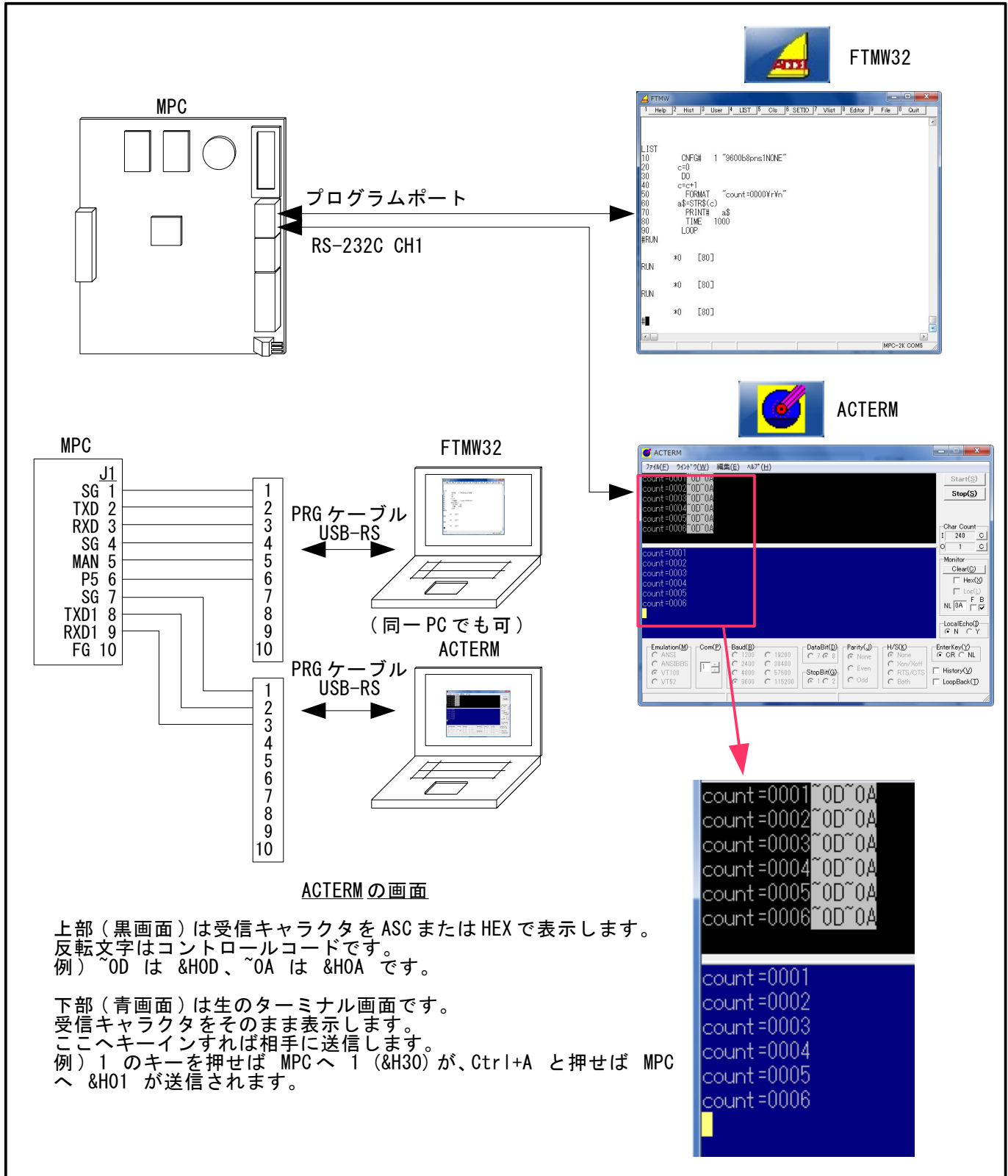


テーマ	RS-232C、RS-485 通信内容の確認方法
使用機器	MPC、FTMW32、ACTERM (汎用ターミナルソフト)、LINEMON (ラインモニタ)

