Panasonic®

プログラマブルコントローラ
MEWNET FP SERIES
MEWNET-Hリンクソフト(システムコール編)
プログラミングマニュアル

MEWNET-Hリンクソフト（システムコール編）プログラミングマニュアル
FAF−167① '95・2月

松下電工

安全に関するご注意

ケガや事故防止のため、以下のことを必ずお守りください。

据付、運転、保守、点検の前に、必ずこのマニュアルをお読みいただき、正しくご使用下さい。
機器の知識、安全の情報、その他注意事項のすべてを習熟してからご使用下さい。

このマニュアルでは、安全注意事項のレベルを「警告」と「注意」に区分しています。

⚠警告

取扱いを誤った場合に、使用者が死亡または重傷を負う危険の状態を生じることが想定される場合

●本製品の故障や外部要因による異常が発生しても、システム全体が安全側に働くように本製品の外部で安全対策を行ってください。

●燃焼性ガスの雰囲気では使用しないでください。
爆発の原因となります。

●本製品を火中に投棄しないでください。
電池や電子部品などが破裂する原因となります。

⚠注意

取扱いを誤った場合に、使用者が傷害を負うかまたは物的損害のみが発生する危険の状態が生じることが想定される場合

●異常発熱や発煙を防止するため、本製品の保証特性・性能の数値に対し余裕をもたせて使用してください。

●分解、改造はしないでください。
異常発熱や発煙の原因となります。

●通電中は端子に触れないでください。
感電のおそれがあります。

●非常停止、インターロック回路は外部で構成してください。

●電線やコネクタは確実に接続してください。
接続不十分な場合は、異常発熱や発煙の原因となります。

●製品内部に液体、可燃物、金属などの異物を入れないでください。
異常発熱や発煙の原因となります。

●電源を入れた状態では施工（接続、取り外しなど）しないでください。
感電のおそれがあります。

著作権および商標に関する記述

●Windows および WindowsNT は米国 Microsoft Corporation の米国およびその他の国における登録商標です。

●その他の会社および製品名は、各社の商標または登録商標です。

●商品改良のため、仕様、外観およびマニュアルの内容を予告なく変更することがありますので、ご了承ください。

はじめに

このたびは、MEWNET-Hリンクソフトをお買い求めいただき、誠にありがとうございます。
本製品には、MS-DOS/IBM-DOS上で弊社リンクボードを使用するための拡張デバイスドライバが含まれています。本製品を使用することにより、MS-DOS/IBM-DOSのユーザプログラムから弊社MEWNET-Hリンクボードの各種機能が使用できます。

このマニュアルは、MEWNET-Hリンクソフトの組み込み方法、拡張デバイスドライバを使用してユーザプログラムを作成する方法、機能コマンドの詳細説明を掲載しています。
ご使用になるまえに、必ずマニュアルをお読みいただき、内容をよくご理解いただいたうえで、正しくご活用いただきますようお願いいたします。

なお、MEWMET-Hリンクボードの取り扱い方法については、『MEWNET-Hリンクボード 導入マニュアル』をご覧ください。
BASIC言語でユーザプログラムを作成する方法については、『MEWNET-Hリンクソフト プログラミングマニュアル（BASIC編）』をご覧ください。

●お願い

このマニュアルの内容に関しては万全を期していますが、ご不審な点や誤りなどお気付きの点がございましたら、お手数ですが弊社までご連絡ください。

目　次

1章 ユーザープログラムの作成方法

2章 機能コマンド概要

3章　機能コマンドリファレンス

4章 ＭＥＷＮＥＴ－Ｈのプロトコル

1章 ユーザープログラムの作成方法

1 － 1 MEWNET-Hリンクソフトの使用方法

1-1-1 システムディスクの使用方法

MEWNET-Hリンクソフト（MS-DOS版）のシステムディスクには、MS-DOSデバイスドライバファイルとMS-DOS版BASICの機械語プログラムが収録されています。このマニュアルは、システムディスクに含まれるMS-DOSデバイスドライバを使用してユーザープログラムを作成する方法について説明します。MS-DOS版BASIC機械語プログラムの使用方法については、『MEWNET-Hリンクソフト（BASIC版）プログラミングマニュアル』をお読みください。

適応機種	システムディスク収録ファイル		ご注文品番
	MS-DOSデバイスドライバ	MS-DOS版BASIC機械語プログラム	
NEC PC98シリーズ用	MHPCMS.SYS	MHPCBS.SYS	AFP666008
IBM PC/AT互換機用	MHIBMS.SYS	MHIBBS.SYS	AFP666408
富士通 FMRシリーズ用	MHFMMS.SYS	MHFMBS.SYS	AFP666208

MS-DOSデバイスドライバのインストールは次の手順で行います。

①デバイスドライバファイルをコピーする

```
A>CD ¥
A>COPY B:MH**MS.SYS
```

②デバイスドライバファイルをCONFIG.SYSに登録する

```
A>CD ¥
A>COPY CON >> CONFIG.SYS
device=MH**MS.SYS
^Z
```

注　意

・上記「MH**MS.SYS」には、ご使用になるデバイスドライバ名を記述してください。
・MS-DOSの起動ディスクドライブは、必ずしもドライブAとは限りません（とくにハードディスクが起動ドライブの場合は注意してください）。起動ドライブがA以外の場合は、上記「A>」を「B>」「C>」「D>」‥‥に読み替えてください。

BASICプログラムでは、MH**BS.SYS（機械語プログラム）をメモリにロードし、CALL命令で機械語プログラムに実行を移します。詳細については、『MEWNET-Hリンクソフト（BASIC版）プログラミングマニュアル』をお読みください。

1-1-2 パラメータスイッチ

デバイスドライバをCONFIG.SYSに登録する際に、以下のパラメータが使用できます。

■パラメータ設定書式例

```
device=MHPCMS.SYS /0/I60/B1
                  ␣
                  スペース
```

パラメータ	指定	内容
(1)リンクソフト組み込み完了時のメッセージ表示	/0	メッセージ表示なし。 (ただしパラメータ設定エラー時の表示は行われます。)
	/1	メッセージ表示あり。
	無指定	メッセージ表示ありとなります。
(2)使用するソフトウェア割り込み(int40h以降)のNo.	/Ixx	xxにユーザーに解放されているNo.を指定します。 PC98シリーズ用・・・・・int40h以降 PC/ATシリーズ用・・・・int60h以降 FMRシリーズ用・・・・・・int50h以降 ｘｘ省略時は無指定時と同じになります。
	無指定	パラメータ無指定時は次のNo.が指定されます。 PC98シリーズ用・・・・・int40h PC/ATシリーズ用・・・・int60h FMRシリーズ用・・・・・・int50h
(3)MEWNET-P用ユーザープログラム*のサポート	/P	MEWNET-P用ユーザープログラム使用可能。 この時686バイト分だけ使用するメモリサイズが増加します。
	/H	MEWNET-P用ユーザープログラム使用不可。
	無指定	MEWNET-P用ユーザープログラム使用不可となります。
(4)登録送受信要求タイプコマンド**同時使用可能数	/Kxx	xxに10進数で0～16の範囲で指定します。 xx省略時は使用可能数＝0になります。 1登録あたり96バイト分だけ使用するメモリサイズが増加します。 使用可能数＝0を指定すると登録送受信要求タイプコマンドでのＲ／Ｗはできなくなります。
	無指定	使用可能数＝4となります。
(5)リンクボード使用可能数	/Bx	xに1～4の範囲で指定します。 x省略時はエラーになります。 ボード1枚あたり416バイト分だけ使用するメモリサイズが増加します。 パラメータに/Pを指定した時は、ユーザーソフト内で使用して いる登録ボードNo.値＋1の値を指定してください。
	無指定	使用可能枚数＝4となります。

注　意

・パラメータの設定順序に制約はありませんが、同一内容のパラメータを複数指定することはできません。

・MEWNET-P用ユーザープログラム*を使用する場合は、必ず「/P」を設定してください。

・登録送受信タイプコマンド**については、「2-1-8」および各機能コマンドの説明をお読みください。

・パラメータ設定に誤りがあった場合、右記のエラーメッセージが表示されます。正常に組み込まれた場合の表示については、次ページをお読みください。

```
MEWNET-Hﾘﾝｸｿﾌﾄ ﾊﾟﾗﾒｰﾀにｴﾗｰがあります
(INT-21以外での使用は出来ません)
```

補　足

・パラメータ設定状態は、デバイスドライバ組み込み時のメッセージ（MS-DOS起動時の表示）で確認できます。

■組み込み完了メッセージ

```
ＭＥＷＮＥＴ－Ｈリンクソフト（Ｖｅｒ１．１）を組み込みました
＝＝動作仕様＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝

    使用可能ＩＮＴ　ＮＯ.          ＝　２１Ｈまたは４０Ｈ
    ＭＥＷＮＥＴ－Ｐ用互換モード   ＝　サポートしていません
    パケット登録型同時使用可能数   ＝　　４
    リンクボード使用可能枚数       ＝　　４
    リンクソフトサイズ             ＝　４Ｄ４Ｅ（バイト）
```

・各機能パラメータ初期状態におけるMEWNET-Hリンクソフトメモリ使用サイズは下記の通りです。

■メモリ使用サイズ

PC98シリーズ用	約20kバイト
PC/AT互換機用	約20kバイト
FMRシリーズ用	約23kバイト

1-1-3 パラメータの設定例

(1)組み込み完了メッセージ表示無しで、INT初期状態使用、MEWNET-P使用不可、登録送受信要求タイプコマンド同時使用可能数＝４、リンクボード使用可能数＝４

```
DEVICE=MH**MS.SYS /O/H/K4/B4
```

または

```
DEVICE=MH**MS.SYS /O
```

(2)組み込み完了メッセージ表示無しで、INT60h使用、MEWNET-P使用不可、登録送受信要求タイプコマンド同時使用可能数＝４、リンクボード使用可能数＝４

```
DEVICE=MH**MS.SYS /O/I60/H/K4/B4
```

または

```
DEVICE=MH**MS.SYS /O/I60
```

(3)組み込み完了メッセージ表示有りで、INT60h使用、MEWNET-P使用可、リンクボード使用可能数＝４、登録送受信要求タイプコマンド同時使用可能数＝４

```
DEVICE=MH**MS.SYS /1/I60/P/K4/B4
```

または

```
DEVICE=MH**MS.SYS /I60/P
```

(4)組み込み完了メッセージ表示有りで、INT初期状態使用、MEWNET-P使用可、登録送受信要求タイプコマンド同時使用可能数＝４、リンクボード使用可能数＝４

```
DEVICE=MH**MS.SYS /1/P/K4/B4/
```

または

```
DEVICE=MH**MS.SYS /P
```

(5)組み込み完了メッセージ表示有りで、INT60h使用、MEWNET-P使用可、登録送受信要求タイプコマンド同時使用可能数＝0、リンクボード使用可能数＝1

```
DEVICE=MH**MS.SYS /1/I60/P/K0/B1
```

または

```
DEVICE=MH**MS.SYS /I60/P//K0/B1
```

(6)組み込み完了メッセージ表示無しで、INT初期状態使用、MEWNET-P使用不可、登録送受信要求タイプコマンド同時使用可能数＝8、リンクボード使用可能数＝2

```
DEVICE=MH**MS.SYS /O/H/K8/B2/
```

または

```
DEVICE=MH**MS.SYS /O/K8/B2
```

補　足

・上記「MH**MS.SYS」には、ご使用になるデバイスドライバ名を記述してください（P.6をお読みください）。

例：NEC PC98シリーズでMS-DOSデバイスドライバを使用される場合は、
「MHPCMS.SYS」と記述してください。

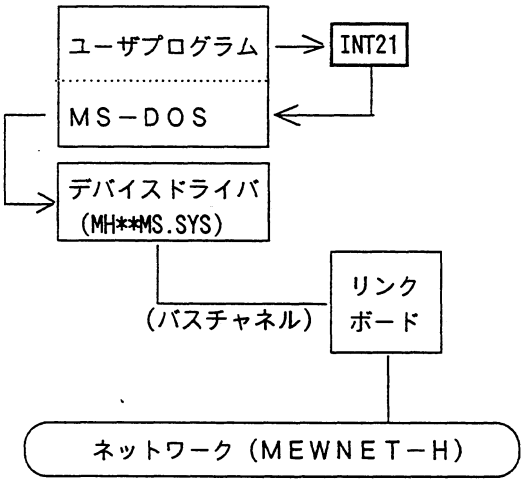
1－2 ユーザープログラムの作成方法

1-2-1 MS-DOSファンクションコール（INT21h）

MEWNET-Hリンクソフト（MH**MS.SYS）は、デバイスドライバであり、MS-DOSファンクションコール（INT21h）を実行することにより、各種機能コマンドを実行することができます。
コマンドの実行は、I/Oパラメータのセグメントアドレスとオフセットアドレスをds:dxレジスタにセットし、割り込みを実行します。

```
ah      ←44h
al      ←03h
bx      ←ﾌｧｲﾙﾊﾝﾄﾞﾙ
ds:dx   ←I/Oﾊﾟﾗﾒｰﾀｴﾘｱ先頭ｱﾄﾞﾚｽ
int     ←21h
```

使用するファンクションコールは、ファンクション44hで、コード03hを実行することにより、キャラクタ型デバイスドライバ（MH**MS.SYS）にデータを送ります。



コマンド実行後の結果（リターン値）は、axレジスタに格納されます。
0：正常終了時
その他：エラーコード（「3－9」参照）

■アセンブラプログラム
MASMのプログラムは、次のようになります（COMモデルの場合）。ファイルハンドルのオープンには、MS-DOSシステムコール（INT21h）のファンクション3Dhを使用します。

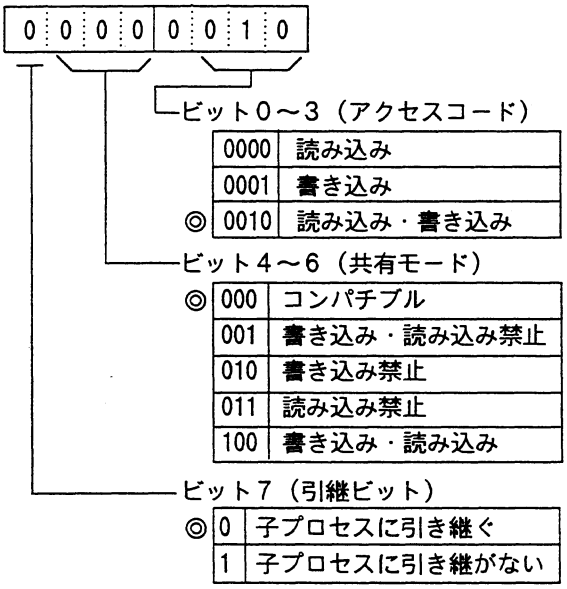
```
;-------- MEWNET-Hリンクソフト ------
code    segment
        assume cs:code,ds:code,es:code,ss:code
cr      equ     0dh
lf      equ     0ah
        org     100h
START:
        mov     ax,cs
        mov     ds,ax
        mov     es,ax
;------- ﾌｧｲﾙﾊﾝﾄﾞﾙ取得 ---
        mov     ah,3dh          'ﾌｧｲﾙﾊﾝﾄﾞﾙｵｰﾌﾟﾝ
        mov     al,02h
        mov     dx,offset FNAME
        int     21h
        mov     [FILHDL],ax  'axの値をFILHDLに格納
;------- 機能コマンド実行 ------
CMD:
        mov     ah,44h
        mov     al,03h
        mov     bx,[FILHDL]
        moc     dx,offset IO_PM ←
        int     21
;------- データ --------
FILHDL dw       ?               'ﾌｧｲﾙﾊﾝﾄﾞﾙ格納ｴﾘｱ
FNAME  db       'MH**MS  ',0        'ﾌｧｲﾙ ﾊﾟｽ名
;------- I/Oパラメータ -----
IO_PM  dw       0000h
       db       0,1
;
code    ends
        end     START
```

■ファイルハンドルのオープン

ファイルハンドルのオープンには、MS-DOSシステムコール（INT21h）のファンクション3Dhを使用します。

```
ah    ←3Dh
al    ←ファイルアクセスモード
ds:dx ←パス名の格納アドレス
int2  ←21h
```

ファイルアクセスモード（1バイト）には、通常2を指定します。



ファイルハンドルのオープンを実行した後のリターン値は、axレジスタに格納されます。

①キャリーフラグがセットされない場合、axレジスタにはファイルハンドルがセットされる。

②キャリーフラグがセットされた場合、axレジスタにはエラーコードがセットされます。

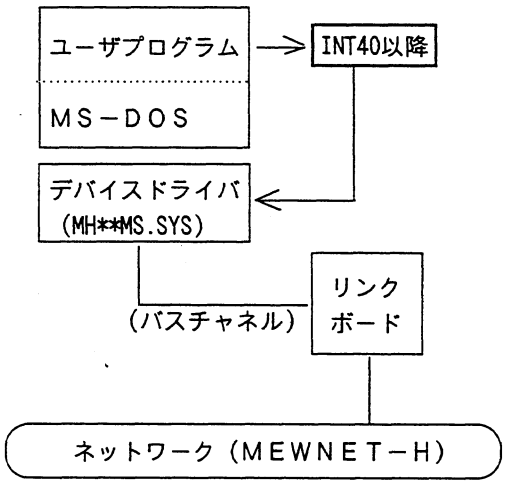
AX	エラー内容
01h	シェアリングモードが無効
02h	ファイル名が存在しない。
03h	パス名が無効か存在しない。
04h	オープンされているファイルが多すぎる。
05h	ディレクトリかボリュームＩＤをオープンしようとしたか、書き込み禁止ファイルに書き込もうとした。
0Ch	アクセスコードが0、1、2のいずれでもない。

1-2-2 ソフトウェア割り込み（INT40h以降）

MEWNET-Hリンクソフト（MH**MS.SYS）をINT40h以降のソフトウェア割り込みに割り当てることによっても、各種機能コマンドを実行することができます。この場合、必ずINT40h以降のユーザーに解放されているソフトウェア割り込みを使用してください。ハードウェア、BIOS、MS-DOSなどに割り当てられている割り込みは使用できません。

```
ds:dx ←I/Oﾊﾟﾗﾒｰﾀｴﾘｱ先頭ｱﾄﾞﾚｽ
int   ←40h以降
```

命令の実行は、I/Oパラメータのセグメントアドレスとオフセットアドレスをds:dxレジスタにセットし、割り込みを実行します。ソフトウェア割り込み（INT40h以降）への割り当ては、デバイスドライバのCONFIG.SYSへの登録時のパラメータスイッチで行います。割り込みを実行することにより、キャラクタ型デバイスドライバに、データが送られます。



コマンド実行後の結果（リターン値）は、axレジスタに格納されます。

0：正常終了時
その他：エラーコード（「3－9」参照）

■アセンブラプログラム

MASMのプログラムは、次のようになります（COMモデルの場合）。INT40h以降のソフトウェア割り込みを使用する場合です。

```
;-------- MEWNET-Hリンクソフト ------
code    segment
        assume cs:code,ds:code,es:code,ss:code
cr      equ     0dh
lf      equ     0ah
        org     100h
START:
        mov     ax,cs
        mov     ds,ax
        mov     es,ax
;------- 機能コマンド実行 ------
CMD:
        mov     dx,offset IO_PM
        int     60
;------- I/Oパラメータ -----
IO_PM   dw      0000h
        db      0,1
;
code    ends
        end     START
```

注　意

・この場合、CONFIG.SYSにデバイスドライバを登録する際に、システムコール時のINT No.をパラメータスイッチに設定しておいてください（「1-1-2」参照）。

```
DEVICE=MHPCMS.SYS /I60
                  └──INT60hの場合
```

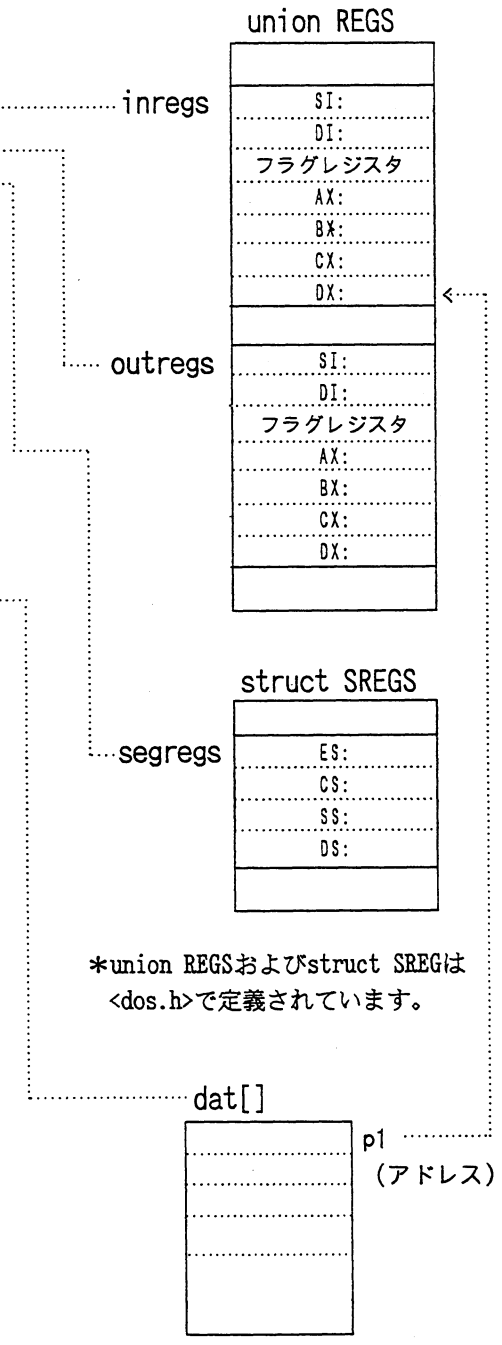
■指定できるINT No.

