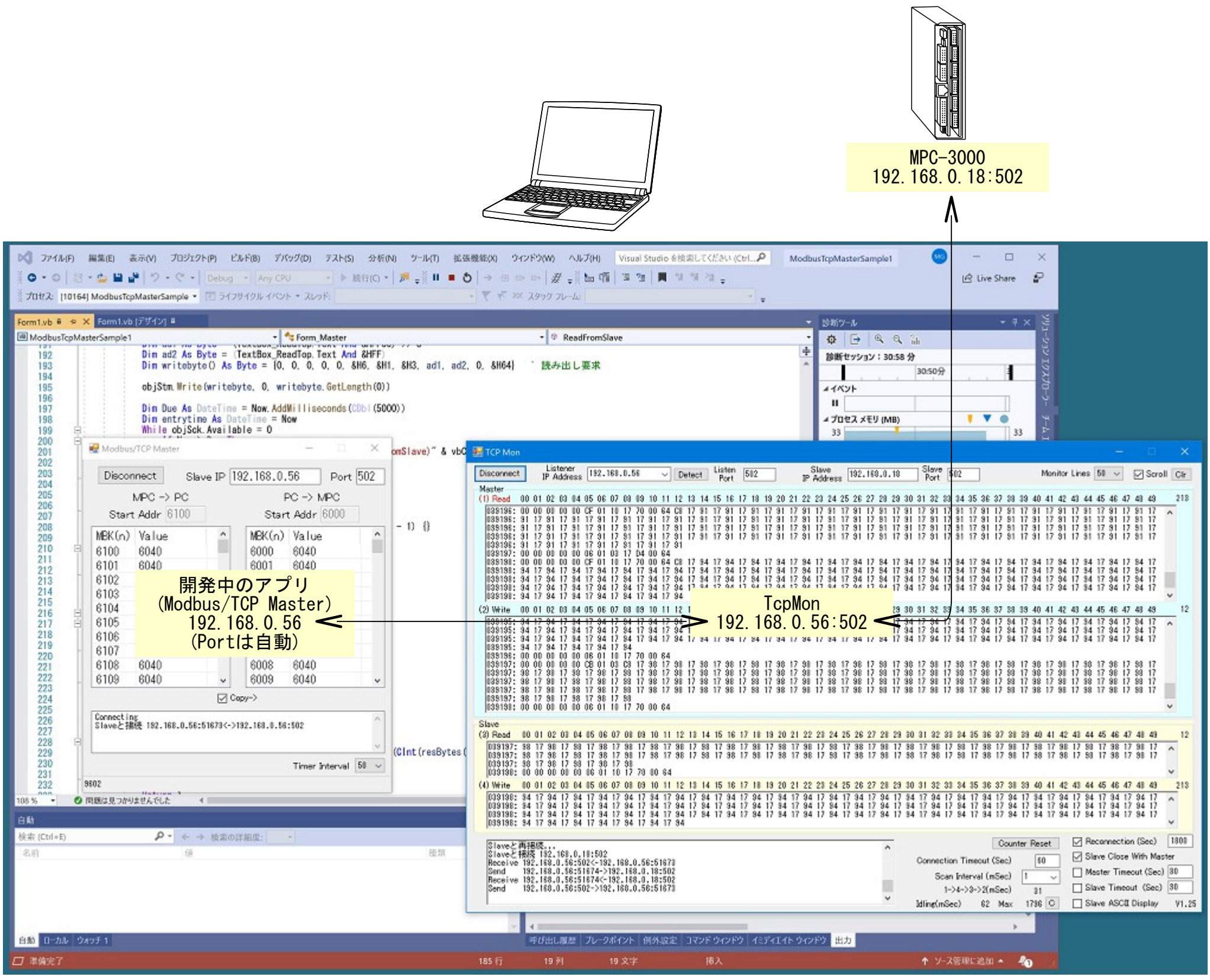


Listener IP 192.168.0.67 Linten Port 502
Slave IP 192.168.0.196 Slave Port 502

接続・通信時のエンドポイント

000000: Masterと接続 192.168.0.20:1030
Slaveと接続 192.168.0.196:502
Receive 192.168.0.67:502<-192.168.0.20:1030
Send 192.168.0.67:49358->192.168.0.196:502
Receive 192.168.0.67:49358<-192.168.0.196:502
Send 192.168.0.67:502->192.168.0.20:1030