### ■ターミナルソフトで MPC の通信確認

- ACTERM を接続機器に見立てて通信のシミュレーションをします。
- 使用ポート、CH 番号と結線は適宜変更してください。
- MPC-2000 シリーズの受信は INPUT、送信は PRINT だけです。PUT、GET 等の 1 キャラ専用コマンドはありません。
- ACTERM は FTMW フルセットアップで ACCEL フォルダにインストールされています。
- この例は MPC へ接続する機器の代替として用いていますが、MPC の代わりに ACTERM と接続機器をつなぎ通信内容を確認することもできます。



### ◆ダイレクトコマンドによる受信確認例

```
#CNFG# 1 "9600b8pns1NONE"
#PR LOF(1) /* 受信バッファカウンタ
0 /* 空

/* ここで ACTERM から &H01 (Ctrl+A) を送信

#PR LOF(1)
1 /* 受信バッファカウンタ=1 キャラ
#INPUT# 1 CHR_C|1 a$ /* 1文字入力。
#PR LOF(1) /* 入力後のバッファカウンタ
0
#PRX ASC(a$) /* 入力内容 HEX 表示
00000001
#INPUT# 1 CHR_C|1 a$ /* 受信するまで待ち状態。ここで ACTERM から &H02 (Ctrl+B) を送信
#PRX ASC(a$)
00000002
```

### ◆受信プログラム例1：1キャラクタ受信

```
10 CNFG# 1 "9600b8pns1NONE"
20 DO
30 INPUT# 1 CHR_C|1 a$
40 PRX ASC(a$)
50 LOOP
#RUN

00000001 /* ACTERM から &H01 (Ctrl+A) を送信
00000002 /* ACTERM から &H02 (Ctrl+B) を送信
00000031 /* ACTERM から "1" を送信
00000032 /* ACTERM から "2" を送信
0000000D /* ACTERM から &HOD (Enter) を送信
```

### ◆受信プログラム例2：1キャラクタ受信、タイムアウト付

```
10 CNFG# 1 "9600b8pns1NONE"
20 DO
30 INPUT# 1 CHR_C|1 TMOUT|2 a$
40 IF rse==1 THEN
50 PRINT "TIME OUT"
60 ELSE
70 PRX ASC(a$)
80 END_IF
90 LOOP
#RUN

TIME OUT /* 2秒以内に受信しないとタイムアウト
00000001 /* ACTERM から &H01 (Ctrl+A) を送信
00000002 /* ACTERM から &H02 (Ctrl+B) を送信
00000031 /* ACTERM から "1" を送信
00000032 /* ACTERM から "2" を送信
0000000A /* ACTERM から &HOA (Ctrl+J) を送信
TIME OUT
```

### ◆受信プログラム例3：&H03(ETX)コードが来るまで1キャラ受信、タイムアウト付

```
10 CNFG# 1 "9600b8pns1NONE"
20 DO
30 b$=""
40 DO
50 INPUT# 1 CHR_C|1 TMOUT|2 a$
60 IF rse==1 THEN
70 PRINT "TIME OUT"
80 BREAK
90 ELSE
100 IF ASC(a$)==&H00000003 THEN
110 PRINT b$
120 BREAK
130 END_IF
140 b$=b$a$
```

```

150      END_IF
160      LOOP
170      LOOP
#RUN

```

TIME OUT

```

123      /* ACTERM から 123~03 を送信
ABC      /* ACTERM から ABC~03 を送信

```

◆ダイレクトコマンドによる送信確認例

```
#CNFG# 1 "9600b8pns1NONE"
```

```

#PRINT# 1 CHR$(1)          /* &H01 を送信 ACTERM は ~01 と表示
#PRINT# 1 CHR$(&H0D)       /* &H0D(CR)を送信 ACTERM は ~0D と表示
#PRINT# 1 CHR$(&H31)       /* 1 を送信
#PRINT# 1 STR_LEN|1 CHR$(0) /* &H00(NULL)を送信 ACTERM は ~00 と表示

#a$="1234567"+CHR$(&H03)   /* 文字列 + コントロールコード
#PRINT# 1 a$              /* 文字列送信 ACTERM は 1234567~03 と表示

#a$="1234567¥r"          /* 文字列 + CR
#PRINT# 1 a$             /* 文字列送信 ACTERM は 1234567~0D

#a$="1234567¥n"          /* 文字列 + LF
#PRINT# 1 a$             /* 1234567~0A

#a$="1234567¥r¥n"        /* 文字列 + CRLF
#PRINT# 1 a$             /* 1234567~0D~0A