PC98シリーズ用	int40h以降
PC/AT互換機用	int60h以降
FMRシリーズ用	int50h以降

■Cプログラム

Cのプログラムでは、int86x関数を使用します。INT40h以降のソフトウェア割り込みを使用する場合、次のようになります。

```
/***** MEWNET-Hリンクソフト *****/
#include <dos.h>
#include <stdio.h>
#include <stdlib.h>

static union REGS inregs;
static union REGS outregs;
struct SREGS segregs;
static char far *p1;
static unsigned char dat[48];

void main(void)
{
        int err;

        /* ----- 機能コマンド実行 ----- */

        dat[0]=0x00;
        dat[1]=0x10;
        dat[2]=0;
        dat[3]=1;

        if((err = mew_cmd()) != 0)
        {
                printf("実行エラー return code = %x¥n",err);
                exit(-1);
        }
        printf("実行正常完了　ＯＫ！¥n");

int mew_cmd(void)
{
        p1 = (char far *)&dat[0];

        inregs.x.dx = FP_OFF(p1);
        segregs.ds = FP_SEG(p1);

        int86x(0x60,&inregs,&outregs,&segregs);
        return(outregs.x.ax);
}
```



1－3 ユーザープログラム作成上の注意

MEWNET-Hリンクソフトは、内部で割り込みコントローラを使用しています。したがって、ユーザープログラムで独自に割り込み処理を使用する場合十分に注意してください。

使用している割り込みベクタ
・インターバルタイマ割り込み
　PC98シリーズ……int1Ch
　PC/ATシリーズ……int1Ah
　FMRシリーズ……int96h
・キーボード割り込み
　PC98シリーズ……int18h
　（PC/AT･FMRは使用していません。）

(1)MEWNET-Hリンクソフトを割り込み処理内（インターバルタイマ等）で使用する場合、MEWNET-Hリンクソフトの次の機能は使用できません。

①インターバルタイマ割り込み使用時
・MEWTOCOL-COM、MEWTOCOL-DAT、CHの各タイムアウト付きモード。
・送受信チャンネル使用許可/禁止。
・ＰＣリンク受信。

［理由］
MEWNET-Hリンクソフト内部で、
・タイムアウト時に割り込みを使用しています。
・送受信チャンネル使用許可/禁止時に割り込みを使用しています。
・ＰＣリンク受信時に割り込みを使用しています。

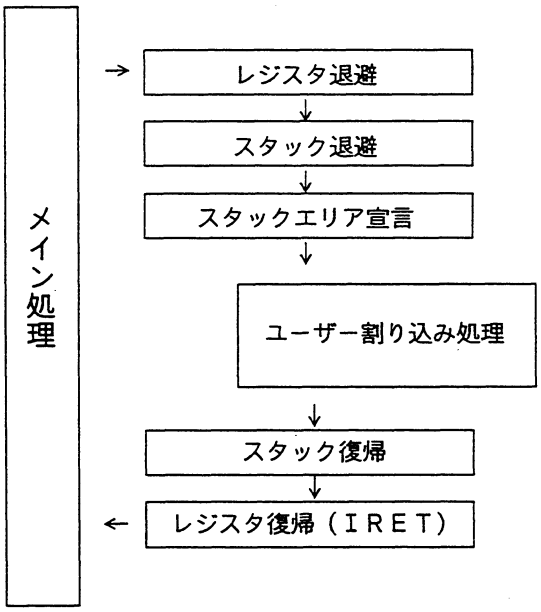
②キーボード割り込み使用時
・MEWTOCOL-COM、MEWTOCOL-DAT、CHの各完了待ちモード。

［理由］
MEWNET-Hリンクソフト内部で、
・完了待ちモード時に割り込みを使用しています。

＊基本的には、ユーザープログラム内の割り込み処理内では、MEWNET-Hリンクソフトを使用しないようにするのが安全です。

(2)MEWNET-Hリンクソフトを割り込み処理内（インターバルタイマ等）で使用している場合、メイン処理ループ内ではMEWNET-Hリンクソフトは使用しないでください。使用した場合、動作は保証されません。

(3)ユーザーソフトで、MEWNET-Hリンクソフト以外の機能を使用する合は、割り込み処理内部で、独自にスタックエリアを確保してください。この時、必ずスタックポインタ（ＳＰ）、スタックセグメント（ＳＳ）のＰＵＳＨ、ＰＯＰは行ってください。



2章
機能コマンド概要

2－1 機能コマンド概要

2-1-1 機能コマンド一覧

機能	機能コマンド名	機能コマンドコード
通信開始手続き	MEWNET-Hリンクソフト初期設定	Ｈ１０００
	使用ボード登録および起動	Ｈ１１００
	CH OPEN CLOSE	Ｈ１６０ｘ
ＰＣリンク	PC LINK OPEN/CLOSE	Ｈ１６１ｘ
	ＰＣリンクパラメータ読み出し	Ｈ１９００
	PC LINK WRITE	Ｈ１５４ｘ
	PC LINKランダムWRITE	Ｈ１５５ｘ
	PC LINK READ	Ｈ１４４ｘ
	PC LINKＫランダムREAD	Ｈ１４５ｘ
コンピュータリンク	MEWTOCOL-COM WRITE→READ	Ｈ１５６ｘ
	MEWTOCOL-COM WRITE	Ｈ１５２ｘ
	MEWTOCOL-COM READ	Ｈ１４２ｘ
シリアル伝送	シリアル伝送機能の使用登録／解除	Ｈ１８０ｘ
	多重接続マスター用制御コマンドのWRITE/READ	Ｈ１８１ｘ
	シリアルデータ送信	Ｈ１８２ｘ
	シリアルデータ受信	Ｈ１８３ｘ
データ転送	MEWTOCOL-DAT WRITE	Ｈ１５３ｘ
	MEWTOCOL-DAT READ	Ｈ１４３ｘ
コンピュータ間通信	コンピュータ間通信WRITE（文字型）	Ｈ１５０ｘ
	コンピュータ間通信WRITE（整数型）	Ｈ１５１ｘ
	コンピュータ間通信READ（文字型）	Ｈ１４０ｘ
	コンピュータ間通信READ（整数型）	Ｈ１４１ｘ
登録受信要求解除	登録受信要求タイプコマンド（タイムアウト無し）解除	Ｈ１４６０
コントロール／ステータスレジスタアクセス	コントロールレジスタWRITE	Ｈ１２００
	コントロール／ステータスレジスタREAD	Ｈ１３００

2-1-2 通信開始手続きの機能コマンド

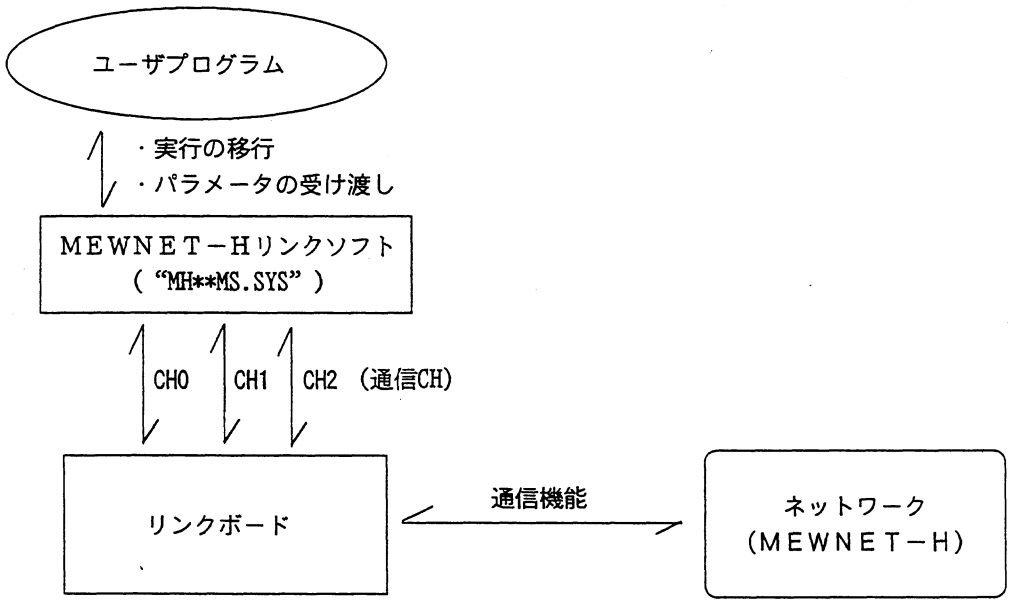
MEWNET-Hリンクソフトを使用してMEWNET-Hリンクボードの各機能を使用するためには、先に通信開始手続きの機能コマンドを実行しておかなければなりません。

(1)MEWNET-Hリンクソフト初期設定
MEWNET-Hリンクソフトの内部領域を初期化し、使用ボード枚数を設定します。

(2)使用ボード登録および起動
ボードの割り込みNo.、および各ボード毎のセグメントNo.を登録することにより、登録ボード番号を取得します。以降のコマンドの実行ではこの登録ボード番号を使用します。

(3)通信チャネルの使用許可／禁止
通信コマンドで使用する送受信チャネルの使用の許可または禁止を設定します。送受信チャネルは、コンピュータリンク、コンピュータ間通信、データ転送の各用途に設けられていて、設定前の初期値ではコンピュータリンクだけが使用許可になっています。

チャネル番号(CH No.)	用途
0	コンピュータリンク
1	コンピュータ間通信
2	データ転送



■MEWNET-Hリンクソフト初期設定

コマンドコード	機能
H1000	MEWNET-Hリンクソフトを使用するために、使用ボード枚数を設定します。

■使用ボード登録および起動

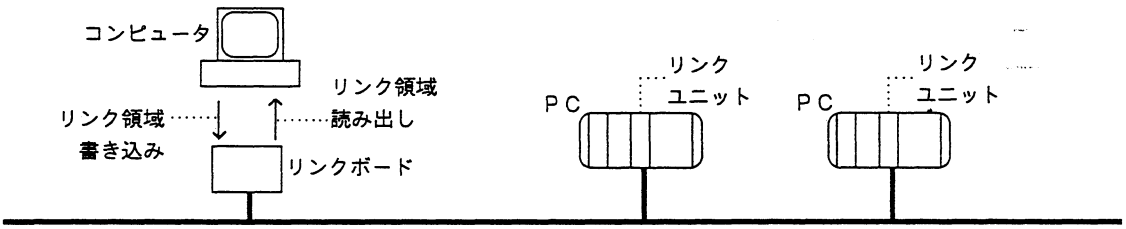
コマンドコード	機能
H1100	ボードの割り込みNo.、および各ボード毎の使用メモリエリアのセグメントNo.を登録します。 これにより、MEWNET-Hリンクソフトで管理する登録ボードNo.が得られます。

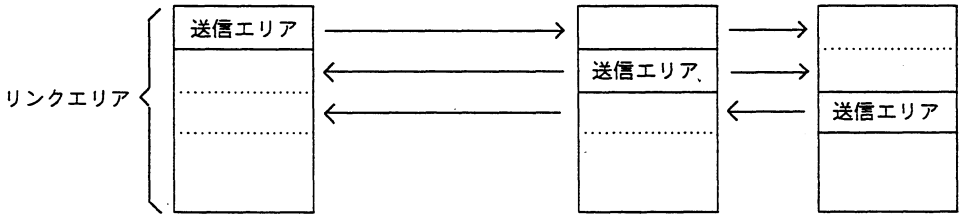
■CH OPEN/CLOSEコマンドのタイプ

コマンドコード	タイプ
H1600	指定ＣＨの送受信を禁止します。
H1601	指定ＣＨの送受信を許可します。
H1602	指定ＣＨの現在状態（禁止または許可）を読み出します。

2-1-3 ＰＣリンクの機能コマンド

・MEWNET-Hリンクソフトを使用してユーザプログラムを作成することにより、リンクボードのリンク領域の書き込みおよび読み出しができます。（書き込みは自局の送信エリアのみ。）
・ＰＣリンクでは、通信局間でリレー、レジスタの情報をサイクリックに逐次伝送します。リンクボードの場合、ボード上の4,096点分のリレーと4,096ワード分のレジスタがリンクされます。
・リンク領域の他局への送信エリアは、MEWNET-Hシステム設定ソフト（ＵＳ２－Ｈ）を使用して設定します。設定ソフトにより、ＰＣリンク領域の割り付けが設定されると、、ＰＣおよびリンクボードは自動的にデータの共有を行います（リンクのための特別なプログラムは不要です）。





■ＰＣリンク機能仕様

項目	仕様
リンク局数	最大64局／１ネットワーク
リンク点数	リンクリレー：最大4,096点（256ワード／512バイト） データリンク：最大4,096ワード（8,192バイト）
１局あたりの 送信点数	リンクリレー：最大4,096点（ワード単位） データリンク：最大4,096ワード（ワード単位）

■PC LINK OPEN/CLOSEコマンドのタイプ

コマンドコード	タイプ
Ｈ１６１０	ＰＣリンク使用禁止
Ｈ１６１１	ＰＣリンク使用許可

■ＰＣリンクパラメータ読み出し

コマンドコード	機能
Ｈ１９００	ＰＣリンクのパラメータ（動作設定）読み出し

■PC LINK WRITEコマンドのタイプ

コマンドコード	タイプ	
Ｈ１５４０	送信完了待ち（PC98シリーズのみKey Break付き）	PC LINK OPEN機能付
Ｈ１５４１	送信完了待ち（タイムアウト付き）	PC LINK OPEN機能付
Ｈ１５４２	送信要求のみ	PC LINK OPEN機能付
Ｈ１５４３	送信完了センス（Ｈ１５４２コマンドの完了検知用）	
Ｈ１５４４	データ書き込み（送信要求無し）	
Ｈ１５４５	ＰＣリンク送信停止	

■PC LINKランダムWRITEコマンドのタイプ

コマンドコード	タイプ	
Ｈ１５５０	PC LINKランダムWRITE停止	
Ｈ１５５１	PC LINKランダムWRITE起動	PC LINK OPEN機能付

■PC LINK READコマンドのタイプ

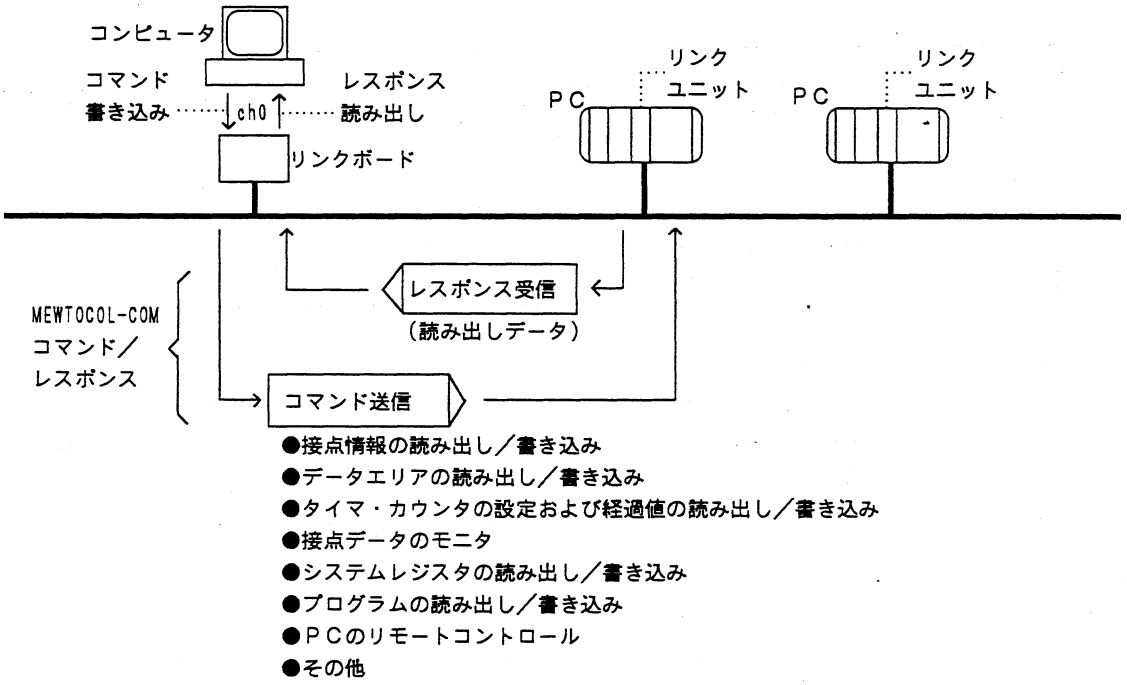
コマンドコード	タイプ	
Ｈ１４４０	受信要求（データは読み出さない）	PC LINK OPEN機能付
Ｈ１４４１	受信要求＋データ読み出し（タイムアウト付き）	PC LINK OPEN機能付
Ｈ１４４２	受信完了センス（完了時データ読み出し付き）	
Ｈ１４４４	データ読み出し（受信要求なし）	

■PC LINKランダムREADコマンドのタイプ

コマンドコード	タイプ	
Ｈ１４５０	PC LINKランダムREAD停止	
Ｈ１４５１	PC LINKランダムREAD起動	PC LINK OPEN機能付

2-1-4 コンピュータリンクの機能コマンド

・MEWNET-Hリンクソフトを使用してユーザプログラムを作成することにより、会話型専用手順MEWTOCOL-COMのコマンドをＰＣに送信し、ＰＣからはレスポンスを受信します。
・MEWTOCOL-COMコマンド／レスポンスの送受信により、接点情報の読み出し／書き込み、データエリアの読み出し／書き込み、タイマ・カウンタの設定および経過値の読み出し／書き込み、接点データのモニタ、システムレジスタの読み出し／書き込み、プログラムの読み出し／書き込み、ＰＣのリモートコントロール、その他を実行します。
・ユーザプログラムから、最大４階層までのネットワークを横断してコマンド送信先ＰＣを指定することができます。



注　意

・リンクボードのRS232Cインターフェイスをコンピュータリンク機能に設定した場合の使用方法、およびモデムを使用するコンピュータリンクの使用方法については、MEWNET-Hリンクユニットのマニュアルをお読みください。

■コンピュータリンク機能仕様

項目	仕様
リンク局数	最大64局／１ネットワーク
ネットワーク間接続	最大４階層（自局が属するネットワークを含む）　　深さ0～3
通信データ量	最大2048バイト／パケット　複数パケット送受信可能
通信形態	コンピュータ（上位）局→ＰＣ（下位）局
通信手順	MEWTOCOL-COM（会話型専用手順） コンピュータ側　：コマンド送信 ＰＣ側　　：レスポンス送信（応答）

■MEWTOCOL-COM WRITE→READコマンドのタイプ

コマンドコード	タイプ
H1560	送受信完了待ち（PC98シリーズのみKEY Break付き）
H1561	送受信完了待ち（タイムアウト付き）
H1565	登録送受信要求（タイムアウト付き）

■MEWTOCOL-COM WRITEコマンドのタイプ

コマンドコード	タイプ
H1520	送信完了待ち（PC98シリーズのみKEY Break付き）
H1521	送信完了待ち（タイムアウト付き）
H1522	送信要求のみ
H1523	送信完了センス
H1525	登録送信要求（タイムアウト付き）

■MEWTOCOL-COM READコマンドのタイプ

コマンドコード	タイプ
H1420	受信完了待ち（PC98シリーズのみKEY Break付き）
H1421	受信完了待ち（タイムアウト付き）
H1422	受信完了センス（センス時データ読み出し付き）
H1423	受信バッファクリア（空読み）
H1424	登録受信要求（タイムアウト無し）
H1425	登録受信要求（タイムアウト付き）

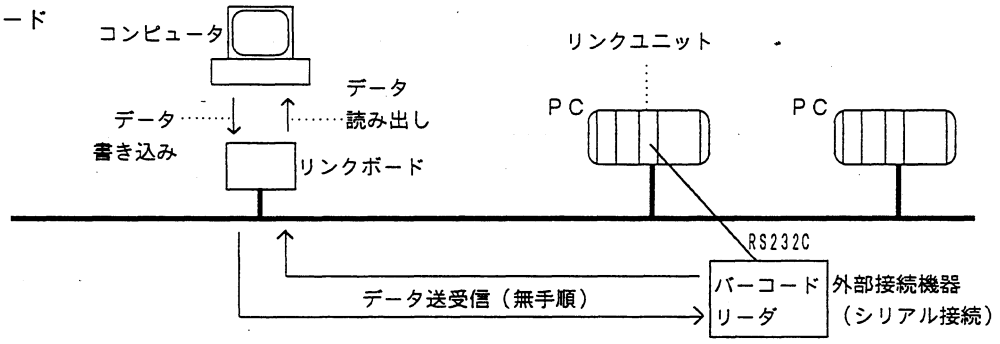
注　意

・登録送受信要求タイプ、登録送信要求タイプ、登録受信要求タイプの機能コマンドには、同時に使用できる（実行状態にある）コマンド数に制限があります。同時に使用できるコマンド数は、MEWNET-Hリンクソフト（デバイスドライバ）のCONFIG.SYSへの登録時のパラメータスイッチで、0～16まで指定できます。CONFIG.SYSでの指定がない場合は、同時に実行できるコマンド数は4つまでです。CONFIG.SYSでの指定については、「1-1-2」をお読みください。

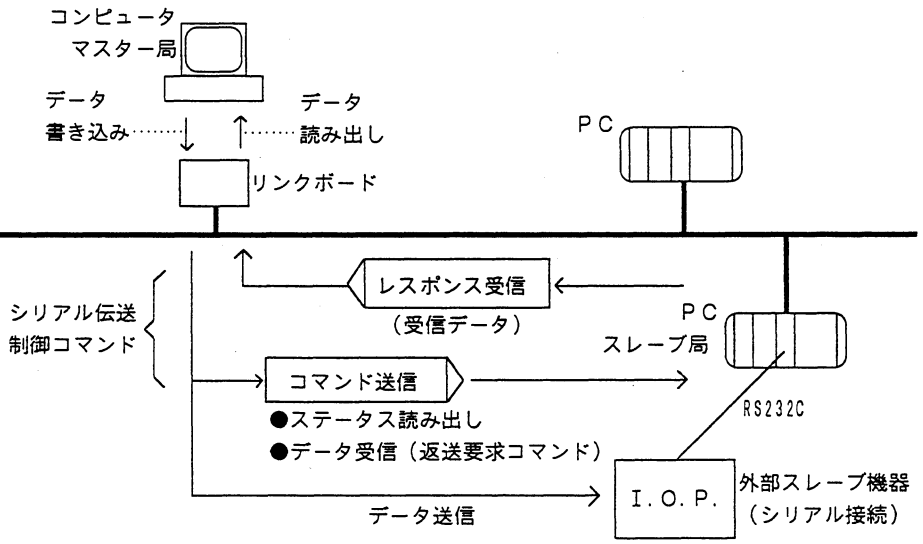
2-1-5 シリアル伝送の機能コマンド

・MEWNET-Hリンクソフトを使用してユーザプログラムを作成することにより、リンクボードを装着しているコンピュータとネットワーク上のリンクユニット（またはリンクボード）のRS232Cシリアルインターフェイスに接続された外部機器との間でデータの送受信ができます。
・シリアル伝送には、次の２つの動作モードがあります。
　(1) １：１通信の「単一接続モード」
　(2) １：ｎ通信の「多重接続モード」
・ユーザプログラムで制御コードを送信することにより、最大４階層を横断して相手先リンクユニット（またはリンクボード）を指定できます。
・動作モード、相手先通信局、各局のRS232Cインターフェイスの通信条件は、ＭＥＷＮＥＴ－Ｈシステム設定ソフト（ＵＳ２－Ｈ）を使用して、あらかじめ設定しておきます。

■単一接続モード



■多重接続モード



■シリアル伝送機能仕様

項目	仕様
リンク局数	最大64局／１ネットワーク
ネットワーク間接続	最大４階層（自局が属するネットワークを含む）　　深さ0～3
通信データ量	最大2046バイト／パケット（ターミネータを除くユーザデータ部）
通信形態	コンピュータマスター局←→ＰＣスレーブ局RS232Cインターフェイス 単一接続モード　１：１接続　（階層間対応：最大４階層） 多重接続モード　１：ｎ接続　（階層間対応：最大４階層）
通信手順	無手順または会話型専用手順（シリアル伝送制御コマンド）