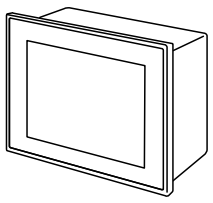
- 使用例3
  アプリ開発中のPCでもモニタができます。



■ モニターツール(2) UdpMon.exe
Master・Slave UDP通信のモニタ

Proface GP4301TW
　本体 イーサネット設定
　　IPアドレス 192.168.0.96、サブネットマスク 255.255.255.0、ポート 8000
　　デフォルトゲートウェイ 192.168.0.248
　周辺機器 ＞ 接続機器設定
　　接続機器
　　　パナソニックデバイスSUNX(株) MEWTOCOL7-COMイーサネット UDP
　　通信設定
　　　ポート番号 固定/自動、タイムアウト 5sec、リトライ 2、送信ウェイト 0
　　機器設定
　　　接続機器名 PLC1、IPアドレス 192.168.0.67、ポート番号 8000

[Master]
タッチパネル



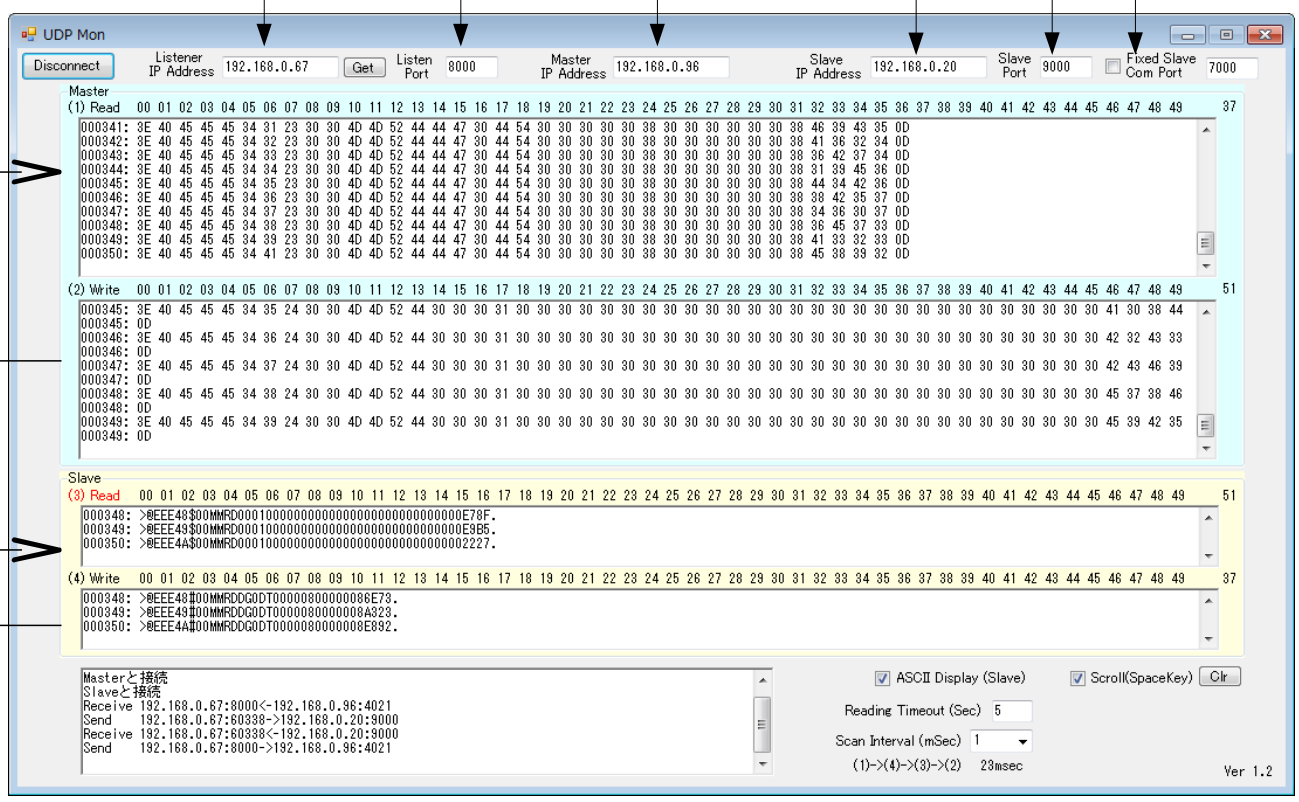
(4)
Write

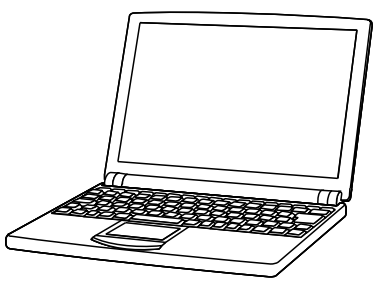
(1)
Read

自己 IP Address, 通信ポート

GPのIP Address

MPCのIP Address, ポート, 通信ポート固定



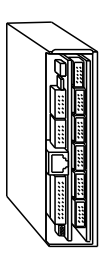


UdpMon.exe
IP Address 192.168.0.67
Port 8000

(3)
Read

(2)
Write

[Slave] MPC



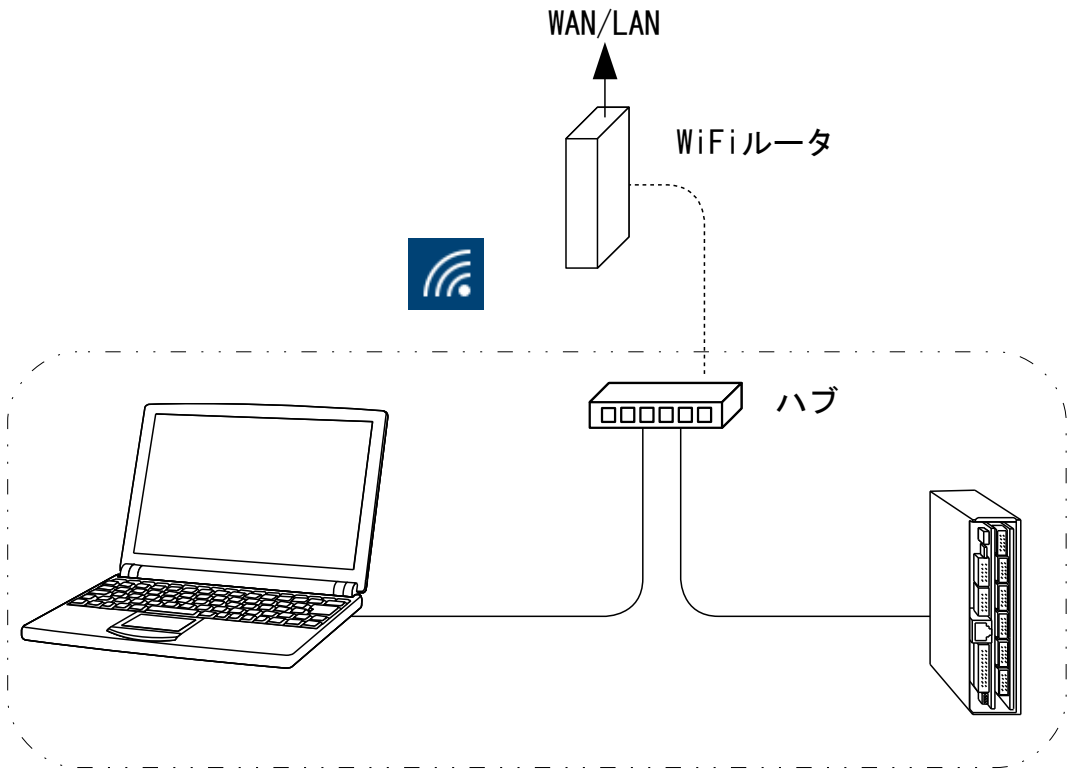
#SET_IP
IPadrs = 192.168.0.20
 MASK = 255.255.255.0
Gateway= 192.168.0.248
OFF Delay ACK
MACadrs= 001EC08DAB6E