```

◆送信プログラム例 1：変数を文字列に書式変換して送信

```

10      CNFG# 1 "9600b8pns1NONE"
20      c=0
30      DO
40      c=c+1
50      FORMAT "count=0000¥r¥n"
60      a$=STR$(c)
70      PRINT# a$
80      TIME 1000
90      LOOP
#RUN

```

・ ACTERM の表示

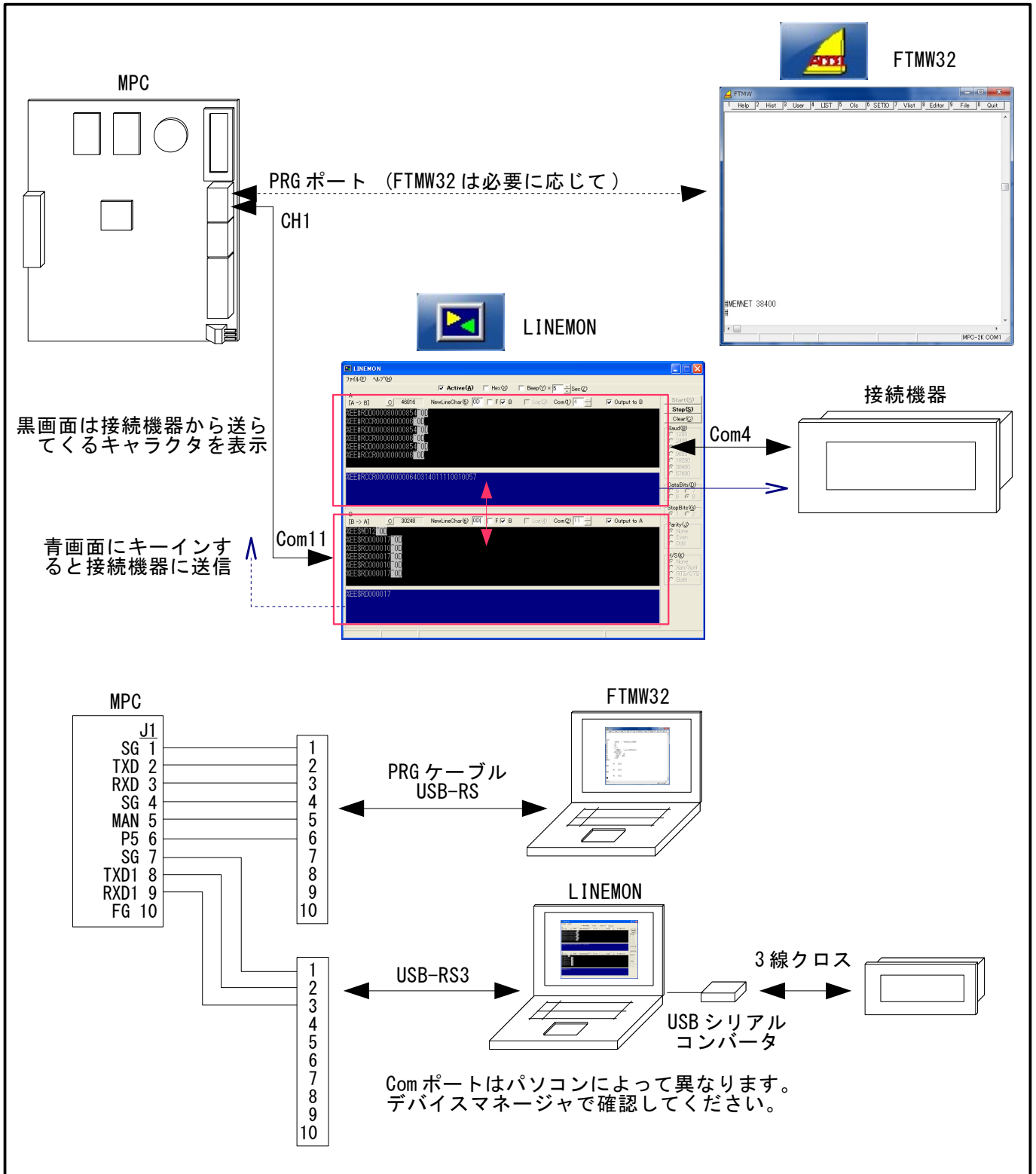
```

count=0001~0D~0A
count=0002~0D~0A
count=0003~0D~0A
count=0004~0D~0A
count=0005~0D~0A
count=0006~0D~0A