■シリアル伝送機能の使用登録／解除コマンドのタイプ

コマンドコード	タイプ
H1800	全解除
H1801	解除（ボード番号指定）
H1802	登録

■多重接続マスター用制御コマンドのWRITE/READコマンドのタイプ

コマンドコード	タイプ
H1810	送受信完了待ち（PC98のみKeyBreak付き）
H1811	送受信完了待ち（タイムアウト付き）
H1815	登録送受信要求（タイムアウト付き）

■シリアルデータ送信コマンドのタイプ

コマンドコード	タイプ
H1820	送信完了待ち（PC98のみKeyBreak付き）
H1821	送信完了待ち（タイムアウト付き）
H1822	送信要求のみ
H1823	送信完了センス（H1822の完了検知用）
H1826	強制停止（再送／スレーブ時の送信中断用）

■シリアルデータ受信コマンドのタイプ（単一接続時のデータ受信）

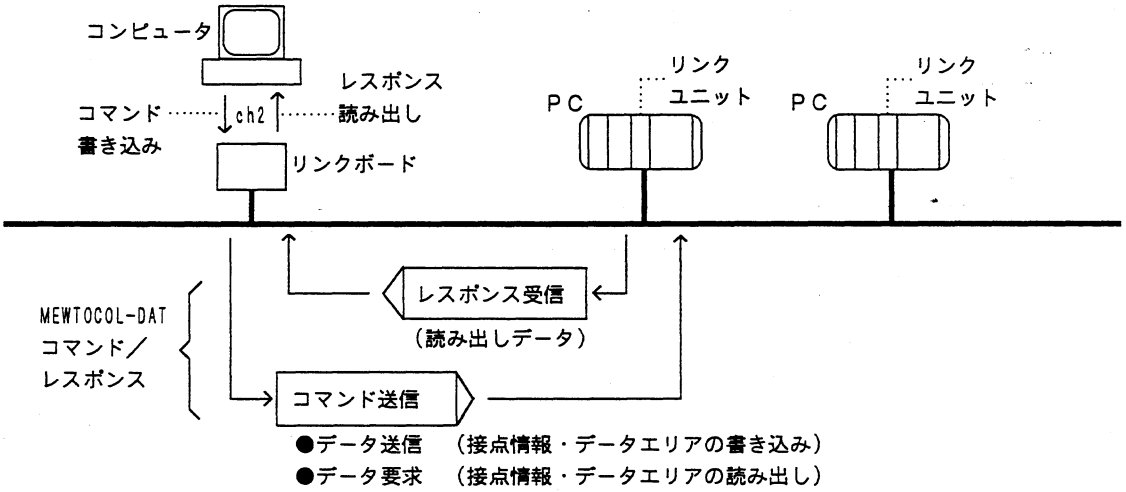
コマンドコード	タイプ
H1830	受信完了待ち（PC98のみKeyBreak付き）
H1831	受信完了待ち（タイムアウト付き）
H1832	受信完了センス
H1833	受信データ破棄

注　意

・登録送受信要求タイプの機能コマンドには、同時に使用できる（実行状態にある）コマンド数に制限があります。同時に使用できるコマンド数は、MEWNET-Hリンクソフト（デバイスドライバ）のCONFIG.SYSへの登録時のパラメータスイッチで、0～16まで指定できます。CONFIG.SYSでの指定がない場合は、同時に実行できるコマンド数は4つまでです。CONFIG.SYSでの指定については、「1-1-2」をお読みください。

2-1-6 データ転送の機能コマンド

・MEWNET-Hリンクソフトを使用してユーザプログラムを作成することにより、会話型専用手順MEWTOCOL-DATのコマンドをＰＣに送信し、ＰＣからはレスポンスを受信します。
・MEWTOCOL-DATコマンド／レスポンスの送受信により、ＰＣの接点情報およびデータエリアの読み出し／書き込みを実行します。
・ユーザープログラムから、最大４階層までのネットワークを横断してコマンド送信先ＰＣを指定することができます。



■データ転送機能仕様

項目	仕様
リンク局数	最大64局／１ネットワーク
ネットワーク間接続	最大４階層（自局が属するネットワークを含む）　深さ0～3
通信データ量	最大1020ワード／パケット
通信形態	コンピュータ局←→ＰＣ局（ＰＣからのデータの受信可能）
通信手順	MEWTOCOL-DAT（会話型専用手順） コンピュータ側　：コマンド送信 ＰＣ側　　：レスポンス送信（応答）

■MEWTOCOL-DAT WRITEコマンドのタイプ

コマンドコード	タイプ
Ｈ１５３０	送信完了待ち（PC98シリーズのみKEY Break付き）
Ｈ１５３１	送信完了待ち（タイムアウト付き）
Ｈ１５３２	送信要求のみ
Ｈ１５３３	送信完了センス
Ｈ１５３５	登録送信要求（タイムアウト付き）

■MEWTOCOL-DAT READコマンドのタイプ

コマンドコード	タイプ
Ｈ１４３０	受信完了待ち（PC98シリーズのみKEY Break付き）
Ｈ１４３１	受信完了待ち（タイムアウト付き）
Ｈ１４３２	受信完了センス
Ｈ１４３３	受信バッファクリア（空読み）
Ｈ１４３４	登録受信要求（タイムアウトなし）
Ｈ１４３５	登録受信要求（タイムアウト付き）

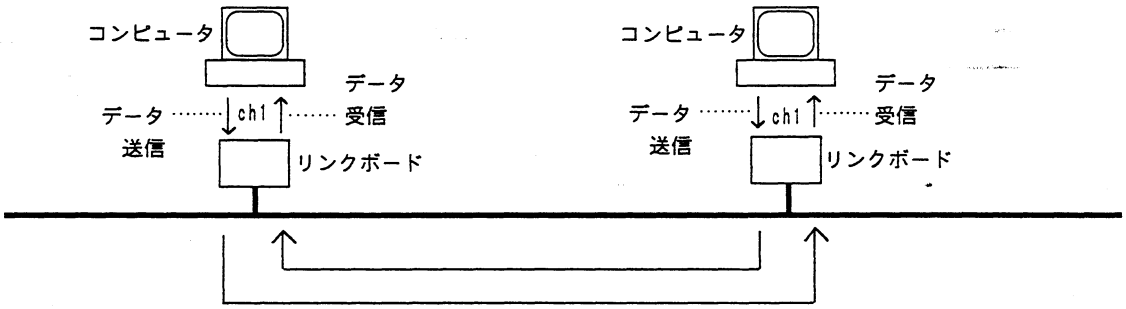
注　意

・ＰＣが発行したＳＥＮＤ・ＲＥＣＶ命令を受信した場合は、ユーザプログラムで登録受信要求タイプの機能コマンドを実行することにより、送受信バッファ内のデータを読み出すことができます。

・登録送信要求タイプ、登録受信要求タイプの機能コマンドには、同時に使用できる（実行状態にある）コマンド数に制限があります。同時に使用できるコマンド数は、MEWNET-Hリンクソフト（デバイスドライバ）のCONFIG.SYSへの登録時のパラメータスイッチで、0～16まで指定できます。CONFIG.SYSでの指定がない場合は、同時に実行できるコマンド数は4つまでです。CONFIG.SYSでの指定については、「1-1-2」をお読みください。

2-1-7 コンピュータ間通信の機能コマンド

・MEWNET-Hリンクソフトを使用してユーザプログラムを作成することにより、ネットワーク上のコンピュータ間で相互にデータの送受信ができます（バイナリコードおよび文字コードを透過的に伝送できます）。
・ユーザプログラムから、最大４階層までのネットワークを横断して相手先コンピュータを指定することができます。



■コンピュータ間通信機能仕様

項目	仕様
リンク局数	最大64局／１ネットワーク
ネットワーク間接続	最大４階層（自局が属するネットワークを含む）　深さ0～3
通信データ量	最大2048バイト／パケット
通信形態	コンピュータ局→コンピュータ局 １：１通信
通信手順	ユーザ手順に基づく（Binary/ASCII透過的）

■コンピュータ間通信WRITE（文字型）コマンドのタイプ

コマンドコード	タイプ
H1500	送信完了待ち（PC98シリーズのみKEY Break付き）
H1501	送信完了待ち（タイムアウト付き）
H1502	送信要求のみ
H1503	送信完了センス
H1505	登録送信要求（タイムアウト付き）

■コンピュータ間通信WRITE（整数型）コマンドのタイプ

コマンドコード	タイプ
H1510	送信完了待ち（PC98シリーズのみKEY Break付き）
H1511	送信完を待ち（タイムアウト付き）
H1512	送信要求のみ
H1513	送信完了センス
H1515	登録送信要求（タイムアウト付き）

■コンピュータ間通信READ（文字型）コマンドのタイプ

コマンドコード	タイプ
H1400	受信完了待ち（PC98シリーズのみKEY Break付き）
H1401	受信完了待ち（タイムアウト付き）
H1402	受信完了センス（センス時受信データ読み出し付き）
H1403	受信バッファクリア（空読み）
H1404	登録受信要求（タイムアウト無し）
H1405	登録受信要求（タイムアウト付き）

■コンピュータ間通信READ（整数型）コマンドのタイプ

コマンドコード	タイプ
H1410	受信完了待ち（PC98シリーズのみKEY Break付き）
H1411	受信完了待ち（タイムアウトなし）
H1412	受信完了センス（センス時受信データ読み出し付き）
H1413	受信バッファクリア（空読み）
H1414	登録受信要求（タイムアウト無し）
H1415	登録受信要求（タイムアウト付き）

注　意

・登録送信要求タイプ、登録受信要求タイプの機能コマンドには、同時に使用できる（実行状態にある）コマンド数に制限があります。同時に使用できるコマンド数は、MEWNET-Hリンクソフト（デバイスドライバ）のCONFIG.SYSへの登録時のパラメータスイッチで、0～16まで指定できます。CONFIG.SYSでの指定がない場合は、同時に実行できるコマンド数は4つまでです。CONFIG.SYSでの指定については、「1-1-2」をお読みください。

2-1-8 登録受信要求解除の機能コマンド

・MEWNET-Hリンクソフトの各種機能コマンドの＜登録送信要求タイプ＞＜登録受信要求タイプ＞では、コマンド実行時に送信要求または受信要求の登録みを行い、実際にアクセスが成立した時点（送信が許可された時点または受信データを受け取った時点）で、割り込み機能により受信処理（ボードに対する送信要求または受信要求）を再開します。
・＜登録送信要求タイプ＞＜登録受信要求タイプ＞のタイムアウト付きコマンドでは、指定タイム経過をもって、送信要求または受信要求の登録が解除されます。
・＜登録受信要求タイプ＞のタイムアウト無しコマンドを実行すると、受信データを受け取るまで登録は解除されません。登録を解除する場合は、「登録受信要求タイプコマンド（タイムアウト無し）解除」コマンドを実行する必要があります。

■登録送受信要求タイプコマンドについて

・＜登録送信要求タイプ＞＜登録受信要求タイプ＞のタイムアウト付きコマンド
H１５６５　コンピュータリンクMEWTOCOL-COM WRITE→READ
H１５２５　コンピュータリンクMEWTOCOL-COM WRITE
H１４２５　コンピュータリンクMEWTOCOL-COM READ
H１８１５　シリアルデータ伝送多重接続マスター用制御コマンドWRITE→READ
H１５３５　データ転送MEWTOCOL-DAT WRITE
H１４３５　データ転送MEWTOCOL-DAT READ
H１５０５　コンピュータ間通信WRITE（文字型）
H１５１５　コンピュータ間通信WRITE（整数型）
H１４０５　コンピュータ間通信READ（文字型）
H１４１５　コンピュータ間通信READ（整数型）

・＜登録受信要求タイプ＞のタイムアウト無しコマンド
H１４２４　コンピュータリンクMEWTOCOL-COM READ
H１４３４　データ転送MEWTOCOL-DAT READ
H１４０４　コンピュータ間通信READ（文字型）
H１４１４　コンピュータ間通信READ（整数型）

以上の登録送受信要求タイプの機能コマンドには、同時に使用できる（実行状態にある）コマンド数に制限があります。同時に使用できるコマンド数は、MEWNET-Hリンクソフト（デバイスドライバ）のCONFIG.SYSへの登録時のパラメータスイッチで、0～16まで指定できます。CONFIG.SYSでの指定がない場合は、同時に実行できるコマンド数は4つまでです。CONFIG.SYSでの指定については、「1-1-2」をお読みください。

■登録受信要求解除の機能コマンド

コマンドコード	機能
H１４６０	登録受信要求タイプコマンド（タイムアウト無し）解除

2-1-9 コントロール／ステータスレジスタのアクセス

・MEWNET-Hリンクソフトは、リンクボードごとにコントロールレジスタとステータスレジスタを管理することができます。
・コントロールレジスタには、通信CHごとの送受信完了待ちタイムアウト値を設定することができます。コントロールレジスタを設定するには、ユーザプログラム上で機能コマンド「コントロールレジスタWRITE」を実行します。
・ステータスレジスタはに、リンクボードの動作状態、通信CHごとの各通信機能の動作状態が格納されます。ステータスレジスタの読み出しには、ユーザプログラム上で機能コマンド「コントロール／ステータスレジスタREAD」を実行します。

■コントロールレジスタWRITE

コマンドコード	機能
H1200	コントロールレジスタの内容を書き込みます。

■コントロール／ステータスレジスタREAD

コマンドコード	機能
H1300	コントロールレジスタおよびステータスレジスタの内容を読み出します。

2－2 機能コマンドコード一覧表

コマンドコード	機能	備考
Ｈ１０００	初期設定	使用ボード数の指定
Ｈ１１００	使用ボード登録および起動	登録後ON LINE処理
Ｈ１２００	コントロールレジスタWRITE	送信タイムアウト値設定
Ｈ１３００	コントロール／ステータスレジスタREAD	動作状態モニタ用
Ｈ１４２ｘ	MEWTOCOL-CON READ	コンピュータリンク機能で使用
Ｈ１５２ｘ	MEWTOCOL-COM WRITE	
Ｈ１５６ｘ	MEWTOCOL-COM WRITE→READ	
Ｈ１４３ｘ	MEWTOCOL-COM READ	データ転送機能で使用
Ｈ１５３ｘ	MEWTOCOL-COM WRITE	
Ｈ１４４ｘ	PC LINC READ	ＰＣリンク機能で使用
Ｈ１５４ｘ	PC LINK WRITE	
Ｈ１４５ｘ	PC LINKランダムREAD	
Ｈ１５５ｘ	PC LINKランダムWRITE	
Ｈ１４０ｘ	コンピュータ間通信READ（文字型）	コンピュータ間通信機能で使用
Ｈ１５０ｘ	コンピュータ間通信WRITE（文字型）	
Ｈ１４１ｘ	コンピュータ間通信READ（整数型）	
Ｈ１５１ｘ	コンピュータ間通信WRITE（整数型）	
Ｈ１４６０	登録受信要求タイプコマンド（タイムアウト無し）解除	登録受信要求解除
Ｈ１６０ｘ	CH OPEN/CLOSE	受信ＣＨの指定
Ｈ１６１ｘ	PC LINK OPEN/CLOSE	ＰＣリンク使用指定用
Ｈ１８ｘｘ	シリアル伝送機能	シリアル伝送機能用
Ｈ１９００	ＰＣリンクパラメータ読み出し	ＰＣリンク機能で使用

3章
機能コマンドリファレンス

3－1 通信開始手続きの機能コマンド

3-1-1 リンクソフト初期設定

通信開始手続き／MEWNET-Hリンクソフト初期設定（コマンドコード&H1000）

機　能

リンクソフトを使用するための初期設定を行います。

解　説

・MEWNET-Hリンクソフトで使用する内部領域を初期化し、使用するリンクボードの枚数を登録します。
・機能コマンドを実行するには、各種プログラミング言語でI/Oパラメータのセグメントアドレスとオフセットアドレスをds:dxレジスタにセットし、割り込みを実行するユーザープログラムを作成してください。
・コマンド実行後の結果（リターン値）は、axレジスタに格納されます。
　0：正常終了時
　その他：エラーコード（「3－9」参照）

注　意

・MEWNET-Hリンクソフトを使用する時は、必ず一番最初に実施してください。
・本コマンドを途中で実施した場合、各登録内容（使用ボード数登録）を初期化します。

文　例

●MS-DOSファンクションコール（int21h）

```
;------- 初期設定 ------
INI:
        mov     bx,[FILHDL]
        mov     ah,44h
        mov     al,03h
        mov     dx,offset INIT
        int     21h
;------- I/Oﾊﾟﾗﾒｰﾀ ------
INIT    dw      1000h
        db      0,1
        dw      22 dup(?)
```

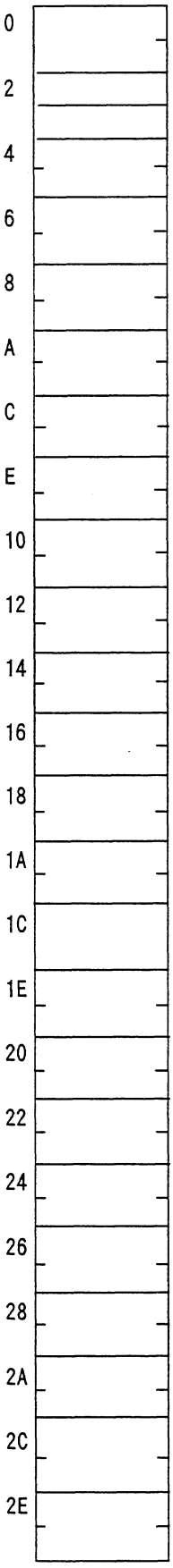
●ソフトウェア割り込み（int60h）

```
;------- 初期設定 ------
INI:
        mov     dx,offset INIT
        int     60h
;------- I/Oﾊﾟﾗﾒｰﾀ ------
INIT    dw      1000h
        db      0,1
        dw      22 dup(?)
```

I/Oパラメータ

△：設定する［渡す値］
▼：実行後に格納される［得る値］

dsﾚｼﾞｽﾀ ←I/Oパラメータの先頭セグメントアドレス
dxﾚｼﾞｽﾀ ←I/Oパラメータの先頭オフセットアドレス



△［コマンドコード］&H1000

△ 設定値：00固定で使用してください。
△［使用ボード枚数］1～4

3-1-2 使用ボード登録および起動

通信開始手続き／使用ボード登録および起動（コマンドコード&H1100）

機　能

リンクボードの登録と起動を行います。

解　説

・使用するMEWNET-Hリンクボードのセグメントおよび割り込みNo.をMEWNET-Hリンクソフトが管理する［登録ボードNo.］に登録します。
・エラーが無ければ、ボードの起動を行い、通信できる状態になります。
・機能コマンドを実行するには、各種プログラミング言語でI/Oパラメータのセグメントアドレスとオフセットアドレスをds:dxレジスタにセットし、割り込みを実行するユーザープログラムを作成してください。
・コマンド実行後の結果（リターン値）は、axレジスタに格納されます。
　0：正常終了時
　その他：エラーコード（「3－9」参照）

注　意

・以降のコマンドで登録ボードNo.を使用する際には、本コマンドで指定した［登録ボードNo.］を指定します。
・使用ボードの［使用セグメントアドレスNo.］［割り込みNo.］については、次ページ以降をお読みください。

文　例

●MS-DOSファンクションコール（int21h）

```
;------- ﾎﾞｰﾄﾞ登録･起動------
BODST:
        mov     bx,[FILHDL]
        mov     ah,44h
        mov     al,03h
        mov     dx,offset BODST
        int     21h
;------- I/Oﾊﾟﾗﾒｰﾀ ------
BODST   dw      1100h
        db      0,0,0,0
        dw      21 dup(?)
```

●ソフトウェア割り込み（int60h）

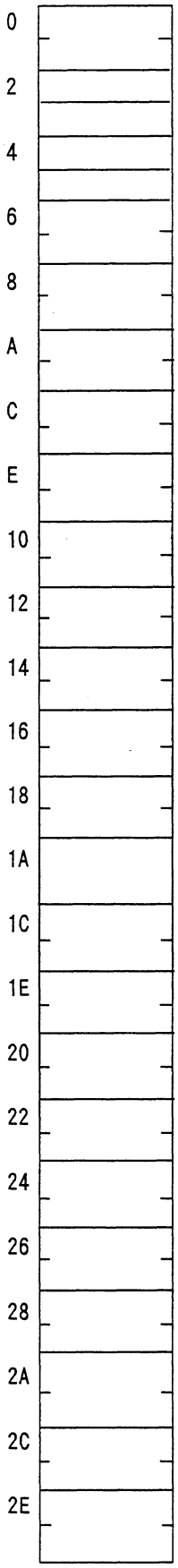
```
;------- ﾎﾞｰﾄﾞ登録･起動------
BODST:
        mov     dx,offset BODST
        int     60h
;------- I/Oﾊﾟﾗﾒｰﾀ ------
BODST   dw      1100h
        db      0,0,0,0
        dw      21 dup(?)
```

I/Oパラメータ

△：設定する［渡す値］
▼：実行後に格納される［得る値］

dsレジスタ ←I/Oパラメータの先頭セグメントアドレス
dxレジスタ ←I/Oパラメータの先頭オフセットアドレス

PC98

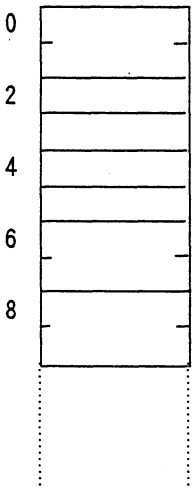


△［コマンドコード］&H1100

△［登録ボードNo.］0～3：リンクソフトで管理する登録ボードNo.を指定します。
△［使用セグメントアドレスNo.］0～42：リンクボードで設定した使用セグメントアドレスNo.。
△［割り込みNo.］0～6：リンクボードで設定した割り込みNo.を指定します。

設定値については、P.36～P.37をお読みください。

PC／AT

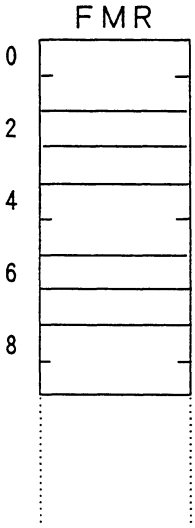


△［コマンドコード］&H1100

△［登録ボードNo.］0～3
△［使用セグメントアドレスNo.］0～31
△［割り込みNo.］3～15

設定値については、P.38～P.39をお読みください。

FMR



△［コマンドコード］&H1100

△［登録ボードNo.］0～3
△［基板I/Oポートアドレス No.］0～31：ロータリSW1～2
△［コントロール・ステータスI/Oポートアドレス］ロータリSW3～5