1 : = UDP MEWTOCOL7 9000 Slave

MPCのPort

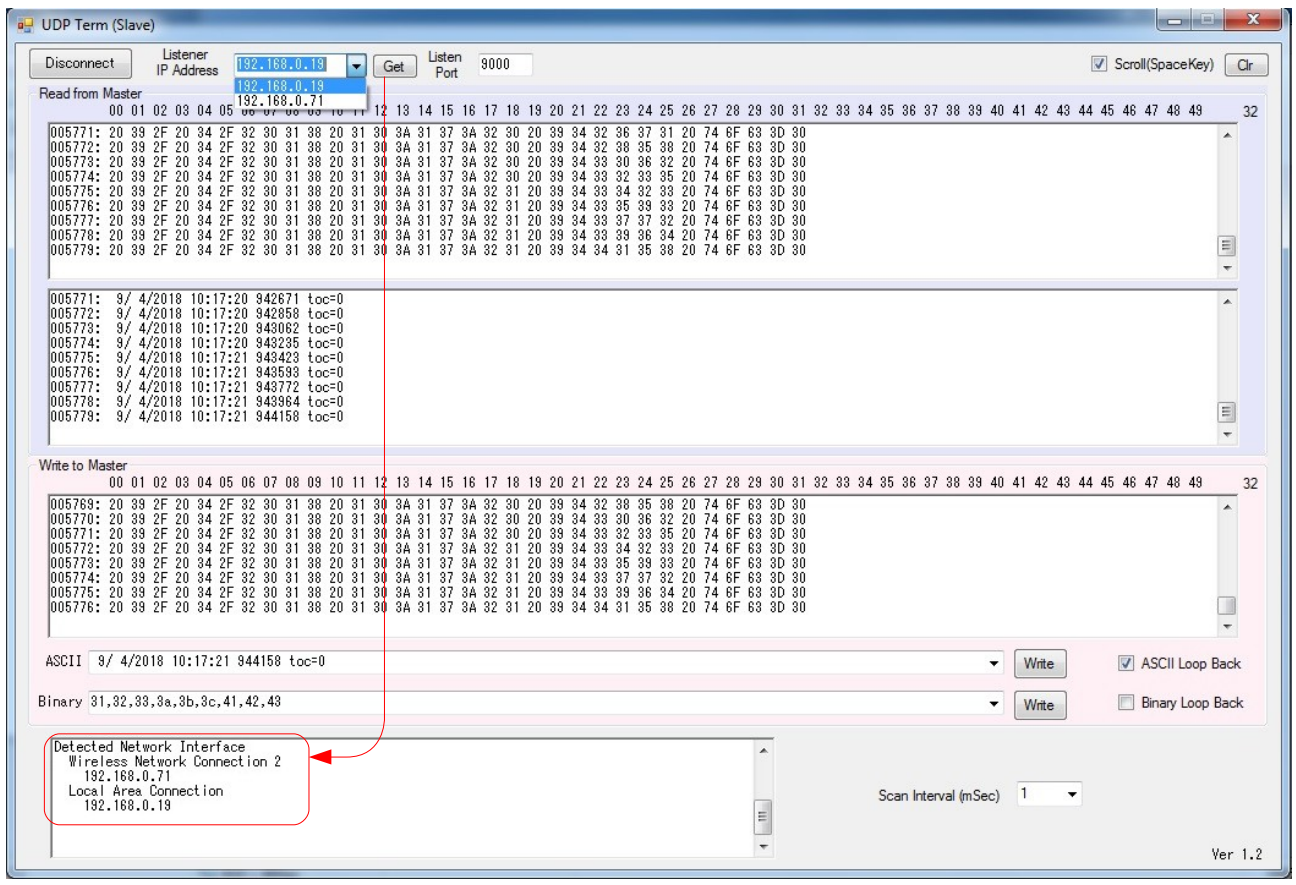
■設定コマンド
SET_IP UDP MEWTOCOL7 9000

■ 自己IP Addressの設定例
IP Addressが複数ある場合



Wireless(WiFi)とEtherケーブルが
接続されているPC

Getボタンを押すとローカルIP Addressが2つ見つかります。
下のTextboxにインターフェースの種類が表示されるので確認して設定してください。
このPCはLocal Areaの方が具合が良いので19を設定。
通信は、MPC直結かLANと分離された状態(二点鎖線内)にするとより安定。



詳細はipconfigで