```

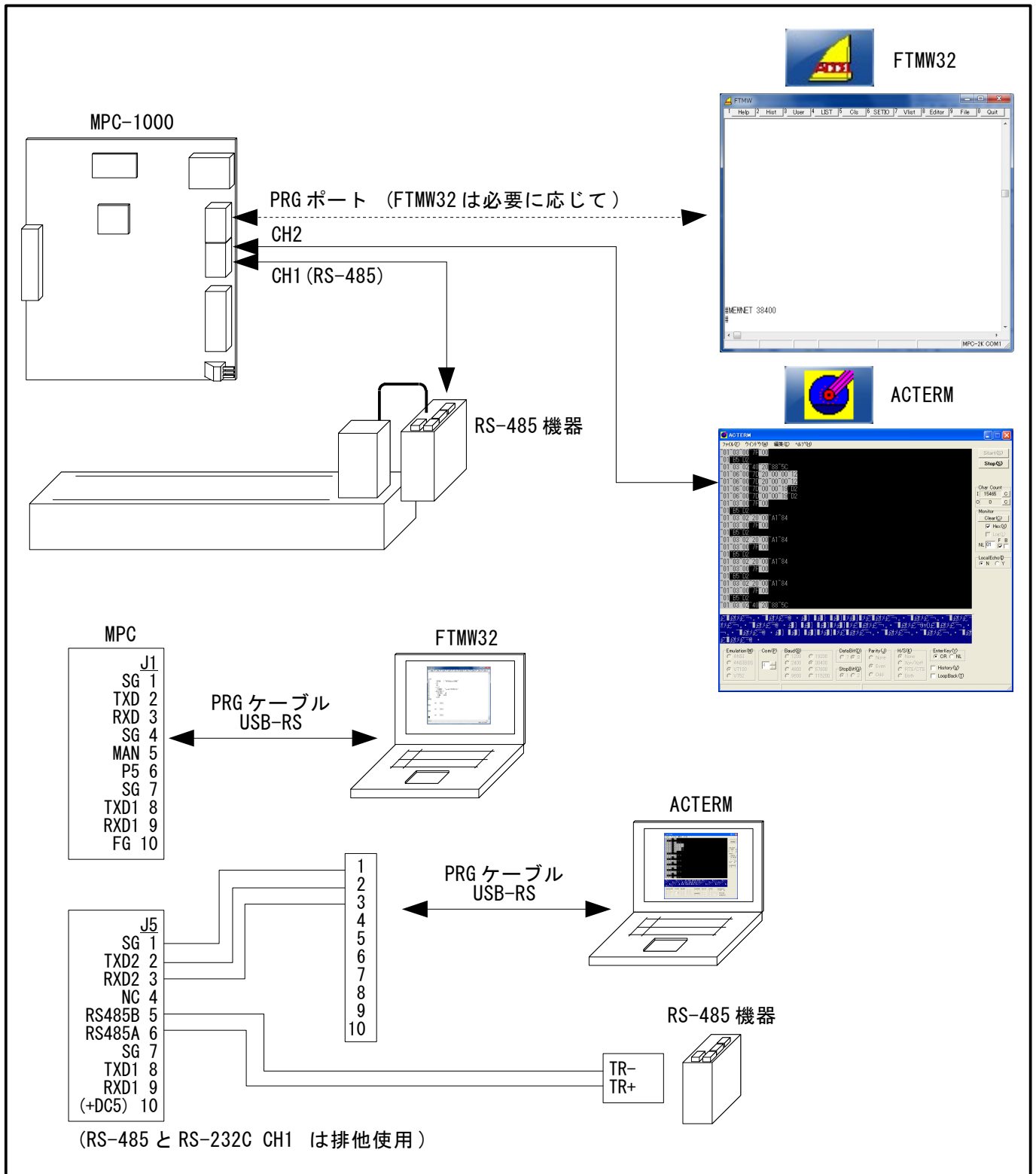
## ■ラインモニタで通信内容を見る

- LINEMON を MPC と接続機器の間に入れて通信内容をモニタします。
- LINEMON は上記 ACTERM を 2 連結したようなアプリです。
- 使用ポート、CH 番号と結線は適宜変更してください。
- LINEMON を入れると通信に遅延が発生します。直結状態とタイミングが変わりますのでハンドシェークのないプロトコルなどは注意してください。
- LINEMON は FTMW フルセットアップで ACCEL フォルダにインストールされています。



## ■空きポートを使ってモニター

- パソコンに RS-485 機器を接続するには RS-485/232C コンバータが必要ですが、この例では MPC の空き RS-232C ポートに ACTERM を接続して通信内容をモニターします。
- 空き RS-232C ポートがない場合は MRS-MCOM を使用します。
- 接続機器が RS-232C でも ok です。



◆プログラム例

MPCのプログラムに ACTERM へ表示するためのコマンドを仕込みます。

- ・ 送信内容の表示  
 PRINT# 1 STR\_LEN|DATA\_CNT+2 SEND\$ /\* RS-485 機器へ送信  
 PRINT# 2 STR\_LEN|DATA\_CNT+2 SEND\$ /\* <表示用>同じデータを ACTERM にも送信
- ・ 受信内容の表示  
 INPUT# 1 CHR\_C|1 RES\$ /\* RS-485 機器から受信  
 PRINT# 2 STR\_LEN|1 RES\$ /\* <表示用>RS-485 機器から受信したキャラクタを ACTERM へ送信

◆画面サンプル

オリエンタルモーター(株) ARD-KD の Modbus 通信のモニター例

The screenshot shows the ACTERM software interface. The main window displays a log of communication data. Annotations point to specific lines in the log:

- 送信クエリ (-JOG コマンド)**: "スレプ=01, 書込=06, アドレス=007D, データ=2000, CRC 下位, CRC 上位" これは「ドライブ入力指令」の「-JOG」ビットを立てる
- レスポンス**: 正常なら送信クエリと同じ
- 送信クエリ (-JOG コマンド)**: "スレプ=01, 書込=06, アドレス=007D, データ=2000, CRC 下位, CRC 上位" これは「ドライブ入力指令」の「-JOG」ビットを立てる
- レスポンス**: 正常なら送信クエリと同じ
- 送信クエリ (状態確認) ※1**: "スレプ=01, 読出=03, アドレス=007F, レジスタ数=0001, CRC 下位, CRC 上位" これは「ドライブ出力指令」を読む
- レスポンス**: "スレプ=01, 読出=03, データバイト数=02, データ=2000, CRC 下位, CRC 上位" これは「ドライブ出力指令」の「MOVE」ビットが立っている=移動中
- 送信クエリ (状態確認) 同上**
- レスポンス**: "スレプ=01, 読出=03, データバイト数=02, データ=4020, CRC 下位, CRC 上位" これは「ドライブ出力指令」の「END」と「READY」ビットが立っている=ドライブがREADY状態

The interface also shows a menu bar (File, Edit, Help), a main text area with hex and ASCII data, and a control panel on the right with buttons for Start, Stop, Char Count, Monitor, and LocalEcho. At the bottom, there are settings for Emulation, Com, Baud, DataBit, Parity, H/S, and EnterKey.

※1) ACTERM の改行コード(NL)に&H01 を設定しているなので、クエリーに&H01 が含まれると改行表示されますが、実際には1文字列です。

--End Of File--