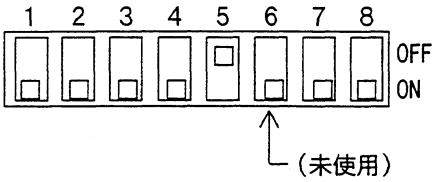
△［割り込みNo.］4～14

設定値については、P.40～P.41をお読みください。

補　足

ＰＣ９８シリーズ

■使用セグメントアドレスNo.の設定（デップSW1）



リンクボードの制御用エリアとして使用するＰＣ９８シリーズのメモリエリア（拡張ＲＯＭエリアのアドレス）を指定します。出荷状態では左図のように設定されています。

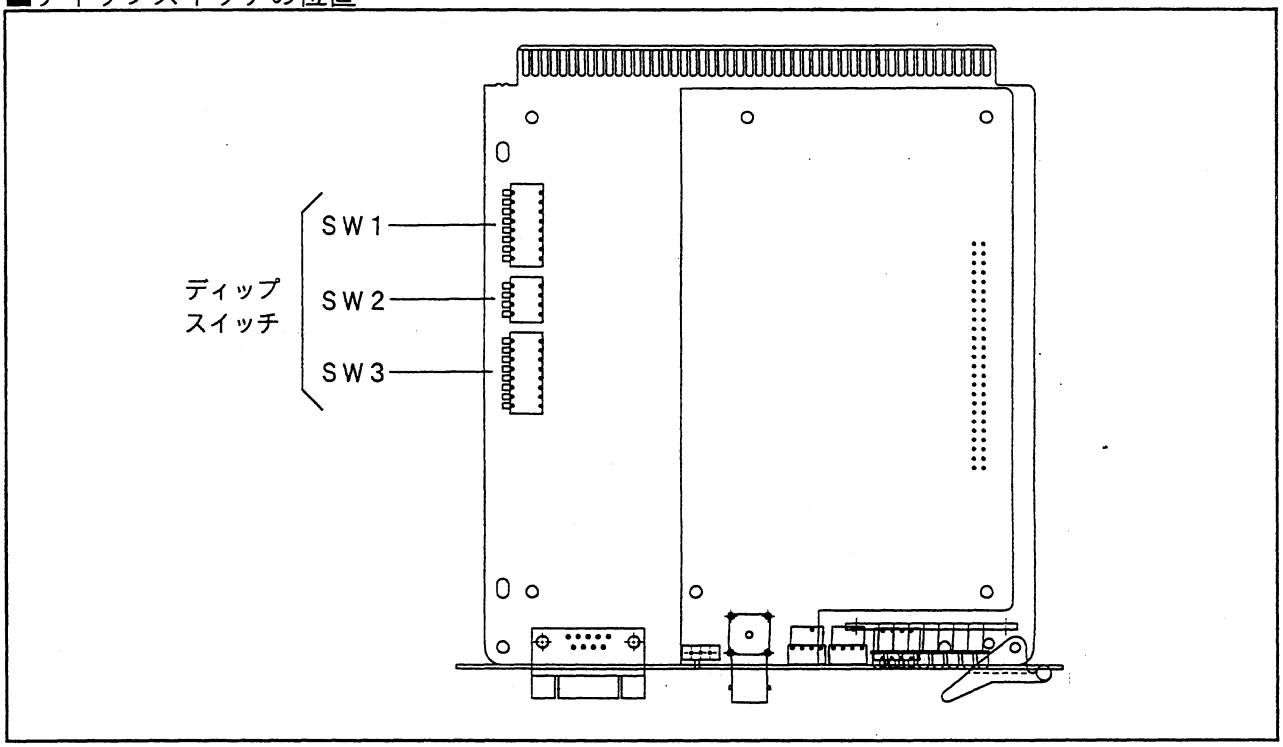
SW1	SW2	SW3	SW4	SW5	SW7	ユーザ使用メモリエリア	ｾｸﾞﾒﾝﾄｱﾄﾞﾚｽNo.
ON	ON	ON	ON	ON	ON	C0000～C0FFF	0
OFF	ON	ON	ON	ON	ON	C1000～C1FFF	1
ON	OFF	ON	ON	ON	ON	C2000～C2FFF	2
OFF	OFF	ON	ON	ON	ON	C3000～C3FFF	3
ON	ON	OFF	ON	ON	ON	C4000～C4FFF	4
OFF	ON	OFF	ON	ON	ON	C5000～C5FFF	5
ON	OFF	OFF	ON	ON	ON	C6000～C6FFF	6
OFF	OFF	OFF	ON	ON	ON	C7000～C7FFF	7
OFF	ON	OFF	ON	ON	OFF	E5000～E5FFF	8
ON	OFF	OFF	ON	ON	OFF	E6000～E6FFF	9
OFF	OFF	OFF	ON	ON	OFF	E7000～E7FFF	10
ON	ON	ON	OFF	ON	ON	C8000～C8FFF	11
OFF	ON	ON	OFF	ON	ON	C9000～C9FFF	12
ON	OFF	ON	OFF	ON	ON	CA000～CAFFF	13
OFF	OFF	ON	OFF	ON	ON	CB000～CBFFF	14
ON	ON	OFF	OFF	ON	ON	CC000～CCFFF	15
OFF	ON	OFF	OFF	ON	ON	CD000～CDFFF	16
ON	OFF	OFF	OFF	ON	ON	CE000～CEFFF	17
OFF	OFF	OFF	OFF	ON	ON	CF000～CFFFF	18

SW1	SW2	SW3	SW4	SW5	SW7	ユーザ使用メモリエリア	ｾｸﾞﾒﾝﾄｱﾄﾞﾚｽNo.
ON	ON	ON	OFF	ON	OFF	E8000～E8FFF	19
OFF	ON	ON	OFF	ON	OFF	E9000～E9FFF	20
ON	OFF	ON	OFF	ON	OFF	EA000～EAFFF	21
OFF	OFF	ON	OFF	ON	OFF	EB000～EBFFF	22
ON	ON	OFF	OFF	ON	OFF	EC000～ECFFF	23
OFF	ON	OFF	OFF	ON	OFF	ED000～EDFFF	24
ON	OFF	OFF	OFF	ON	OFF	EE000～EEFFF	25
OFF	OFF	OFF	OFF	ON	OFF	EF000～EFFFF	26
ON	ON	ON	ON	OFF	ON	D0000～D0FFF	27
OFF	ON	ON	ON	OFF	ON	D1000～D1FFF	28
ON	OFF	ON	ON	OFF	ON	D2000～D2FFF	29
OFF	OFF	ON	ON	OFF	ON	D3000～D3FFF	30
ON	ON	OFF	ON	OFF	ON	D4000～D4FFF	31
OFF	ON	OFF	ON	OFF	ON	D5000～D5FFF	32
ON	OFF	OFF	ON	OFF	ON	D6000～D6FFF	33
OFF	OFF	OFF	ON	OFF	ON	D7000～D7FFF	34
ON	ON	ON	OFF	OFF	ON	D8000～D8FFF	35
OFF	ON	ON	OFF	OFF	ON	D9000～D9FFF	36
ON	OFF	ON	OFF	OFF	ON	DA000～DAFFF	37
OFF	OFF	ON	OFF	OFF	ON	DB000～DBFFF	38
ON	ON	OFF	OFF	OFF	ON	DC000～DCFFF	39
OFF	ON	OFF	OFF	OFF	ON	DD000～DDFFF	40
ON	OFF	OFF	OFF	OFF	ON	DE000～DEFFF	41
OFF	OFF	OFF	OFF	OFF	ON	DF000～DFFFF	42

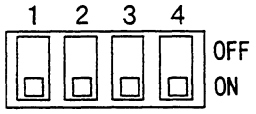
注　意

・1台のパソコンに複数のリンクボードを装着する場合は、使用メモリエリア（セグメントアドレスNo.）が重複しないように設定してください。
・▒▒は、ハイレゾモード用です。
・メモリエリアのアドレスは、システム予約、ハードディスクＢＩＯＳ、ＦＭ音源（サウンド）機能、ＥＭＳページフレーム、各種拡張機能ボードと重複しないように注意してください。
・SW8は、ON固定で使用してください。

■ディップスイッチの位置



■割り込みNo.の設定（ディップSW2）



リンクボード用の割り込みNo.（INT No.）を指定します。出荷状態ではすべてＯＮに設定されています。

SW1	SW2	SW3	割り込みNo.	他の用途	ノーマルモード	ハイレゾモード
ON	ON	ON	INT0		○	○
OFF	ON	ON	INT1		○	○
ON	OFF	ON	INT2	マウス機能(ハイレゾモード)	○	
OFF	OFF	ON	INT3	ＳＡＳＩ ＨＤＤ		
ON	ON	OFF	INT41(42)	内蔵ＦＤＤ		
OFF	ON	OFF	INT5	ＦＭ音源（サウンド）	△	△
ON	OFF	OFF	INT6	マウス機能(ノーマルモード)		

注意　○印の場合も、他の拡張ボード等と割り込みＮｏ．が重複することはできません。
　　△印は、他の用途を無効に設定すれば使用できます。

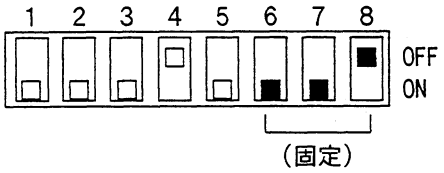
注　意

・1台のパソコンに複数のリンクボードを装着する場合は、すべてのリンクボードで割り込みNo.を同一に設定してください。

・割り込みNo.は、マウス、ＦＭ音源（サウンド）、ハードディスク、各種拡張ボードと重複しないように設定してください。

ＰＣ／ＡＴ互換機

■使用セグメントアドレスNo.の設定（デップSW1）



リンクボードの制御用エリアとして使用するＡＴ互換機のメモリエリア（拡張ＲＯＭエリアのアドレス）を指定します。出荷状態では左図のように設定されています。

SW1	SW2	SW3	SW4	SW5	ユーザ使用メモリエリア	ｾｸﾞﾒﾝﾄｱﾄﾞﾚｽNo.
ON	ON	ON	ON	ON	C0000～C0FFF	0
OFF	ON	ON	ON	ON	C1000～C1FFF	1
ON	OFF	ON	ON	ON	C2000～C2FFF	2
OFF	OFF	ON	ON	ON	C3000～C3FFF	3
ON	ON	OFF	ON	ON	C4000～C4FFF	4
OFF	ON	OFF	ON	ON	C5000～C5FFF	5
ON	OFF	OFF	ON	ON	C6000～C6FFF	6
OFF	OFF	OFF	ON	ON	C7000～C7FFF	7
ON	ON	ON	OFF	ON	C8000～C8FFF	8
OFF	ON	ON	OFF	ON	C9000～C9FFF	9
ON	OFF	ON	OFF	ON	CA000～CAFFF	10
OFF	OFF	ON	OFF	ON	CB000～CBFFF	11
ON	ON	OFF	OFF	ON	CC000～CCFFF	12
OFF	ON	OFF	OFF	ON	CD000～CDFFF	13
ON	OFF	OFF	OFF	ON	CE000～CEFFF	14
OFF	OFF	OFF	OFF	ON	CF000～CFFFF	15
ON	ON	ON	ON	OFF	D0000～D0FFF	16
OFF	ON	ON	ON	OFF	D1000～D1FFF	17
ON	OFF	ON	ON	OFF	D2000～D2FFF	18
OFF	OFF	ON	ON	OFF	D3000～D3FFF	19
ON	ON	OFF	ON	OFF	D4000～D4FFF	20
OFF	ON	OFF	ON	OFF	D5000～D5FFF	21
ON	OFF	OFF	ON	OFF	D6000～D6FFF	22
OFF	OFF	OFF	ON	OFF	D7000～D7FFF	23
ON	ON	ON	OFF	OFF	D8000～D8FFF	24
OFF	ON	ON	OFF	OFF	D9000～D9FFF	25
ON	OFF	ON	OFF	OFF	DA000～DAFFF	26
OFF	OFF	ON	OFF	OFF	DB000～DBFFF	27
ON	ON	OFF	OFF	OFF	DC000～DCFFF	28
OFF	ON	OFF	OFF	OFF	DD000～DDFFF	29
ON	OFF	OFF	OFF	OFF	DE000～DEFFF	30
OFF	OFF	OFF	OFF	OFF	DF000～DFFFF	31

ＶＧＡを使用する場合はこの領域は使用できません。

リンクボードの使用メモリには、この領域の使用をお奨めいたします。

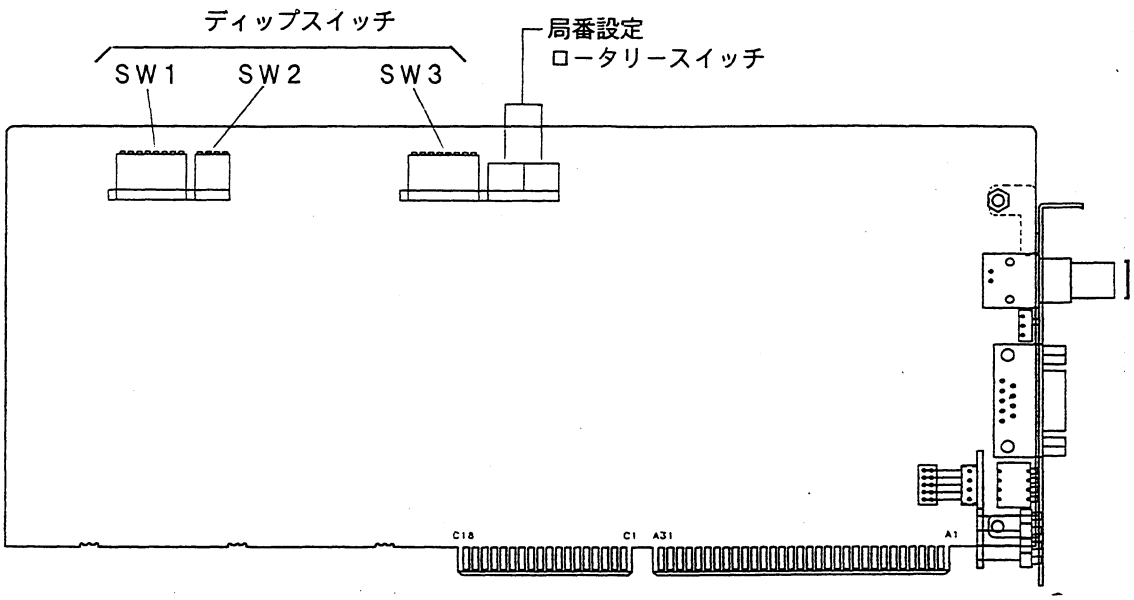
ＥＭＳを使用する場合はこの領域は使用できません。

注　意

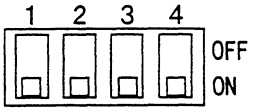
・1台のパソコンに複数のリンクボードを装着する場合は、使用メモリエリア（セグメントアドレスNo.）が重複しないように設定してください。

・メモリエリアのアドレスは、システム予約、ハードディスクＢＩＯＳ、ＦＭ音源（サウンド）機能、ＥＭＳページフレーム、各種拡張機能ボードと重複しないように注意してください。

■ディップスイッチの位置



■割り込みNo.の設定（ディップSW2）



リンクボード用の割り込みNo.（INT No.）を指定します。出荷状態ではすべてＯＮに設定されています。

SW1	SW2	SW3	SW4	割り込みＮｏ.	他の用途	使用の可否
ON	ON	ON	OFF	IRQ3	COM2/COM4	△
OFF	ON	ON	OFF	IRQ4	COM1/COM3	△
ON	OFF	ON	OFF	IRQ5	パラレルポート2	△
OFF	OFF	ON	OFF	IRQ6	フロッピーディスクドライブ	
ON	ON	OFF	OFF	IRQ7	パラレルポート	
OFF	ON	OFF	OFF	IRQ9	ＩＲＱ２関連	
ON	ON	ON	ON	IRQ10		○
OFF	ON	ON	ON	IRQ11		○
ON	OFF	ON	ON	IRQ12	PS/2マウス	
OFF	OFF	ON	ON	IRQ14	ハードディスク	
ON	ON	OFF	ON	IRQ15		○

注意　○印の場合も、他の拡張ボード等と割り込みＮｏ．が重複することはできません。
△印は、他の用途を無効に設定すれば使用できます。

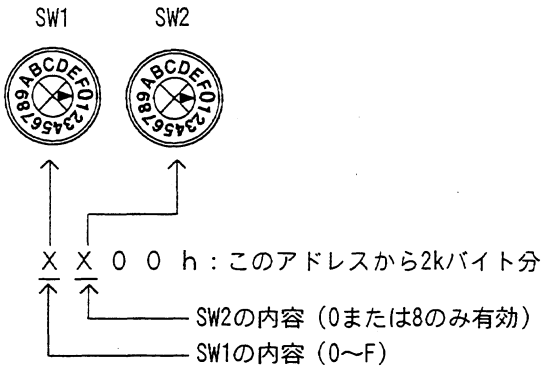
注　意

・1台のパソコンに複数のリンクボードを装着する場合は、すべてのリンクボードで割り込みNo.を同一に設定してください。

・割り込みNo.は、マウス、ＦＭ音源（サウンド）、ハードディスク、各種拡張ボードと重複しないように設定してください。

FMRシリーズ

■基板I/Oポートアドレス No.の設定（ロータリSW1･2）



ロータリスイッチSW1とSW2により、リンクボードの制御用に使用するFMRシリーズのI/Oポートアドレスの No.を指定します。出荷状態では、SW1は7、SW2は0に設定されています。

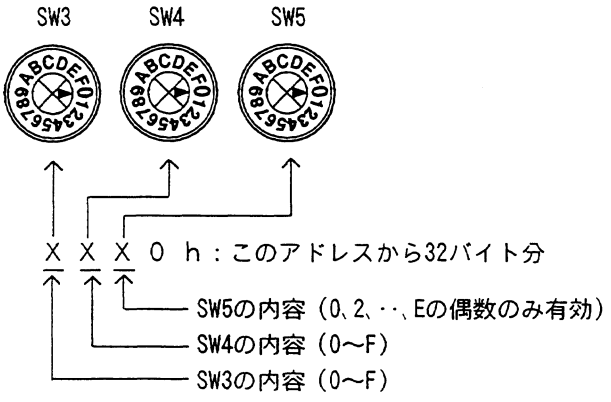
注　意

- 基板I/Oポートアドレスは、コントロール・ステータスポートアドレスと重複しないように設定してください。
- 1台のパソコンに複数のリンクボードを装着する場合は、I/Oポートアドレスが重複しないように設定してください。
- 下表の*マーク付きのアドレス No.は、使用しないでください、また、機種によっては、これ以外にも使用できないポートアドレスがありますので、詳しくはFMRのマニュアルをご覧ください。

SW1	SW2	使用I/Oポート	アドレスNo.
0	0	0000～07FF	0*
1	0	1000～17FF	1
2	0	2000～27FF	2
3	0	3000～37FF	3*
4	0	4000～47FF	4
5	0	5000～57FF	5
6	0	6000～67FF	6
7	0	7000～77FF	7
8	0	8000～87FF	8
9	0	9000～97FF	9
A	0	A000～A7FF	10
B	0	B000～B7FF	11
C	0	C000～C7FF	12
D	0	D000～D7FF	13
E	0	E000～E7FF	14
F	0	F000～F7FF	15*

SW1	SW2	使用I/Oポート	アドレスNo.
0	8	0800～0FFF	16*
1	8	1800～1FFF	17
2	8	2800～2FFF	18
3	8	3800～3FFF	19*
4	8	4800～4FFF	20
5	8	5800～5FFF	21
6	8	6800～6FFF	22
7	8	7800～7FFF	23
8	8	8800～8FFF	24
9	8	9800～9FFF	25
A	8	A800～AFFF	26
B	8	B800～BFFF	27
C	8	C800～CFFF	28
D	8	D800～DFFF	29
E	8	E800～EFFF	30
F	8	F800～FFFF	31*

■コントロール・ステータスI/Oポートアドレスの設定（ロータリSW3･4･5）

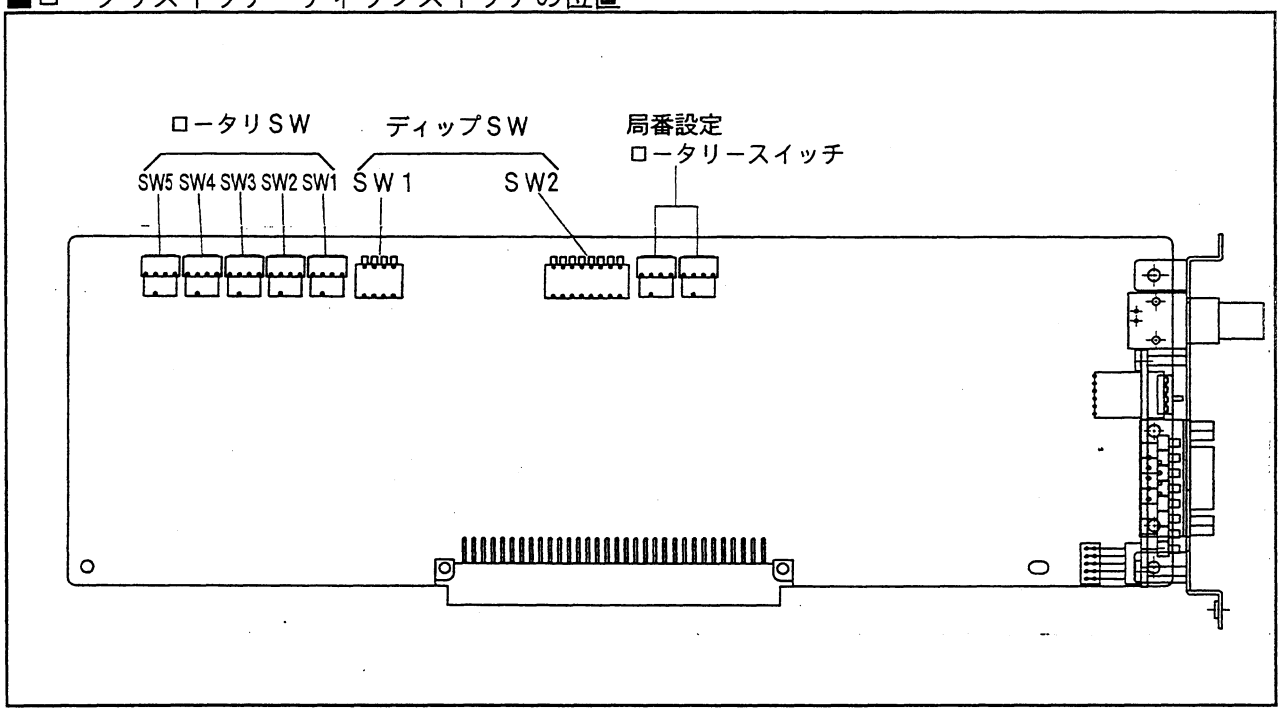


ロータリスイッチSW3、SW4、SW5により、リンクボードの通信用に使用するFMRシリーズのI/Oポートのアドレスを指定します。出荷状態では、SW3は7、SW4は8、SW5は0に設定されています。

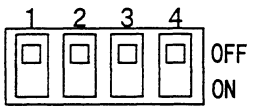
注　意

- コントロール・ステータスI/Oポートアドレスは、基板ポートアドレスと重複しないように設定してください。
- 1台のパソコンに複数のリンクボードを装着する場合は、I/Oポートアドレスが重複しないように設定してください。

■ロータリスイッチ・ディップスイッチの位置



■割り込みNo.の設定（ディップSW1）



リンクボード用の割り込みNo.（INT No.）を指定します。出荷状態ではすべてＯＦＦに設定されています。

SW1	SW2	SW3	SW4	割り込みＮｏ.	他の用途	使用の可否
OFF	OFF	OFF	ON	INT4		○
OFF	OFF	OFF	OFF	INT5		○
OFF	ON	ON	ON	INT10		○
OFF	ON	ON	OFF	INT11		○
OFF	ON	OFF	ON	INT13		○
OFF	ON	OFF	OFF	INT14		○

注意　○印の場合も、他の拡張ボード等と割り込みNo.が重複することはできません。

注　意

・１台のパソコンに複数のリンクボードを装着する場合は、すべてのリンクボードで同一の割り込みNo.を指定してください。

・割り込みNo.は、マウス、ＦＭ音源（サウンド）、ハードディスク、各種拡張ボードと重複しないように設定してください。

3-1-3 CH OPEN/CLOSE

通信開始手続き／通信チャネル使用許可／禁止（コマンドコード&H160x）

機　能

通信チャネルの使用を許可または禁止します。

解　説

・通信チャネルの使用を1CH単位で許可または禁止します。
・機能コマンドを実行するには、各種プログラミング言語でI/Oパラメータのセグメントアドレスとオフセットアドレスをds:dxレジスタにセットし、割り込みを実行するユーザープログラムを作成してください。
・コマンド実行後の結果（リターン値）は、axレジスタに格納されます。
　0：正常終了時
　その他：エラーコード（「3－9」参照）

■コマンドタイプ

コード	機能
&H1600	指定ＣＨ送受信禁止
&H1601	指定ＣＨ送受信許可
&H1602	指定ＣＨ現在状態読み出し

■ＣＨ　No.

No.	機能	初期値
0	コンピュータリンク用	許可
1	コンピュータ間通信用	禁止
2	データ転送用	禁止

注　意

・CH No.には0～2以外は指定しないでください。
・&H1602ではCH No.はセット不要です。

文　例

●MS-DOSファンクションコール（int21h）

```
;------- CH 許可／禁止 ------
CHOP:
        mov     bx,[FILHDL]
        mov     ah,44h
        mov     al,3
        mov     dx,offset CHOP
        int     21h
;------- I/Oﾊﾟﾗﾒｰﾀ ------
CHOP    dw      1601h
        db      0,2
CHST    db      8 dup(?)
        dw      18 dup(?)
```

●ソフトウェア割り込み（int60h）

```
;------- CH 許可／禁止 ------
CHOP:
        mov     dx,offset CHOP
        int     60h
;------- I/Oﾊﾟﾗﾒｰﾀ ------
CHOP    dw      1601h
        db      0,2
CHST    db      8 dup(?)
        dw      18 dup(?)
```

I/Oパラメータ

△：設定する［渡す値］
▼：実行後に格納される［得る値］

dsレジスタ ←I/Oパラメータの先頭セグメントアドレス
dxレジスタ ←I/Oパラメータの先頭オフセットアドレス

0	△［コマンドコード］&H160x
2	△［登録ボードNo.］0～3（リンクソフトで管理するボードNo.）
4	△［ＣＨ　No.］0～2（許可／禁止するチャンネル番号）
6	▼［許可／禁止フラグ］コマンド実行後、ＣＨ許可／禁止状態が格納されます（下表参照）。
8	
A	
C	
E	
10	
12	
14	
16	
18	
1A	
1C	
1E	
20	
22	
24	
26	
28	
2A	
2C	
2E	

■許可／禁止フラグのＣＨ位置

オフセットアドレス ＼ ビット	7	6	5	4	3	2	1	0
						CH 2	CH 1	CH 0
4	7	6	5	4	3	2	1	0
5	15	14	13	12	11	10	9	8
6	23	22	21	20	19	18	17	16
7	31	30	29	28	27	26	25	24
8	39	38	37	36	35	34	33	32
9	47	46	45	44	43	42	41	40
A	55	54	53	52	51	50	49	48
B	63	63	61	60	59	58	57	56

＊CH No.0～2以外はシステム予約です。

＜初期値＞

	7	6	5	4	3	2	1	0
オフセット4	0	0	0	0	0	0	0	1

CH0のみ許可となっています

3-1-4 通信開始手続きのプログラム例

アセンブラプログラム（MS-DOSシステムコールint21）

```
; ********************************************************
; *              プログラム例：通信開始手続き            *
; *                    <AC0.ASM>                         *
; ********************************************************
; *                                  PC98シリーズ        *
; *                                                      *
; *                                                      *
; ********************************************************
                                        ;
CODE    SEGMENT
        ASSUME CS:CODE,DS:CODE,ES:CODE,SS:CODE


CR              EQU     0DH             ; CRｺｰﾄﾞ
LF              EQU     0AH             ; LFｺｰﾄﾞ
                                        ;
                                        ;
START:
                MOV     AX,CS
                MOV     DS,AX
                MOV     ES,AX
                MOV     SS,AX
                MOV     AX,OFFSET STPT
                MOV     SP,AX
                                        ;
; ----------------------------------------
;               初 期 設 定
; ----------------------------------------
INI:
                MOV     DX,OFFSET MSG1  ; 初期設定メッセージ表示
                CALL    MESSAGE
                MOV     DX,OFFSET INIT  ; 初期設定用I/Oﾊﾟﾗﾒｰﾀ ｱﾄﾞﾚｽ設定
                CALL    COMMAND         ; ﾘﾝｸﾘｿﾌﾄ呼出
                JNB     BDST
                JMP     DOS             ; ｴﾗｰ時 MS-DOSへ戻る
                                        ;
; ----------------------------------------
;       ボ ー ド 登 録 ・ 起 動
; ----------------------------------------
                                        ;
BDST:
                MOV     DX,OFFSET MSG4  ; ﾎﾞｰﾄﾞ登録･起動メッセージ表示
                CALL    MESSAGE
                MOV     DX,OFFSET BOARD         ; ﾎﾞｰﾄﾞ登録用I/Oﾊﾟﾗﾒｰﾀ ｱﾄﾞﾚｽ
                CALL    COMMAND         ; ﾘﾝｸﾘｿﾌﾄ呼出
                JNB     CH20P
                JMP     DOS             ; ｴﾗｰ時 MS-DOSへ戻る
                                        ;
; ----------------------------------------
;             CH－2 OPEN
; ----------------------------------------
                                        ;
CH20P:
                MOV     DX,OFFSET MSG5  ; CH－2 OPENメッセージ
                CALL    MESSAGE
                MOV     DX,OFFSET CH0P  ; CH OPEN I/Oﾊﾟﾗﾒｰﾀｱﾄﾞﾚｽ
                CALL    COMMAND
                JNB     LP
                JMP     DOS             ; ｴﾗｰ時 MS-DOSへ戻る
                                        ;
; ========================================
; =          MS－DOSへ戻る処理          =
; ========================================
                                        ;
DOS:
                MOV     AH,4CH
                INT     21H
                                        ;
                                        ;
```

```
; ==========================================
; =          メッセージの画面表示処理          =
; ==========================================
                                        ;
                MOV     AH,9            ; DXｱﾄﾞﾚｽから'$'迄を画面表示
                INT     21H
                RET
                                        ;
                                        ;
; ==========================================
; =          リンクソフト呼出処理            =
; ==========================================
                                        ;
COMMAND:
                INT     60H             ; ｼｽﾃﾑｺｰﾙ
                CMP     AX,0            ; AX>0時ｴﾗｰ
                JE      CMD2            ;
                                        ;
                CALL    BI2ASC          ; ｴﾗｰｺｰﾄﾞ ASCII変換
                MOV     DX,OFFSET MSG3  ; ｴﾗｰﾒｯｾｰｼﾞ画面表示
                CALL    MESSAGE
                STC
CMD2:
                RET
                                        ;
                                        ;
; ==========================================
; =       エラーコードのＡＳＣＩＩ変換       =
; ==========================================
                                        ;
BI2ASC:
                MOV     BL,AH
                MOV     AH,AL
                MOV     BH,BL
                AND     AL,0FH
                AND     BL,0FH
                MOV     CL,4
                SHR     AH,CL
                SHR     BH,CL
                ADD     BX,3030H
                ADD     AX,3030H
                CMP     AL,3AH
                JB      B0
                ADD     AL,7
B0:
                CMP     AH,3AH
                JB      B1
                ADD     AH,7
B1:
                CMP     BL,3AH
                JB      B2
                ADD     BL,7
B2:
                CMP     BH,3AH
                JB      B3
                ADD     BH,7
B3:
                MOV     [MSG3_1+0],BH
                MOV     [MSG3_1+1],BL
                MOV     [MSG3_1+2],AH
                MOV     [MSG3_1+3],AL
                RET
                                        ;
                                        ;
; ==========================================
; =            Ｉ／Ｏパラメータ              =
; ==========================================
                                        ; = 初期設定 =
INIT            DW      1000H           ; 初期設定ｺﾏﾝﾄﾞｺｰﾄﾞ
                DB      0               ; ｱﾌﾟﾘｹｰｼｮﾝ･ｿﾌﾄﾓｰﾄﾞ
                DB      1               ; 使用ﾎﾞｰﾄﾞ枚数
                                        ;
                                        ; = ﾎﾞｰﾄﾞ登録･起動 =
BOARD           DW      1100H           ; ﾎﾞｰﾄﾞ登録･起動ｺﾏﾝﾄﾞｺｰﾄﾞ
                DB      0               ; ﾎﾞｰﾄﾞ登録No.
                DB      27              ; 使用ｾｸﾞﾒﾝﾄｱﾄﾞﾚｽNo.27(D0000H)
                DB      0               ; 割り込みNo.
                DB      0
```

パソコン機種別変更箇所
（左記はＰＣ９８シリーズの場合）

```
                                        ; = CH OPEN =
CHOP        DW      1601H               ; ｺﾏﾝﾄﾞｺｰﾄﾞ
            DB      0                   ; 登録ﾎﾞｰﾄﾞNO.
            DB      2                   ; OPEN CH NO.
            DB      8 DUP(0)            ; CH状態格納用
                                        ;
                                        ;
                                        ;
; ========================================
; =        メッセージ ／ バッファ領域        =
; ========================================
                                        ;
MSG1        DB      '初期設定：ボード１枚使用指定を行います'
            DB      CR,LF,'$'

MSG2        DB      'ドライバーが組み込まれていません'
            DB      CR,LF,'$'

MSG3        DB      'エラーが発生しました  ｴﾗｰｺｰﾄﾞ===>'
MSG3_1      DB      0,0,0,0
            DB      CR,LF,'$'

MSG4        DB      '使用ボードNo.０  セグメントNo.２７',CR,LF
            DB      '割り込みレベル０を登録します'
            DB      CR,LF,'$'

MSG5        DB       'ＣＨ２ ＯＰＥＮ===>$'
            DB      CR,LF,'$'

STACK       DW      256 DUP(0)                  ; ｽﾀｯｸ領域
STPT        DB      0


            CODE    ENDS
            END     START
```

パソコン機種別変更箇所

ＰＣ／ＡＴ互換機

```
BOARD       DW      1100H               ; ﾎﾞｰﾄﾞ登録･起動ｺﾏﾝﾄﾞｺｰﾄﾞ
            DB      0                   ; ﾎﾞｰﾄﾞ登録No.
            DB      8                   ; 使用ｾｸﾞﾒﾝﾄｱﾄﾞﾚｽNo.
            DB      10                  ; 割り込みﾚﾍﾞﾙNo.
            DB      0
```

ＦＭＲシリーズ

```
BOARD       DW      1100H               ; ﾎﾞｰﾄﾞ登録･起動ｺﾏﾝﾄﾞｺｰﾄﾞ
            DB      0                   ; ﾎﾞｰﾄﾞ登録No.
            DB      7                   ; 基板I/OﾎﾟｰﾄｱﾄﾞﾚｽNo.
            DW      7800H               ; ｺﾝﾄﾛｰﾙ・ｽﾃｰﾀｽI/Oﾎﾟｰﾄｱﾄﾞﾚｽ
            DB      5                   ; 割り込みﾚﾍﾞﾙNo.
            DB      0
```

Cプログラム（ソフトウェア割り込みint60）

```
/*************************************************************/
/*                  プログラム例：通信手続き開始             */
/*                          (CO.C)                           */
/*************************************************************/
/*                    ＰＣ９８シリーズ                       */
/*                                                           */
/*                                                           */
/*************************************************************/

#include  <dos.h>
#include  <stdio.h>
#include  <stdlib.h>

static union REGS inregs;
static union REGS outregs;
struct SREGS segregs;
static char far *p1;
static unsigned char dat[48];

void main( void )
{
        int     err;

/* -------------------------*/
/* ------ 初期設定処理 ------*/
/* -------------------------*/

        dat[0] = 0x00;          /* 初期設定ｺﾏﾝﾄﾞ･ｺｰﾄﾞ=1000H  */
        dat[1] = 0x10;
        dat[2] = 0;             /* ｱﾌﾟﾘｹｰｼｮﾝ･ｿﾌﾄﾓｰﾄﾞ=0        */
        dat[3] = 1;             /* 使用ボード枚数=1           */

        printf("初期設定===>");

        if((err = mew_send()) != 0)
        {
                printf("ｴﾗｰ.....ｺｰﾄﾞ= %x¥n", err);
                exit(-1);
        }
        printf("ＯＫ！¥n");

/* ---------------------------------*/
/* ------ 使用ボード登録 及び起動 ------*/
/* ---------------------------------*/

        dat[0] = 0x00;          /* 使用ボード登録及び起動          */
        dat[1] = 0x11;          /*ｺﾏﾝﾄﾞ･ｺｰﾄﾞ=1100H
        dat[2] = 0x00;          /* 登録ボードNo.=0                */
        dat[3] = 27;            /* 使用ｾｸﾞﾒﾝﾄｱﾄﾞﾚｽNo=27           */
        dat[4] = 0;             /* 割り込みNo.=0                  */

        printf("使用ボード登録 & 起動===>");

        if((err = mew_send()) != 0)
        {
                printf("ｴﾗｰ.....ｺｰﾄﾞ= %x¥n", err);
                exit(-1);
        }
        printf("ＯＫ！¥n");

/* ---------------------------*/
/* ------ ＣＨ２ ＯＰＥＮ ------*/
/* ---------------------------*/

        dat[0] = 0x01;          /* CH OPEN ｺﾏﾝﾄﾞ･ｺｰﾄﾞ=1601H  */
        dat[1] = 0x16;
        dat[2] = 0x00;          /* 登録ﾎﾞｰﾄﾞNo.=0             */
        dat[3] = 0x02;          /* OPEN CH No. =2             */

        printf("ＣＨ２ ＯＰＥＮ===>");
```

パソコン機種別変更箇所
（左記はＰＣ９８シリーズの場合）

```
        if((err = mew_send()) != 0)
        {
                printf("ｴﾗｰ...ｺｰﾄﾞ= %x¥n",err);
                exit(-1);
        }
        printf("ＯＫ！¥n");

/* --------------------------*/
/* ------ ﾘﾝｸｿﾌﾄ呼び出し------*/
/* --------------------------*/
int mew_send(void)
{

        p1 = (char far *)&dat[0];

        inregs.x.dx = FP_OFF(p1);
        segregs.ds = FP_SEG(p1);

        int86x( 0x60, &inregs, &outregs, &segregs);
        return(outregs.x.ax);
}
```

パソコン機種別変更箇所

ＰＣ／ＡＴ互換機

```
        dat[0] = 0x00;          /* 使用ボード登録及び起動    */
        dat[1] = 0x11;          /* ｺﾏﾝﾄﾞｺｰﾄﾞ=1100H           */
        dat[2] = 0x00;          /* 登録ボードNo.=0           */
        dat[3] = 8;             /* 使用ｾｸﾞﾒﾝﾄｱﾄﾞﾚｽNo.=8      */
        dat[4] = 10;            /* 割り込みNo.=10            */
```

ＦＭＲシリーズ

```
        dat[0] = 0x00;          /* 使用ボード登録及び起動    */
        dat[1] = 0x11;          /* ｺﾏﾝﾄﾞｺｰﾄﾞ=1100H           */
        dat[2] = 0x00;          /* 登録ボードNo.=0           */
        dat[3] = 7;             /* 基板I/OﾎﾟｰﾄｱﾄﾞﾚｽNo.=7     */
        dat[4] = 0x00;          /* ｽﾃｰﾀｽ･ｺﾝﾄﾛｰﾙI/Oﾎﾟｰﾄｱﾄﾞﾚｽ */
        dat[5] = 0x78;          /* =7800H                    */
        dat[6] = 5;             /* 割り込みNo.=5             */
```

3－2 ＰＣリンクの機能コマンド

3-2-1 PC LINK OPEN/CLOSE

ＰＣリング通信／ＰＣリンク使用許可／禁止（コマンドコード&H161x）

機　能

ＰＣリンク機能の使用許可（OPEN）または使用禁止（CLOSE）を指示します。

解　説

・リンクボードはＰＣリンク起動状態になっている時、リンクソフトから使用許可（ＯＰＥＮ）を指示することによりＰＣ ＬＩＮＫ機能が使用可能になります。また、リンクソフトから使用禁止（ＣＬＯＳＥ）を指示すると、ＰＣリンク機能は使用できません。
・機能コマンドを実行するには、各種プログラミング言語でI/Oパラメータのセグメントアドレスとオフセットアドレスをds:dxレジスタにセットし、割り込みを実行するユーザープログラムを作成してください。
・コマンド実行後の結果（リターン値）は、axレジスタに格納されます。
　0：正常終了時
　その他：エラーコード（「３－９」参照）

■コマンドタイプ

コード	機能
&H1610	ＰＣリンク使用禁止
&H1611	ＰＣリンク使用許可

注　意

・PC LINK WRITE/READコマンド等でもOPEN可能ですが、WRITE/READ前にリンクボードのＰＣリンク機能を動作させておきたい場合は、使用許可のコマンドを使用します。
・リンクボード起動後の初期値は、使用禁止（CLOSE）です。

文　例

●MS-DOSファンクションコール（int21h）

```
;------- PCﾘﾝｸ許可／禁止 ------
PCLNK:
        mov     bx, [FILHDL]
        mov     ah, 44h
        mov     al, 3
        mov     dx, offset  PCLNK
        int     21h
;------- I/Oﾊﾟﾗﾒｰﾀ ------
PCLNK   dw      1611h
        db      0, 0
        dw      22 dup(?)
```

●ソフトウェア割り込み（int60h）

```
;------- PCﾘﾝｸ許可／禁止 ------
PCLNK:
        mov     dx, offset  PCLNK
        int     60h
;------- I/Oﾊﾟﾗﾒｰﾀ ------
PCLNK   dw      1611h
        db      0, 0
        dw      22 dup(?)
```

I/Oパラメータ

△：設定する［渡す値］
▼：実行後に格納される［得る値］

dsレジスタ ←I/Oパラメータの先頭セグメントアドレス
dxレジスタ ←I/Oパラメータの先頭オフセットアドレス

0	△［コマンドコード］&H161x
2	△［登録ボードNo.］0～3（リンクソフトで管理するボードNo.）
4	△ 設定値：00固定で使用してください。
6	
8	
A	
C	
E	
10	
12	
14	
16	
18	
1A	
1C	
1E	
20	
22	
24	
26	
28	
2A	
2C	
2E	

■ＰＣリンクの概要

ＭＥＷＮＥＴ－Ｈリンクボードでは、リンクリレー用に512バイト、データリンク用に4096ワード分のＰＣ　ＬＩＮＫ用の領域をもっています。その領域内での、各局のＰＣＬＩＮＫ送受信範囲の設定は、別途設定ソフトにて行い、リンクボードに登録されます。

リンクボードは、リンクソフトからの受信要求時に、その設定された範囲で、データの受信を行います。
読み出しポインタとは、リンクボード内部ＰＣＬＩＮＫ領域のポインタ値で、別途設定ソフトにて設定したレジスタNo.とは異なります。

例：下記のように設定ソフトでリンクリレーの設定を行った場合。

設定内容・・・・ WL10～40ワード分をリンクリレー領域として
No.1=WL10～10ワードを送信領域
No.2=WL20～10ワードを送信領域
No.3=WL30～10ワードを送信領域
No.4=WL40～10ワードを送信領域

レジスタＮｏ．では．．．

	No.1用	No.2用	No.3用	No.4用
WL10	▒	WL10	WL10	WL10
WL20	WL20	▒	WL20	WL20
WL30	WL30	WL30	▒	WL30
WL40	WL40	WL40	WL40	▒

▒は送信領域です。

リンク内部領域ポインタでは．．．

	No.1用	No.2用	No.3用	No.4用
0	▒	0	0	0
20	20	▒	20	20
40	40	40	▒	40
→60	60	60	60	▒

▒は送信領域です。
└ポインタ値

3-2-2 ＰＣリンクパラメータ読み出し

ＰＣリンク通信／ＰＣリンクパラメータ読み出し（コマンドコード&H1900）

機　能

ＰＣリンクパラメータを指定したバッファに読み出します。

解　説

・ＰＣリンク送信領域（リレーリンク領域、データリンク領域）の書き込みポインタ、書き込みサイズ等の情報を指定したバッファに読み出します。
・機能コマンドを実行するには、各種プログラミング言語でI/Oパラメータのセグメントアドレスとオフセットアドレスをds:dxレジスタにセットし、割り込みを実行するユーザープログラムを作成してください。
・コマンド実行後の結果（リターン値）は、axレジスタに格納されます。
　0：正常終了時
　その他：エラーコード（「３－９」参照）

注　意

・各局の送信サイズには、バイト数が格納されます。PC LINK WRITE等で使用する時は、ワード数に変換してご使用下さい。
・パラメータがリンクボードに設定されていない場合、140Hエラーとなり、読み出しは行われません。また、読み出したパラメータにエラーが存在する場合も、140Hエラーが返ります（読み出しは行います）。

文　例

●MS-DOSファンクションコール（int21h）

```
;------- PCﾘﾝｸﾊﾟﾗﾒｰﾀ読み出し ------
PCPRM:
        mov     bx,[FILHDL]
        mov     ah,44h
        mov     al,3
        mov     dx,offset PCPRM
        int     21h
;------- I/Oﾊﾟﾗﾒｰﾀ ------
PCPRM   dw      1190h
        db      0,?
        dw      8,12,
                offset PRBUF,seg PRBUF,12,17 dup(?)
PRBUF   311 dup(?)
```

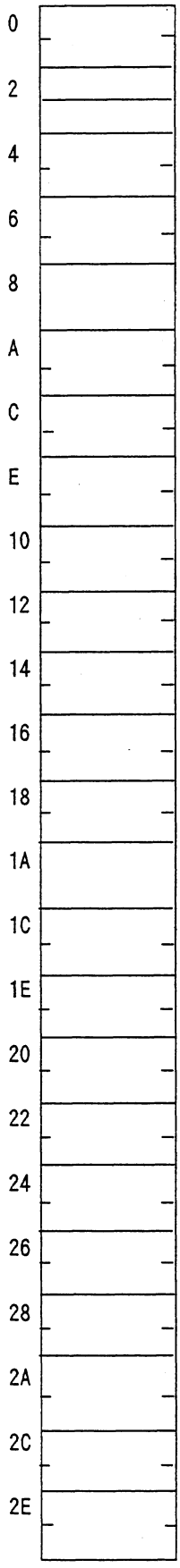
●ソフトウェア割り込み（int60h）

```
;------- PCﾘﾝｸﾊﾟﾗﾒｰﾀ読み出し ------
PCPRM:
        mov     dx,offset PCPRM
        int     60h
;------- I/Oﾊﾟﾗﾒｰﾀ ------
PCPRM   dw      1190h
        db      0,?
        dw      8,12,
                obset PRBUF,0,12,17 dup(?)
PRBUF   311 dup(?)
```

I/Oパラメータ

△：設定する［渡す値］
▼：実行後に格納される［得る値］

dsレジスタ ←I/Oパラメータの先頭セグメントアドレス
dxレジスタ ←I/Oパラメータの先頭オフセットアドレス



△［コマンドコード］&H1900

△［登録ボードNo.］0～3（使用するリンクボードの登録ボードNo.）
▼［自局No.］コマンド実行後、自局の局番（ボードのロータリスイッチで設定）が格納されます。
△［読み出し開始要素No.］ＰＣリンクパラメータの読み出し開始要素Ｎｏ．を指定します。

△［読み出し要素数］ＰＣリンクパラメータの読み出し要素数を指定します。

△［パラメータバッファアドレス（オフセット）］
読み出しデータを格納するアドレス（オフセット）を設定します。
△［パラメータバッファアドレス（セグメント）］
読み出しデータを格納するアドレス（セグメント）を設定します。
△［パラメータバッファサイズ］読み出しデータを格納するエリアのサイズを設定します。

■ＰＣリンクパラメータ

要素No.	名称	内容
0		
1		
2 ～ 7		
8	フラグ領域コード	0=WL, 1=WR, 2=WY, 3=WX, 4=SV, 5=EV, 6=LD, 7=SWR, 8=SWD, 9=DT, A=FL
9	フラグ領域先頭ワードNo.	
10	フラグ領域ワード数	
11	エラーレジスタ領域コード	0=WL, 1=WR, 2=WY, 3=WX, 4=SV, 5=EV, 6=LD, 7=SWR, 8=SWD, 9=DT, A=FL
12	エラーレジスタ領域先頭ワードNo.	
13	エラーレジスタ領域ワード数	
14	リレーリンク領域コード	0=WL, 1=WR, 2=WY, 3=WX, 4=SV, 5=EV, 6=LD, 7=SWR, 8=SWD, 9=DT, A=FL
15	リレーリンク領域先頭ワードNo.	
16	リレーリンク領域ワード数	
17	データリンク領域コード	0=WL, 1=WR, 2=WY, 3=WX, 4=SV, 5=EV, 6=LD, 7=SWR, 8=SWD, 9=DT, A=FL
18	データリンク領域先頭ワードNo.	
19	データリンク領域ワード数	
20	ユニットNo. 1用　リレーリンク 送信領域ポインタ	0FFFFH：未使用局　その他：ポインター値 リンクリレ／データーワード数が0の時はステータスのみ送信
21	リレーリンク 送信バイト数	上位ビットＯＮ時は送信できません
22	データリンク 送信領域ポインタ	
23	データリンク 送信バイト数	上位ビットＯＮ時は送信できません
24 ～ 275	ユニットNo. 2～64用	要素No. 20～23と同様に4要素で1局分
276	保証リレー断時リレーリンクエリア指定	0：クリアしない　1：クリアする
277	保証リレー断時データリンクエリア指定	0：クリアしない　1：クリアする
278	リレーリンク受信重複チェック指定	0：チェックしない　1：チェックする
279	データリンク受信重複チェック	0：チェックしない　1：チェックする
280 ～ 311		

3-2-3 PC LINK WRITE

ＰＣリンク通信／ＰＣリンク書き込み（コマンドコード&H154x）

機　能

指定されたバッファに格納されているＰＣリンクデータをリンクボード内ＰＣリンク領域に書き込み、送信要求を行います。

解　説

・実行するとＰＣリンク用ＣＨの使用を自動的に許可し、PC LINK OPEN/CLOSEコマンドで使用を禁止するまで、許可状態を保持します。
・機能コマンドを実行するには、各種プログラミング言語でI/Oパラメータのセグメントアドレスとオフセットアドレスをds:dxレジスタにセットし、割り込みを実行するユーザープログラムを作成してください。
・コマンド実行後の結果（リターン値）は、axレジスタに格納されます。
　0：正常終了時
　その他：エラーコード（「３－９」参照）

■コマンドタイプ

コード	機能
&H1540	送信完了待ち（PC98のみKEY Break付き）
&H1541	送信完了待ち（タイムアウト付き）
&H1542	送信要求のみ
&H1543	送信完了センス（&H1542完了検知用）
&H1544	データ書き込み（送信要求無し）
&H1545	ＰＣリンク送信停止

注　意

・&H1540のKEY Breakは、ESCキーのみです。
・&H1540・&H1541・&H1542はPC LINK OPEN機能付きです。
・&H1543では、I/Oパラメータの4H～15Hまでの設定は不要です。
・&H1554は、データのみ更新し、送信要求はしません。
・タイムアウト値は、コントロールレジスタWRITEコマンドにて変更できます。
・ランダムWRITE機能起動中は、本コマンドはすべて使用できません。

文　例

●MS-DOSファンクションコール（int21h）

```
;----- PCﾘﾝｸWRITE ----
PCWT:
        mov     bx,[FILHDL]
        mov     ah,44h
        mov     al,3
        mov     dx,offset PCWT
        int     21h
;------- I/Oﾊﾟﾗﾒｰﾀ ------
PCWT    dw      1541h
        db      0,5h
        dw      offset LRBUF,seg LRBUF,?,0,10,
                offset LDBUF,seg LDBUF,?,0,10
        db      1,1
        dw      11 dup(?)
LRBUF   dw      256 dup(0)
LDBUF   dw      4096 dup(0)
```

●ソフトウェア割り込み（int60h）

```
;----- PCﾘﾝｸWRITE ----
PCWT:
        mov     dx,offset PCWT
        int     60h
;------- I/Oﾊﾟﾗﾒｰﾀ ------
PCWT    dw      1541h
        db      0,5h
        dw      ofset LRBUF,seg LRBUF,?,0,10,
                ofset LDBUF,seg LDBUF,?,0,10
        db      1,1
        dw      11 dup(?)
LRBUF   dw      256 dup(0)
LDBUF   dw      4096 dup(0)
```

I/Oパラメータ

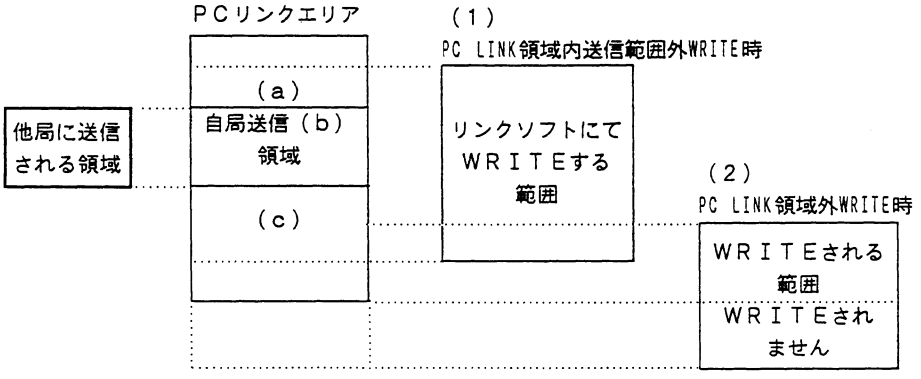
△：設定する［渡す値］
▼：実行後に格納される［得る値］

dsﾚｼﾞｽﾀ ←I/Oパラメータの先頭セグメントアドレス
dxﾚｼﾞｽﾀ ←I/Oパラメータの先頭オフセットアドレス

0	△［コマンドコード］&H154x
2	△［登録ボードNo.］0～3　使用するリンクボードの登録ボードNo.を設定します。
	△ 設定値：05固定で使用してください。
4	△［ＬＲバッファアドレス（オフセット）］　＊ＬＲバッファは最大256ワードまでです。
6	△［ＬＲバッファアドレス（セグメント）］
8	
A	△［ＬＲ書き込みポインタ］
	リンクボード内部リンクリレー領域の先頭アドレスからのオフセット。
C	△［ＬＲ書き込みサイズ］ワード単位
	上記ポインタから書き込むサイズを指定します。
E	△［ＬＤバッファ（オフセット）］　＊ＬＤバッファは最大4096ワードまでです。
10	△［ＬＤバッファ（セグメント）］
12	
14	△［ＬＤ書き込みポインタ］
	リンクボード内部リンクデータ領域の先頭アドレスからのオフセット。
16	△［ＬＤ書き込みサイズ］ワード単位
	上記ポインタから書き込むサイズを指定します。
18	△［動作モードステータス］0：ＰＲＯＧ／1：ＲＵＮ　他局でのモニタ用にセット。
	△［運転状態ステータス］0：異常／1：正常　他局でのモニタ用にセット。
1A	
1C	
1E	
20	
22	
24	
26	
28	
2A	
2C	
2E	

補　足

・一旦送信要求を行うと、次の送信要求または送信停止指示（&H1545）まで、同一データを送信し続けます。（リンクボードがPC LINKパラメータの変更等でPC LINK停止になった時、送信は停止します。）
・リンクボード内に、PC LINK用パラメータが登録されていない時、PC LINKの送信は行えません。
・書き込み範囲が自局の送信領域(b)外である時、リンクボード内のリンクエリアに対するデータ書き込みは行いますが、送信は送信範囲内だけとなります。（ただし、(a)(c)が他局の送信エリアに設定されている場合は、他局の送信データにより更新されてしまいます。）
・下図（１）の場合、141Hエラーが返ります（PC LINK送信範囲外設定エラー）。
・下図（２）の場合、143Hエラーが返ります（PC LINKリンク領域範囲外設定エラー）。



3-2-4 PC LINKランダムWRITE

ＰＣリンク通信／ＰＣリンク書き込み・登録受信タイプ（コマンドコード&H155x）

機　能

指定された送信バッファのデータを指定タイミングで送信します。

解　説

・指定した送信バッファ内にデータをセットするだけで、リンクソフトが自動的に送信します。
・機能コマンドを実行するには、各種プログラミング言語でI/Oパラメータのセグメントアドレスとオフセットアドレスをds:dxレジスタにセットし、割り込みを実行するユーザープログラムを作成してください。
・コマンド実行後の結果（リターン値）は、axレジスタに格納されます。
　0：正常終了時
　その他：エラーコード（「３－９」参照）

■コマンドタイプ

コード	機能
&H1550	PC LINKランダムWRITE停止
&H1551	PC LINKランダムWRITE起動

注　意

・&H1550はI/Oパラメータの4h以降の指定は不要です。
・&H1551はPC LINK OPEN機能付きです。
・ランダムWRITE起動中は、各設定内容の変更はできません（無視されます）。設定内容の変更は、ランダムREADを一旦停止した後に行ってください。
・リンクボード内のリンク領域の初期化には、PC LINK WRITEコマンドを使用してください。
・ランダムWRITE機能起動中は、他のPC LINK WRITEコマンド（&H1540～&H1545）はすべて使用できません。

文　例

●MS-DOSファンクションコール（int21h）

```
;----- PCﾘﾝｸﾗﾝﾀﾞﾑWRITE ----
PCRWT:
        mov     bx, [FILHDL]
        mov     ah, 44h
        mov     al, 03h
        mov     dx, offset PCRWT
        int     21h
;------- I/Oﾊﾟﾗﾒｰﾀ ------
PCRWT   dw      1551h
        db      0, 5h
        dw      offset LRBUF, seg LRBUF, ?, 0, 100h,
                offset LDBUF, seg LDBUF, ?, 0, 100h, ?, ?,
                offset STB, seg STB, offset STC, seg STC
        db      255, 9 dup(?)
SND     dw      0
LRBUF   dw      256 dub(0)
LDBUF   dw      4096 dub(0)
STB     db      1
STC     db      1
```

●ソフトウェア割り込み（int60h）

```
;----- PCﾘﾝｸﾗﾝﾀﾞﾑWRITE ----
PCRWT:
        mov     dx, offset PCRWT
        int     60h
;------- I/Oﾊﾟﾗﾒｰﾀ ------
PCRWT   dw      1551h
        db      0, 5h
        dw      offset LRBUF, seg LRBUF, ?, 0, 100h,
                offset LDBUF, seg LDBUF, ?, 0, 100h, ?, ?,
                offset STB, seg STB, offset STC, seg STC
        db      255, 9 dup(?)
SND     dw      0
LRBUF   dw      256 dub(0)
LDBUF   dw      4096 dub(0)
STB     db      1
STC     db      1
```

I/Oパラメータ

△：設定する［渡す値］
▼：実行後に格納される［得る値］

dsレジスタ ←I/Oパラメータの先頭セグメントアドレス
dxレジスタ ←I/Oパラメータの先頭オフセットアドレス

0		△［コマンドコード］&H155x
2		△［登録ボードNo.］0～3（使用するリンクボードの登録ボードNo.）
		△ 設定値：05固定で使用してください。
4		△［ＬＲバッファアドレス（オフセット）］　＊ＬＤバッファは最大256ワードまでです。
6		△［ＬＲバッファアドレス（セグメント）］
8		
A		△［ＬＲ書き込みポインタ］
		リンクボード内部リンクリレー領域の先頭アドレスからのオフセットを指定します。
C		△［ＬＲ書き込みサイズ］ワード単位
		上記ポインタから書き込むサイズを指定します。
E		△［ＬＤバッファアドレス（オフセット）］　＊ＬＤバッファは最大4096ワードまでです。
10		△［ＬＤバッファアドレス（セグメント）］
12		
14		△［ＬＤ書き込みポインタ］
		リンクボード内部リンクデータ領域の先頭アドレスからのオフセットを指定します。
16		△［ＬＤ書き込みサイズ］ワード単位
		上記ポインタから書き込むサイズを指定します。
18		
1A		
1C		△［動作モードステータス格納アドレス（オフセット）］―― 格納値　0：ＰＲＯＧ
		1：ＲＵＮ
1E		△［動作モードステータス格納アドレス（セグメント）］
20		△［運転状態ステータス格納アドレス（オフセット）］―― 格納値　0：異常
		1：正常
22		△［運転状態ステータス格納アドレス（セグメント）］
24		△［書き込みタイミング］リンクソフトがＰＣ ＬＩＮＫ送信データを書き込むタイミング。

1～254	書き込み時間指定タイプ（ユーザソフト：リンクソフト非同期型）。指定値×200msのタイミング毎にデータを書き込み、書き込み完了後に送信要求完了フラグに0をセットします。 ただし、このタイプでは、データのワード保証はされませんので注意してください。
255	書き込み指定タイプ（ユーザソフト：リンクソフト同期型）。送信要求完了フラグ =1 時に送信データを書き込み、書き込み完了後に送信要求完了フラグに0をセットします。データ更新時は、ユーザプログラム内で送信要求完了フラグに1を設定します。

26		
28		
2A		
2C		
2E		▼△PCWTF［送信要求完了フラグ］書き込み完了時に０またはエラーコードが設定されます。

3-2-5 PC LINK READ

ＰＣリンク通信／ＰＣリンク読み出し（コマンドコード&H144x）

機　能

ＰＣリンクデータの受信要求、または指定されたバッファへ受信データを読み出します。

解　説

・実行するとＰＣリンク用ＣＨの使用を自動的に許可し、PC LINK OPEN/CLOSEコマンドで使用を禁止するまで、許可状態を保持します。
・機能コマンドを実行するには、各種プログラミング言語でI/Oパラメータのセグメントアドレスとオフセットアドレスをds:dxレジスタにセットし、割り込みを実行するユーザープログラムを作成してください。
・コマンド実行後の結果（リターン値）は、axレジスタに格納されます。
　0：正常終了時
　その他：エラーコード（「３－９」参照）

■コマンドタイプ

コード	機能
&H1440	受信要求（データは読み出さない）
&H1441	受信要求＋データ読み出し（タイムアウト付）
&H1442	受信完了センス（完了時データ読み出し付き）
&H1444	データ読み出し（受信要求無し）

注　意

・&H1440・&H1441はPC LINK OPEN機能付きです。
・&H1440では、パラメータ4h以降の設定は不要です。データは読み出しません。
・&H1444では、受信の要求は行いません。
・&H1442では、完了検知時、指定バッファへデータを読み出します。
・PC LINKパラメータが登録されていない時、PC LINKの受信は行えません。
・タイムアウト値は、コントロールレジスタWRITEコマンドにて変更できます。
・ランダムREAD機能起動中は、本コマンドはすべて使用できません。

文　例

●MS-DOSファンクションコール（int21h）

```
;----- PCﾘﾝｸREAD ----
PCRD:
        mov     bx,[FILHDL]
        mov     ah,44h
        mov     al,3
        mov     dx,offset PCRD
        int     21h
;------- I/Oﾊﾟﾗﾒｰﾀ ------
PCRD    dw      1441h
        db      0,5h
        dw      offset LRBUF,seg LRBUF,256,0,100h,
                offset LDBUF,seg LDBUF,4096,0,100h
STA     dw      8 dub(?)
STB     dw      8 dub(?)
STC     dw      8 dub(?)
LRBUF   dw      256 dub(?)
LDBUF   dw      4096 dub(?)
```

●ソフトウェア割り込み（int60h）

```
;----- PCﾘﾝｸREAD ----
PCRD:
        mov     dx,offset PCRD
        int     60h
;------- I/Oﾊﾟﾗﾒｰﾀ ------
PCRD    dw      1441h
        db      0,5h
        dw      offset LRBUF,seg LRBUF,256,0,100h,
                offset LDBUF,seg LDBUF,4096,0,100h
STA     dw      8 dub(?)
STB     dw      8 dub(?)
STC     dw      8 dub(?)
LRBUF   dw      256 dub(?)
LDBUF   dw      4096 dub(?)
```

I/Oパラメータ

△：設定する［渡す値］
▼：実行後に格納される［得る値］

dsﾚｼﾞｽﾀ ←I/Oパラメータの先頭セグメントアドレス
dxﾚｼﾞｽﾀ ←I/Oパラメータの先頭オフセットアドレス

0	△［コマンドコード］&H144x
2	△［登録ボードNo.］0～3　使用するリンクボードの登録ボードNo.を設定します。
	△ 設定値：05固定で使用してください。
4	△［ＬＲバッファアドレス（オフセット）］　＊ＬＲバッファは最大256ワードまでです。
6	△［ＬＲバッファアドレス（セグメント）］
8	△［ＬＲバッファサイズ］ワード単位
A	△［ＬＲ読み出しポインタ］
	リンクボード内部リンクリレー領域の先頭アドレスからのオフセットを指定します。
C	△［ＬＲ読み出しサイズ］ワード単位
	上記ポインタから読み出すサイズを指定します。
E	△［ＬＤバッファ（オフセット）］　＊ＬＤバッファは最大4096ワードまでです。
10	△［ＬＤバッファ（セグメント）］
12	△［ＬＤバッファサイズ］ワード単位
14	△［ＬＤ読み出しポインタ］
	リンクボード内部リンクデータ領域の先頭アドレスからのオフセットを指定します。
16	△［ＬＤ読み出しみサイズ］ワード単位
	上記ポインタから読み出すサイズを指定します。
18	▼［通信保証ステータス］
	0：ＰＣリンクをしていない
1A	1：ＰＣリンクをしている
1C	
1E	
20	▼［動作モードステータス］
	0：ＰＲＯＧ
22	1：ＲＵＮ
24	
26	
28	▼［運転状態ステータス］
	0：異常
2A	1：正常
2C	
2E	

■各ステータス格納位置

（各ステータスのオフセットアドレス）	7	6	5	4	3	2	1	0 (bit)	
n+0	08	07	06	05	04	03	02	01	ユニットNo.
1	16	15	14	13	12	11	10	09	
2	24	23	22	21	20	19	18	17	
3	32	31	30	29	28	27	26	25	
4	40	39	38	37	36	35	34	33	
5	48	47	46	45	44	43	42	41	
6	56	55	54	53	52	51	50	49	
7	64	63	62	61	60	59	58	57	

＊各ビット状態がユニットNo.ごとのフラグ値に相当します。

3-2-6 PC LINKランダムREAD

ＰＣリンク通信／ＰＣリンク読み出し・登録受信タイプ（コマンドコード&H145x）

機　能

指定されたタイミングで受信要求を行い、受信したデータを指定バッファに格納します。。

解　説

・本コマンドで起動後は、リンクソフトを呼び出さなくても、指定したタイミングでPC LINK情報が指定バッファに自動的に格納されます。
・機能コマンドを実行するには、各種プログラミング言語でI/Oパラメータのセグメントアドレスとオフセットアドレスをds:dxレジスタにセットし、割り込みを実行するユーザープログラムを作成してください。
・コマンド実行後の結果（リターン値）は、axレジスタに格納されます。
　0：正常終了時
　その他：エラーコード（「３－９」参照）

■コマンドタイプ

コード	機能
&H1450	PC LINKランダムＲＥＡＤ停止
&H1451	PC LINKランダムＲＥＡＤ起動 （PC LINK OPEN機能付き）

注　意

・&H1450はI/Oパラメータの4h以降の指定は不要です。
・ランダムREADは、PC LINK OPEN機能付きです。
・ランダムREAD起動中は、設定内容の変更はできません（無視されます）。設定内容の変更は、ランダムREADを一旦停止した後に行ってください。
・ランダムREAD起動中は他のPC LINK READコマンドは使用できません。

文　例

●MS-DOSファンクションコール（int21h）

```
;----- PCﾘﾝｸﾗﾝﾀﾞﾑREAD ----
PCRRD:
        mov     bx,[FILHDL]
        mov     ah,44h
        mov     al,3
        mov     dx,offset PCRRD
        int     21h
;------- I/Oﾊﾟﾗﾒｰﾀ ------
PCRRD   dw      1451h
        db      0,5h
        dw      offset LRBUF,seg LRBUF,256,0,100h,
                offset LDBUF,seg LDBUF,4096,0,100,h
        dw      offset STA,seg STB,offset STB,seg STB,
                offset STC,seg STC
        db      255,9 dup(?)
SNDS    dw      0
LRBUF   dw      256 dub(?)
LDBUF   dw      4096 dub(?)
STA     db      8 dup(?)
STB     db      8 dup(?)
STC     db      8 dup(?)
```

●ソフトウェア割り込み（int60h）

```
;----- PCﾘﾝｸﾗﾝﾀﾞﾑREAD ----
PCRRD:
        mov     dx,offset PCRRD
        int     60h
;------- I/Oﾊﾟﾗﾒｰﾀ ------
PCRRD   dw      1451h
        db      0,5h
        dw      offset LRBUF,seg LRBUF,256,0,100h,
                offset LDBUF,seg LDBUF,4096,0,100,h
        dw      offset STA,seg STB,offset STB,seg STB,
                offset STC,seg STC
        db      255,9 dup(?)
SNDS    dw      0
LRBUF   dw      256 dub(?)
LDBUF   dw      4096 dub(?)
STA     db      8 dup(?)
STB     db      8 dup(?)
STC     db      8 dup(?)
```

I/Oパラメータ

△：設定する［渡す値］
▼：実行後に格納される［得る値］

dsレジスタ ←I/Oパラメータの先頭セグメントアドレス
dxレジスタ ←I/Oパラメータの先頭オフセットアドレス

0	△［コマンドコード］設定値：&H145x
2	△［登録ボードNo.］格納値：0～3　使用するリンクボードの登録ボードNo.を設定します。
	△ 設定値：05固定で使用してください。
4	△［ＬＲバッファアドレス（オフセット）］　＊ＬＲバッファは最大256ワードまでです。
6	△［ＬＲバッファアドレス（セグメント）］
8	△［ＬＲバッファサイズ］
A	△［ＬＲ読み出しポインタ］　ＬＲの読み出しポインタを指定します。 リンクボード内部リンクリレー領域の先頭アドレスからのオフセットを指定します。
C	△［ＬＲ読み出しサイズ］ 上記ポインタから読み出すサイズを指定します。ワード単位。
E	△［ＬＤバッファアドレス（オフセット）］　＊ＬＤバッファは最大4096ワードまでです。
10	△［ＬＤバッファアドレス（セグメント）］
12	△［ＬＤバッファサイズ］
14	△［ＬＤ読み出しポインタ］　ＬＤの読み出しポインタを指定します。 リンクボード内部リンクデータ領域先頭アドレスからのオフセットを指定します。
16	△［ＬＤ読み出しサイズ］ 上記ポインタから読み出すサイズを指定します。ワード単位。
18	△［通信保証フラグアドレス（オフセット）］
1A	△［通信保証フラグ格納アドレス（セグメント）］
1C	△［動作モードフラグ格納アドレス（オフセット）］
1E	△［動作モードフラグ格納アドレス（セグメント）］
20	△［運転状態フラグ格納アドレス（オフセット）］
22	△［運転状態フラグ格納アドレス（セグメント）］

■各ステータス格納状態

（各ステータスのオフセットアドレス）	7	6	5	4	3	2	1	0 (bit)	
n+0	08	07	06	05	04	03	02	01	ユニットNo.
1	16	15	14	13	12	11	10	09	
2	24	23	22	21	20	19	18	17	
3	32	31	30	29	28	27	26	25	
4	40	39	38	37	36	35	34	33	
5	48	47	46	45	44	43	42	41	
6	56	55	54	53	52	51	50	49	
7	64	63	62	61	60	59	58	57	

＊各ビット状態がユニットNo.ごとのフラグ値に相当します。

24	△［読み出しタイミング］リンクソフトがＰＣ ＬＩＮＫ送信データを読み出すタイミング。
26	
28	
2A	
2C	
2E	▼△［受信要求完了フラグ］ 読み出し完了時に０またはエラーコードが設定されます。

0	受信完了毎読み出しタイプ（ユーザソフト：リンクソフト非同期型）。受信完了毎にデータを読み出し、読み出し完了後に受信要求完了フラグに0をセットします。
1～254	読み出し時間指定タイプ（ユーザソフト：リンクソフト非同期型）。指定値×200msのタイミングで読み出し、読み出し完了後に受信要求完了フラグに0をセットします。
255	読み出しタイミング指定タイプ（ユーザソフト：リンクソフト同期型）。 受信要求完了フラグ =1 時にPC LINK要求およびデータ読み出しをし、読み出し完了後に受信要求完了フラグに0をセットします。データの更新時は、ユーザプログラム内で再度受信要求完了フラグに1を設定します。

3-2-7 ＰＣリンクのプログラム例

アセンブラプログラム（MS-DOSシステムコールint21）

```
;*********************************************************
; *          ＰＣ ＬＩＮＫ ＲＥＡＤ／ＷＲＩＴＥ          *
; *                    <APCINK.ASM>                       *
; *********************************************************
; *                  ＰＣ９８シリーズ                     *
; *   <PC LINK領域ﾏｯﾌﾟ>                                   *
; *      ﾘﾚｰﾘﾝｸ(LR)   0-09...1番機                        *
; *                  10-19...ﾊﾟﾘｺﾝI/Fﾎﾞｰﾄﾞ               *
; *      ﾃﾞｰﾀﾘﾝｸ(LD)  0-09...1番機                        *
; *                  10-19...ﾊﾟﾘｺﾝI/Fﾎﾞｰﾄﾞ               *
; *                                                       *
; *    ﾘﾚｰﾘﾝｸ(LR) 10-19 及びﾃﾞｰﾀﾘﾝｸ(LD) 10-19 を送信し      *
; *    ﾘﾚｰﾘﾝｸ(LR)  0-09及びﾃﾞｰﾀﾘﾝｸ(LD)  0-09を受信します  *
; *                                                       *
; *  (注)ﾘﾝｸﾎﾞｰﾄﾞ及び1番機に対して事前に、上記内容の       *
; *      PC LINKﾊﾟﾗﾒｰﾀの設定が必要です。                   *
; *                                                       *
; *       使用コマンド：WRITE=1541H(ﾀｲﾑｱｳﾄ付き)           *
; *                     READ =1441H(起動付き読出)         *
; *********************************************************
                                        ;
CODE    SEGMENT
        ASSUME CS:CODE,DS:CODE,ES:CODE,SS:CODE


CR              EQU     0DH             ; CRｺｰﾄﾞ
LF              EQU     0AH             ; LFｺｰﾄﾞ
                                        ;
                                        ;
START:
                MOV     AX,CS
                MOV     DS,AX
                MOV     ES,AX
                MOV     SS,AX
                MOV     AX,OFFSET STPT
                MOV     SP,AX
                                        ;
;  ---------------------------------------------
;                 初 期 設 定
;  ---------------------------------------------
INI:
                MOV     DX,OFFSET MSG1  ; 初期設定メッセージ表示
                CALL    MESSAGE
                MOV     DX,OFFSET INIT  ; 初期設定用I/Oﾊﾟﾗﾒｰﾀ ｱﾄﾞﾚｽ設定
                CALL    COMMAND         ; ﾘﾝｸｿﾌﾄ呼出
                JNB     INI_OK
                JMP     DOS             ; ｴﾗｰ時 MS-DOSへ戻る
                                        ;
INI_OK:
                MOV     DX,OFFSET MSG1_OK ; 初期設定ＯＫﾒｯｾｰｼﾞ 表示
                CALL    MESSAGE
                                        ;
;  ---------------------------------------------
;        ボ ー ド 登 録 ・ 起 動
;  ---------------------------------------------
                                        ;
BDST:
                MOV     DX,OFFSET MSG4  ; ﾎﾞｰﾄﾞ登録･起動メッセージ表示
                CALL    MESSAGE
                MOV     DX,OFFSET BOARD ; ﾎﾞｰﾄﾞ登録用I/Oﾊﾟﾗﾒｰﾀ ｱﾄﾞﾚｽ
                CALL    COMMAND         ; ﾘﾝｸｿﾌﾄ呼出
                JNB     BDST_OK
                JMP     DOS             ; ｴﾗｰ時 MS-DOSへ戻る
                                        ;
BDST_OK:
                MOV     DX,OFFSET MSG4_OK ; ﾎﾞｰﾄﾞ登録&起動OKﾒｯｾｰｼﾞ 表示
                CALL    MESSAGE
                                        ;
;  ---------------------------------------------
;        PC LINK READ/WRITE
```

```
; ---------------------------------------------
                                        ;
LP:
            MOV     DX,OFFSET  PCLKWT_MSG  ; PC LINK WRITEﾒｯｾｰｼﾞ 表示
            CALL    MESSAGE
            CALL    DTKO                   ; 送信ﾃﾞｰﾀ更新
            MOV     DX,OFFSET  PCLKWT
            CALL    COMMAND                ; PC LINK WRITE実施
            JB      LP
                                        ;
            MOV     DX,OFFSET  PCLKWT_OK_MSG      ; PC LINK WRITE OKﾒｯｾｰｼﾞ 表示
            CALL    MESSAGE
                                        ;
            MOV     DX,OFFSET  PCLKRD_MSG  ; PC LINK READ表示
            CALL    MESSAGE
            MOV     DX,OFFSET  PCLKRD      ; PC LINK READ
            CALL    COMMAND
            JB      LP
                                        ;
            MOV     DX,OFFSET  PCLKRD_OK_MSG      ; PC LINK READ OKﾒｯｾｰｼﾞ 表示
            CALL    MESSAGE
                                        ;
            CALL    RXDATA_DSP             ; PC LINK受信内容表示
            JMP     LP
                                        ;
                                        ;
; ============================================
; =          ＭＳ－ＤＯＳへ戻る処理          =
; ============================================
                                        ;
DOS:
            MOV     AH,4CH
            INT     21H
                                        ;
                                        ;
; ============================================
; =          メッセージの画面表示処理          =
; ============================================
                                        ;
MESSAGE:
            MOV     AH,9                   ; DXｱﾄﾞﾚｽから'$'迄を画面表示
            INT     21H
            RET
                                        ;
                                        ;
; ============================================
; =          リンクソフト呼出処理            =
; ============================================
                                        ;
COMMAND:
            INT     60H                    ; ｼｽﾃﾑｺｰﾙ
            CMP     AX,0                   ; AX>0時ｴﾗｰ
            JE      CMD2                   ;
                                        ;
            PUSH    AX
            CALL    BI2ASC                 ; ｴﾗｰｺｰﾄﾞ ASCII変換
            MOV     DX,OFFSET  MSG3        ; ｴﾗｰﾒｯｾｰｼﾞ 画面表示
            CALL    MESSAGE
            STC
            POP     AX
CMD2:
            RET
                                        ;
                                        ;
; ============================================
; =        エラーコードのＡＳＣＩＩ変換       =
; ============================================
                                        ;
BI2ASC:
            MOV     BL,AH
            MOV     AH,AL
            MOV     BH,BL
            AND     AL,0FH
            AND     BL,0FH
            MOV     CL,4
            SHR     AH,CL
```

```
                SHR     BH,CL
                ADD     BX,3030H
                ADD     AX,3030H
                CMP     AL,3AH
                JB      B0
                ADD     AL,7
B0:
                CMP     AH,3AH
                JB      B1
                ADD     AH,7
B1:
                CMP     BL,3AH
                JB      B2
                ADD     BL,7
B2:
                CMP     BH,3AH
                JB      B3
                ADD     BH,7
B3:
                MOV     [MSG3_1+0],BH
                MOV     [MSG3_1+1],BL
                MOV     [MSG3_1+2],AH
                MOV     [MSG3_1+3],AL
                RET
                                        ;
; ========================================
; =        ＢＩＮ==>ＡＳＣＩＩ変換        =
; ========================================
                                        ;
BIN_ASC:                                ; IN...AL=BINｺｰﾄﾞ
                PUSH    SI              ; OUT..AX=ASCIIｺｰﾄﾞ
                PUSH    BX
                PUSH    DX
                                        ;
                MOV     BL,AL
                AND     AL,0F0H         ; 上位４ビット
                SHR     AL,1
                SHR     AL,1
                SHR     AL,1
                SHR     AL,1
                LEA     SI,BINTOASC_TB  ; 変換テーブル
                MOV     AH,0
                ADD     SI,AX
                MOV     DH,BYTE PTR CS:[SI]
                                        ;
                MOV     AL,BL
                AND     AL,0FH          ; 下位４ビット
                LEA     SI,BINTOASC_TB  ; 変換テーブル
                MOV     AH,0
                ADD     SI,AX
                MOV     AL,BYTE PTR CS:[SI]
                MOV     AH,DH
                XCHG    AL,AH
                                        ;
                POP     DX
                POP     BX
                POP     SI
                RET
                                        ;
                                        ;
; ----------------------------------------
;       ＨＥＸ→ＡＳＣＩＩ　変換テーブル
; ----------------------------------------
                                        ;
BINTOASC_TB     DB      "0123456789ABCDEF"
                                        ;
                                        ;
; ----------------------------------------
;                 送信データ内容更新
; ----------------------------------------
                                        ;
DTKO:           MOV     CX,10           ; ﾃﾞｰﾀ数
                LEA     DI,LR           ; ﾃﾞｰﾀ書込先
                ADD     DI,20
                MOV     AX,CS:[LRDT]    ; 前回書込ﾃﾞｰﾀ
                CMP     AX,0FFFFH
```

```
                JNE     DDT1
                XOR     AX,AX                   ; ﾃﾞｰﾀﾘｾｯﾄ
                JMP     DDT2
DDT1:
                ADD     AX,1111H                ; ﾃﾞｰﾀ更新
DDT2:
                MOV     CS:[LRDT],AX            ; 書込ﾃﾞｰﾀ記憶
                CLD
                REP     STOSW                   ; ﾃﾞｰﾀ設定
                                                ;
                MOV     CX,10                   ; ﾃﾞｰﾀ数
                LEA     DI,LD                   ; ﾃﾞｰﾀ書込先
                ADD     DI,20
                MOV     AX,CS:[LDDT]            ; 前回書込ﾃﾞｰﾀ
                CMP     AX,0FFFFH
                JNE     DDT3
                XOR     AX,AX                   ; ﾃﾞｰﾀﾘｾｯﾄ
                JMP     DDT4
DDT3:
                ADD     AX,1111H                ; ﾃﾞｰﾀ更新
DDT4:
                MOV     CS:[LDDT],AX            ; 書込ﾃﾞｰﾀ記憶
                CLD
                REP     STOSW                   ; ﾃﾞｰﾀ設定
                                                ;
                RET
                                                ;
                                                ;
; ----------------------------------------------
;       各種バッファＣＲ，ＬＦ，＄セット
; ----------------------------------------------
                                                ;
CRLFST:
                MOV     BYTE PTR DS:[DI+BX],CR
                MOV     BYTE PTR DS:[DI+BX+1],LF
                MOV     BYTE PTR DS:[DI+BX+2],'$'
                RET
                                                ;
                                                ;
; ==============================================
; =       ＰＣ ＬＩＮＫ受信データ表示          =
; ==============================================
                                                ;
RXDATA_DSP:
                MOV     DX,OFFSET DSP_STA       ; 通信保証表示
                CALL    MESSAGE
                LEA     DI,STA_F
                CALL    STATUS_DSP
                                                ;
                MOV     DX,OFFSET DSP_STB       ; PC動作ﾓｰﾄﾞ表示
                CALL    MESSAGE
                LEA     DI,STB_F
                CALL    STATUS_DSP
                                                ;
                MOV     DX,OFFSET DSP_STC       ; PC運転状態表示
                CALL    MESSAGE
                LEA     DI,STC_F
                CALL    STATUS_DSP
                                                ;
                XOR     CX,CX
                LEA     DI,LR
                LEA     SI,LD
RX_D_LP:
                MOV     DX,OFFSET DSP_LR        ; "LR" 表示
                CALL    MESSAGE
                MOV     AL,CL
                CALL    BIN_ASC
                CALL    BIN_DSP                 ; LRxxのxx表示
                MOV     DX,OFFSET DSP_E
                CALL    MESSAGE                 ; "=" 表示
                MOV     BX,DS:[DI]
                MOV     AL,BH
                CALL    BIN_ASC
                CALL    BIN_DSP                 ; ﾘﾚｰﾘﾝｸ上位ﾊﾞｲﾄ表示
                MOV     AL,BL
                CALL    BIN_ASC
```

```
                CALL    BIN_DSP                 ; ﾘﾚｰﾘﾝｸ下位ﾊﾞｲﾄ表示
                                                ;
                MOV     DX,OFFSET  DSP_LD       ; "LD" 表示
                CALL    MESSAGE
                MOV     AL,CL
                CALL    BIN_ASC
                CALL    BIN_DSP                 ; LDxxのxx表示
                MOV     DX,OFFSET  DSP_E
                CALL    MESSAGE                 ; "=" 表示
                MOV     BX,DS:[SI]
                MOV     AL,BH
                CALL    BIN_ASC
                CALL    BIN_DSP                 ; ﾃﾞｰﾀﾘﾝｸ上位ﾊﾞｲﾄ表示
                MOV     AL,BL
                CALL    BIN_ASC
                CALL    BIN_DSP                 ; ﾃﾞｰﾀﾘﾝｸ下位ﾊﾞｲﾄ表示
                                                ;
                MOV     DX,OFFSET  CRLF         ; 改行復帰
                CALL    MESSAGE
                                                ;
                ADD     DI,2                    ; ﾘﾚｰﾘﾝｸ ﾎﾟｲﾝﾀｰ更新
                ADD     SI,2                    ; ﾃﾞｰﾀﾘﾝｸ ﾎﾟｲﾝﾀｰ更新
                INC     CL
                CMP     CL,10
                JB      RX_D_LP
                                                ;
                MOV     DX,OFFSET  CRLF
                CALL    MESSAGE
                RET
                                                ;
                                                ;
STATUS_DSP:                                     ; 各種ｽﾃｰﾀｽ表示
                MOV     CX,8
ST_D_LP:
                MOV     AL,DS:[DI]
                CALL    BIN_ASC
                CALL    BIN_DSP
                MOV     AX,2020H
                CALL    BIN_DSP
                INC     DI
                LOOP    ST_D_LP
                                                ;
                MOV     DX,OFFSET  CRLF         ; 改行復帰
                CALL    MESSAGE
                                                ;
                RET
                                                ;
                                                ;
BIN_DSP:                                        ; IN...AX=表示文字ｺｰﾄﾞ
                PUSH    AX
                MOV     DL,AL
                MOV     AH,2
                INT     21H
                POP     AX
                MOV     DL,AH
                MOV     AH,2
                INT     21H
                RET
                                                ;
                                                ;
; ==========================================
; =          I／Oパラメータ                =
; ==========================================
                                                ; = 初期設定 =
INIT            DW      1000H                   ; 初期設定ｺﾏﾝﾄﾞｺｰﾄﾞ
                DB      0                       ; ｱﾌﾟﾘｹｰｼｮﾝ･ｿﾌﾄﾓｰﾄﾞ
                DB      1                       ; 使用ﾎﾞｰﾄﾞ枚数
                                                ;
                                                ; = ﾎﾞｰﾄﾞ登録･起動 =
BOARD           DW      1100H                   ; ﾎﾞｰﾄﾞ登録･起動ｺﾏﾝﾄﾞｺｰﾄﾞ
                DB      0                       ; ﾎﾞｰﾄﾞ登録No.
                DB      27                      ; 使用ｾｸﾞﾒﾝﾄｱﾄﾞﾚｽNo.27(D0000H)
                DB      0                       ; 割り込みNo.
                DB      0
                                                ;
                                                ; = PC LINK WRITE =
```

パソコン機種別変更箇所
（左記はＰＣ９８シリーズの場合）

```
PCLKWT          DW      1541H                   ; 書込ｺﾏﾝﾄﾞｺｰﾄﾞ
                DB      0                       ; 登録ﾎﾞｰﾄﾞNO.
                DB      5
                DW      OFFSET LR+20            ; ﾘﾚｰ送信ﾊﾞｯﾌｧOFFSET
WTLRBFSG        DW      SEG LR                  ; ﾘﾚｰ送信ﾊﾞｯﾌｧSEGMENT
                DW      0
LTXPT           DW      20                      ; ﾘﾚｰ送信ﾎﾟｲﾝﾀｰ
LTXSIZ          DW      10                      ; ﾘﾚｰ送信ﾃﾞｰﾀ･ｻｲｽﾞ
                DW      OFFSET LD+20            ; ﾃﾞｰﾀ送信ﾊﾞｯﾌｧOFFSET
WTLDBFSG        DW      SEG LD                  ; ﾃﾞｰﾀ送信ﾊﾞｯﾌｧSEGMENT
                DW      0
DTXPT           DW      20                      ; ﾃﾞｰﾀ送信ﾎﾟｲﾝﾀｰ
DTXSIZ          DW      10                      ; ﾃﾞｰﾀ送信･ｻｲｽﾞ
                DB      1                       ; 動作ﾓｰﾄﾞ
                DB      1                       ; 運転状態
                DW      11 DUP(0)
                                                ;
                                                ;
                                                ; = PC LINK READ(読出) =
PCLKRD          DW      1441H                   ; 読出ｺﾏﾝﾄﾞｺｰﾄﾞ
                DB      0                       ; 登録ﾎﾞｰﾄﾞNO.
                DB      5
                DW      OFFSET LR               ; ﾘﾚｰ受信ﾊﾞｯﾌｧOFFSET
RDLRBFSG        DW      SEG LR                  ; ﾘﾚｰ受信ﾊﾞｯﾌｧSEGMENT
                DW      256                     ; ﾘﾚｰﾊﾞｯﾌｧ･ｻｲｽﾞ
LRXPT           DW      0                       ; ﾘﾚｰ読出ﾎﾟｲﾝﾀｰ
LRXSIZ          DW      10                      ; ﾘﾚｰ読出ｻｲｽﾞ
                DW      OFFSET LD               ; ﾃﾞｰﾀ受信ﾊﾞｯﾌｧOFFSET
RDLDBFSG        DW      SEG LD                  ; ﾃﾞｰﾀ受信ﾊﾞｯﾌｧSEGMENT
                DW      4096                    ; ﾃﾞｰﾀﾊﾞｯﾌｧ･ｻｲｽﾞ
DRXPT           DW      0                       ; ﾃﾞｰﾀ読出ﾎﾟｲﾝﾀｰ
DRXSIZ          DW      10                      ; ﾃﾞｰﾀ読出ｻｲｽﾞ
STA_F           DB      8 DUP(0)                ; 通信保証領域
STB_F           DB      8 DUP(0)                ; PC動作ﾓｰﾄﾞ領域
STC_F           DB      8 DUP(0)                ; PC運転状態領域
                                                ;
                                                ;
; ==========================================
; =          メッセージ ／ バッファ領域          =
; ==========================================
                                                ;
MSG1            DB      '初期設定===>$'
MSG1_OK         DB      'ＯＫ！'
                DB      CR,LF,'$'

MSG3            DB      'ｴﾗｰ.....ｺｰﾄﾞ='
MSG3_1          DB      0,0,0,0
                DB      CR,LF,'$'

MSG4            DB      '使用ボード登録 & 起動===>$'
MSG4_OK         DB      'ＯＫ！'
                DB      CR,LF,'$'

PCLKWT_MSG      DB      'ＰＣ　ＬＩＮＫ送信===>$'
PCLKWT_OK_MSG   DB      'ＯＫ！',CR,LF,'$'

PCLKRD_MSG      DB      'ＰＣ　ＬＩＮＫ受信===>$'
PCLKRD_OK_MSG   DB      'ＯＫ！',CR,LF,'$'

DSP_STA         DB      '[ 通信保証　 ]=$'
DSP_STB         DB      '[ PC動作ﾓｰﾄﾞ ]=$'
DSP_STC         DB      '[ PC運転状態 ]=$'

DSP_LR          DB      'LR$'
DSP_LD          DB      '          LD$'
DSP_E           DB      '=$'

CRLF            DB      CR,LF,'$'

LR              DW      256 DUP(0)                  ; ﾘﾚｰﾘﾝｸﾊﾞｯﾌｧ

LD              DW      4096 DUP(0)                 ; ﾃﾞｰﾀﾘﾝｸﾊﾞｯﾌｧ

LRDT            DW      0FFFFH
LDDT            DW      7777H
```

```
STACK           DW      256 DUP(0)                      ; ｽﾀｯｸ領域
STPT            DB      0


                CODE    ENDS
                END     START
```

```
; ********************************************************
; *      PC LINKﾗﾝﾀﾞﾑREAD/WRITE                            *
; *                    <APCINK2.ASM>                     *
; ********************************************************
; *                  PC98ｼﾘｰｽﾞ                           *
; *  読出し/書込ﾀｲﾐﾝｸﾞ指定ﾀｲﾌﾟ                           *
; *  ﾊﾟｿｺﾝI/Fﾎﾞｰﾄﾞ に設定されているﾊﾟﾗﾒｰﾀでPC LINKを       *
; *  行います。                                          *
; *   また、実行中にﾊﾟﾗﾒｰﾀが変更された場合、ﾊﾟﾗﾒｰﾀの読出   *
; *  を自動的に行い、変更された内容でPC LINKを再開します *
; *                                                      *
; *       使用ｺﾏﾝﾄﾞ：ﾗﾝﾀﾞﾑWRITE=1551H(起動)             *
; *                  ﾗﾝﾀﾞﾑREAD =1451H(起動)             *
; *                  ﾊﾟﾗﾒｰﾀREAD=1900H                   *
; *                  ｽﾃｰﾀｽREAD =1300H                   *
; ********************************************************
                                        ;
CODE    SEGMENT
        ASSUME CS:CODE,DS:CODE,ES:CODE,SS:CODE


CR              EQU     0DH             ; CRｺｰﾄﾞ
LF              EQU     0AH             ; LFｺｰﾄﾞ
                                        ;
                                        ;
START:
                MOV     AX,CS
                MOV     DS,AX
                MOV     ES,AX
                MOV     SS,AX
                MOV     AX,OFFSET STPT
                MOV     SP,AX
                                        ;
;  ---------------------------------------------
;                 初 期 設 定
;  ---------------------------------------------
INI:
                MOV     DX,OFFSET MSG1          ; 初期設定ﾒｯｾｰｼﾞ表示
                CALL    MESSAGE
                MOV     DX,OFFSET INIT          ; 初期設定用I/Oﾊﾟﾗﾒｰﾀ ｱﾄﾞﾚｽ設定
                CALL    COMMAND                 ; ﾘﾝｸｿﾌﾄ呼出
                JNB     INI_OK
                JMP     DOS                     ; ｴﾗｰ時 MS-DOSへ戻る
                                                ;
INI_OK:
                MOV     DX,OFFSET MSG1_OK       ; 初期設定ＯＫﾒｯｾｰｼﾞ 表示
                CALL    MESSAGE
                                        ;
;  ---------------------------------------------
;        ボ ー ド 登 録 ・ 起 動
;  ---------------------------------------------
                                        ;
BDST:
                MOV     DX,OFFSET MSG4          ; ﾎﾞｰﾄﾞ登録･起動ﾒｯｾｰｼﾞ表示
                CALL    MESSAGE
                MOV     DX,OFFSET BOARD         ; ﾎﾞｰﾄﾞ登録用I/Oﾊﾟﾗﾒｰﾀ ｱﾄﾞﾚｽ
                CALL    COMMAND                 ; ﾘﾝｸｿﾌﾄ呼出
                JNB     BDST_OK
```

```
                JMP     DOS                     ; ｴﾗｰ時 MS-DOSへ戻る
                                                ;
BDST_OK:
                MOV     DX,OFFSET  MSG4_OK      ; ﾎﾞｰﾄﾞ登録&起動OKﾒｯｾｰｼﾞ 表示
                CALL    MESSAGE
                                                ;
;  ---------------------------------------------
;               ＰＣリンクパラメータの読出
;  ---------------------------------------------
                                                ;
PAR_RD:
                MOV     DX,OFFSET  PARRD_MSG    ; ﾊﾟﾗﾒｰﾀ読出ﾒｯｾｰｼﾞ 表示
                CALL    MESSAGE
                MOV     DX,OFFSET  PARRD                ; PCﾘﾝｸﾊﾟﾗﾒｰﾀ 読出
                CALL    COMMAND
                JNB     PAR_OK
                JMP     DOS
                                                ;
PAR_OK:
                MOV     DX,OFFSET  PARRD_OK_MSG         ; ﾊﾟﾗﾒｰﾀ内容表示
                CALL    MESSAGE
                                                ;
                LEA     SI,PAR_BUF
                LEA     DI,D_TB
                MOV     BX,DS:[SI+1EH]          ; ﾘﾚｰﾘﾝｸ領域先頭ﾎﾟｲﾝﾀｰ
                MOV     DS:[DI+8],BX
                MOV     DS:LRXPT,BX
                MOV     BX,DS:[SI+20H]
                SHR     BX,1                    ; ﾜｰﾄﾞ変換
                MOV     DS:LRXSIZ,BX            ; ﾘﾚｰﾘﾝｸ領域ｻｲｽﾞｾｯﾄ
                MOV     DS:[DI+10],BX
                                                ;
                MOV     BX,DS:[SI+24H]          ; ﾃﾞｰﾀﾘﾝｸ領域先頭ﾎﾟｲﾝﾀｰ
                MOV     DS:[DI+12],BX
                MOV     DS:DRXPT,BX
                MOV     BX,DS:[SI+26H]
                SHR     BX,1                    ; ﾜｰﾄﾞ変換
                MOV     DS:DRXSIZ,BX            ; ﾃﾞｰﾀﾘﾝｸ受信ｻｲｽﾞｾｯﾄ
                MOV     DS:[DI+14],BX
                                                ;
                MOV     BL,DS:LKNO              ; BL=自局ﾕﾆｯﾄNO.
                XOR     BH,BH
                SHL     BX,1                    ; *2
                SHL     BX,1                    ; *4
                SHL     BX,1                    ; *8
                ADD     BX,20H                  ; BX=自局送信ﾊﾟﾗﾒｰﾀ位置
                MOV     BP,BX
                MOV     BX,DS:[SI+BP]
                MOV     DS:LTXPT,BX             ; ﾘﾚｰﾘﾝｸ送信ﾎﾟｲﾝﾀｰ
                MOV     DS:[DI],BX
                MOV     BX,DS:[SI+BP+2]
                SHR     BX,1                    ; ﾜｰﾄﾞ変換
                MOV     DS:LTXSIZ,BX            ; ﾘﾚｰﾘﾝｸ送信ｻｲｽﾞ
                MOV     DS:[DI+2],BX
                                                ;
                MOV     BX,DS:[SI+BP+4]
                MOV     DS:DTXPT,BX             ; ﾃﾞｰﾀﾘﾝｸ送信ﾎﾟｲﾝﾀ
                MOV     DS:[DI+4],BX
                MOV     BX,DS:[SI+BP+6]
                SHR     BX,1                    ; ﾜｰﾄﾞ変換
                MOV     DS:DTXSIZ,BX            ; ﾃﾞｰﾀﾘﾝｸ送信ｻｲｽﾞ
                MOV     DS:[DI+6],BX
                                                ;
                LEA     SI,P1
                MOV     CX,8
PAR_RD_LP:                                      ; ﾊﾟﾗﾒｰﾀ内容表示
                MOV     DX,SI
                CALL    MESSAGE
                CALL    PAR_DSP
                ADD     SI,19
                ADD     DI,2
                LOOP    PAR_RD_LP
                                                ;
                MOV     DX,OFFSET  CRLF         ; 改行復帰
                CALL    MESSAGE
                                                ;
```

```
; ------------------------------------------
;       ＰＣ ＬＩＮＫﾗﾝﾀﾞﾑＲ／Ｗ起動
; ------------------------------------------
                                        ;
PCLKGO:
            MOV     DX,OFFSET  PCRNWT_ST_MSG      ; ﾗﾝﾀﾞﾑWRITE起動ﾒｯｾｰｼﾞ表示
            CALL    MESSAGE
            MOV     DX,OFFSET  PCLKWT
            CALL    COMMAND               ; ﾗﾝﾀﾞﾑWRITE起動
            JB      PCER
                                        ;
            MOV     DX,OFFSET  PCRNWT_STOK_MSG    ; ﾗﾝﾀﾞﾑWRITE登録OK
            CALL    MESSAGE
            MOV     WORD PTR CS:[WTF],0
                                        ;
            MOV     DX,OFFSET  PCRNRD_ST_MSG      ; ﾗﾝﾀﾞﾑREAD起動ﾒｯｾｰｼﾞ表示
            CALL    MESSAGE
            MOV     DX,OFFSET  PCLKRD     ;
            CALL    COMMAND               ; ﾗﾝﾀﾞﾑREAD起動
            JB      PCER
                                        ;
            MOV     DX,OFFSET  PCRNRD_STOK_MSG    ; ﾗﾝﾀﾞﾑREAD登録OK
            CALL    MESSAGE
            MOV     WORD PTR CS:[RDF],1   ; 受信要求
            JMP     LP
PCER:
            JMP     DOS
                                        ;
; ------------------------------------------
;   ＰＣ ＬＩＮＫ ﾗﾝﾀﾞﾑＲＥＡＤ/ＷＲＩＴＥ
; ------------------------------------------
                                        ;
LP:
            MOV     AX,CS:[WTF]
            CMP     AX,0                  ; 送信完了?
            JNE     RD                    ; NO ===>RD へ
                                          ; <<<PC LINKﾗﾝﾀﾞﾑWRITE>>>
            MOV     DX,OFFSET  PCRNWT_MSG ; ﾗﾝﾀﾞﾑWRITE実施ﾒｯｾｰｼﾞ表示
            CALL    MESSAGE
            CALL    DTKO                  ; 送信ﾃﾞｰﾀ更新
            XOR     DS:WTSTB_F,1          ; 動作ﾓｰﾄﾞ反転
            MOV     WORD PTR CS:[WTF],1   ; 次送信要求
RD:
            MOV     AX,CS:[RDF]
            CMP     AX,0                  ; 受信完了?
            JNE     CK                    ; NO ===>CK へ
                                          ; <<<PC LINKﾗﾝﾀﾞﾑREAD>>>
            MOV     DX,OFFSET  PCRNRD_MSG ; ﾗﾝﾀﾞﾑREAD実施ﾒｯｾｰｼﾞ表示
            CALL    MESSAGE
            CALL    RXDATA_DSP            ; 受信ﾃﾞｰﾀ表示
            MOV     WORD PTR CS:[RDF],1   ; 次受信要求
CK:
            CMP     CS:[WTF],1            ; WRITE状態OK ?
            JA      WTER                  ; NO ===>WTER へ
            CMP     CS:[RDF],1            ; READ状態OK ?
            JNA     LP                    ; YES===>LP へ
RDER:                                     ; ...NO
            MOV     DX,OFFSET  RDER_MSG   ; ﾗﾝﾀﾞﾑREAD異常ﾒｯｾｰｼﾞ表示
            CALL    MESSAGE
            MOV     AX,CS:[RDF]
            JMP     RNRW_ED
WTER:
            MOV     DX,OFFSET  WTER_MSG   ; ﾗﾝﾀﾞﾑWRITE異常ﾒｯｾｰｼﾞ表示
            CALL    MESSAGE
            MOV     AX,CS:[WTF]
RNRW_ED:
            CALL    ERR_DSP               ; 異常内容表示
                                        ;
                                        ;
; ==========================================
; =       ＰＣ ＬＩＮＫステータス読出       =
; ==========================================
                                        ;
PCSTRD:
            MOV     DX,OFFSET  PCSTRD_MSG ; ｽﾃｰﾀｽ読出ﾒｯｾｰｼﾞ表示
```

```
                CALL    MESSAGE
                MOV     DX,OFFSET  STRD
                CALL    COMMAND                 ; PCﾘﾝｸｽﾃｰﾀｽ 読出
                JB      DOS
                                                ;
                LEA     DI,PCST
                MOV     AX,DS:[DI+28]
                CMP     AX,1                    ; PC LINK起動?
                JE      PCSTRD_OK               ; YES===>PCSTRD_OK へ
                                                ; ...NO
                MOV     DX,OFFSET  PCSTRD_BD_MSG        ; PC LINK停止中ﾒｯｾｰｼﾞ 表示
                CALL    MESSAGE
                JMP     PCSTRD
PCSTRD_OK:
                MOV     DX,OFFSET  PCSTRD_OK_MSG        ; PC LINK起動ﾒｯｾｰｼﾞ 表示
                CALL    MESSAGE
                JMP     PAR_RD                  ; ===>ﾊﾟﾗﾒｰﾀ 読出へ
                                                ;
; ============================================
; =          ＭＳ－ＤＯＳへ戻る処理          =
; ============================================
                                                ;
DOS:
                MOV     AH,4CH
                INT     21H
                                                ;
                                                ;
; ============================================
; =          メッセージの画面表示処理          =
; ============================================
                                                ;
MESSAGE:
                MOV     AH,9                    ; DXｱﾄﾞﾚｽから'$'迄を画面表示
                INT     21H
                RET
                                                ;
                                                ;
; ============================================
; =          リンクソフト呼出処理              =
; ============================================
                                                ;
COMMAND:
                INT     60H                     ; ｼｽﾃﾑｺｰﾙ
                CMP     AX,0                    ; AX>0時ｴﾗｰ
                JE      CMD2                    ;
                                                ;
ERR_DSP:
                PUSH    AX
                CALL    BI2ASC                  ; ｴﾗｰｺｰﾄﾞ ASCII変換
                MOV     DX,OFFSET  MSG3         ; ｴﾗｰﾒｯｾｰｼﾞ 画面表示
                CALL    MESSAGE
                STC
                POP     AX
CMD2:
                RET
                                                ;
                                                ;
; ============================================
; =        エラーコードのＡＳＣＩＩ変換       =
; ============================================
                                                ;
BI2ASC:
                MOV     BL,AH
                MOV     AH,AL
                MOV     BH,BL
                AND     AL,0FH
                AND     BL,0FH
                MOV     CL,4
                SHR     AH,CL
                SHR     BH,CL
                ADD     BX,3030H
                ADD     AX,3030H
                CMP     AL,3AH
                JB      B0
                ADD     AL,7
B0:
```

```
                CMP     AH,3AH
                JB      B1
                ADD     AH,7
B1:
                CMP     BL,3AH
                JB      B2
                ADD     BL,7
B2:
                CMP     BH,3AH
                JB      B3
                ADD     BH,7
B3:
                MOV     [MSG3_1+0],BH
                MOV     [MSG3_1+1],BL
                MOV     [MSG3_1+2],AH
                MOV     [MSG3_1+3],AL
                RET
                                                ;
; ============================================
; =        ＢＩＮ==>ＡＳＣＩＩ変換          =
; ============================================
                                                ;
BIN_ASC:                                        ; IN...AL=BINｺｰﾄﾞ
                PUSH    SI                      ; OUT..AX=ASCIIｺｰﾄﾞ
                PUSH    BX
                PUSH    DX
                                                ;
                MOV     BL,AL
                AND     AL,0F0H                 ; 上位４ビット
                SHR     AL,1
                SHR     AL,1
                SHR     AL,1
                SHR     AL,1
                LEA     SI,BINTOASC_TB          ; 変換テーブル
                MOV     AH,0
                ADD     SI,AX
                MOV     DH,BYTE PTR CS:[SI]
                                                ;
                MOV     AL,BL
                AND     AL,0FH                  ; 下位４ビット
                LEA     SI,BINTOASC_TB          ; 変換テーブル
                MOV     AH,0
                ADD     SI,AX
                MOV     AL,BYTE PTR CS:[SI]
                MOV     AH,DH
                XCHG    AL,AH
                                                ;
                POP     DX
                POP     BX
                POP     SI
                RET
                                                ;
                                                ;
; --------------------------------------------
;       ＨＥＸ→ＡＳＣＩＩ　変換テーブル
; --------------------------------------------
                                                ;
BINTOASC_TB     DB      "0123456789ABCDEF"
                                                ;
                                                ;
; --------------------------------------------
;               送信データ内容更新
; --------------------------------------------
                                                ;
DTK0:           MOV     CX,LTXSIZ               ; ﾃﾞｰﾀ数
                LEA     DI,TLR                  ; ﾃﾞｰﾀ書込先
                MOV     AX,CS:[LRDT]            ; 前回書込ﾃﾞｰﾀ
                CMP     AX,0FFFFH
                JNE     DDT1
                XOR     AX,AX                   ; ﾃﾞｰﾀﾘｾｯﾄ
                JMP     DDT2
DDT1:
                ADD     AX,1111H                ; ﾃﾞｰﾀ更新
DDT2:
                MOV     CS:[LRDT],AX            ; 書込ﾃﾞｰﾀ記憶
                CLD
```

```
                REP     STOSW                   ; ﾃﾞｰﾀ設定
                                                ;
                MOV     CX,DTXSIZ               ; ﾃﾞｰﾀ数
                LEA     DI,TLD                  ; ﾃﾞｰﾀ書込先
                MOV     AX,CS:[LDDT]            ; 前回書込ﾃﾞｰﾀ
                CMP     AX,0FFFFH
                JNE     DDT3
                XOR     AX,AX                   ; ﾃﾞｰﾀﾘｾｯﾄ
                JMP     DDT4
DDT3:
                ADD     AX,1111H                ; ﾃﾞｰﾀ更新
DDT4:
                MOV     CS:[LDDT],AX            ; 書込ﾃﾞｰﾀ記憶
                CLD
                REP     STOSW                   ; ﾃﾞｰﾀ設定
                                                ;
                RET
                                                ;
                                                ;
; ==========================================
; =      ＰＣ ＬＩＮＫパラメータ内容の表示      =
; ==========================================
                                                ;
PAR_DSP:
                PUSH    DX
                PUSH    BX
                PUSH    AX
                                                ;
                MOV     BX,DS:[DI]
                MOV     AL,BH
                CALL    BIN_ASC
                CALL    BIN_DSP                 ; 上位ﾊﾞｲﾄ表示
                MOV     AL,BL
                CALL    BIN_ASC
                CALL    BIN_DSP                 ; 下位ﾊﾞｲﾄ表示
                                                ;
                MOV     DX,OFFSET CRLF
                CALL    MESSAGE
                                                ;
                POP     AX
                POP     BX
                POP     DX
                RET
                                                ;
                                                ;
; ==========================================
; =        ＰＣ ＬＩＮＫ受信データ表示        =
; ==========================================
                                                ;
RXDATA_DSP:
                PUSH    DI
                PUSH    AX
                PUSH    BX
                PUSH    CX
                PUSH    DX
                                                ;
                MOV     DX,OFFSET DSP_STA       ; 通信保証表示
                CALL    MESSAGE
                LEA     DI,STA_F
                CALL    STATUS_DSP
                                                ;
                MOV     DX,OFFSET DSP_STB       ; PC動作ﾓｰﾄﾞ表示
                CALL    MESSAGE
                LEA     DI,STB_F
                CALL    STATUS_DSP
                                                ;
                MOV     DX,OFFSET DSP_STC       ; PC運転状態表示
                CALL    MESSAGE
                LEA     DI,STC_F
                CALL    STATUS_DSP
                                                ;
                LEA     DI,LR
                MOV     DX,OFFSET LR_MSG        ; "[ ﾘﾚｰﾘﾝｸ  ]"表示
                CALL    MESSAGE
                                                ;
                MOV     CX,DS:LRXSIZ
```

```
                MOV     BP,LRXPT
LR_DSP_LP:                              ; ﾘﾚｰﾘﾝｸ受信内容表示
                MOV     BX,DS:[DI+BP]
                MOV     AL,BH
                CALL    BIN_ASC
                CALL    BIN_DSP         ; 上位ﾊﾞｲﾄ表示
                MOV     AL,BL
                CALL    BIN_ASC
                CALL    BIN_DSP         ; 下位ﾊﾞｲﾄ表示
                MOV     AX,2020H
                CALL    BIN_DSP
                ADD     BP,2
                LOOP    LR_DSP_LP
                                        ;
                MOV     DX,OFFSET CRLF  ; 改行復帰
                CALL    MESSAGE
                                        ;
                LEA     DI,LD
                MOV     DX,OFFSET LD_MSG ; "[ ﾃﾞｰﾀﾘﾝｸ  ]"表示
                CALL    MESSAGE
                                        ;
                MOV     CX,DS:DRXSIZ
                MOV     BP,DRXPT
LD_DSP_LP:                              ; ﾃﾞｰﾀﾘﾝｸ受信内容表示
                MOV     BX,DS:[DI+BP]
                MOV     AL,BH
                CALL    BIN_ASC
                CALL    BIN_DSP         ; 上位ﾊﾞｲﾄ表示
                MOV     AL,BL
                CALL    BIN_ASC
                CALL    BIN_DSP         ; 下位ﾊﾞｲﾄ表示
                MOV     AX,2020H
                CALL    BIN_DSP
                ADD     BP,2
                LOOP    LD_DSP_LP
                                        ;
                MOV     DX,OFFSET CRLF  ; 改行復帰
                CALL    MESSAGE
                MOV     DX,OFFSET CRLF  ; 改行復帰
                CALL    MESSAGE
                                        ;
                POP     DX
                POP     CX
                POP     BX
                POP     AX
                POP     DI
                RET
                                        ;
                                        ;
STATUS_DSP:                             ; 各種ｽﾃｰﾀｽ表示
                PUSH    CX
                PUSH    AX
                PUSH    DX
                PUSH    DI
                                        ;
                MOV     CX,8
ST_D_LP:
                MOV     AL,DS:[DI]
                CALL    BIN_ASC
                CALL    BIN_DSP
                MOV     AX,2020H
                CALL    BIN_DSP
                INC     DI
                LOOP    ST_D_LP
                                        ;
                MOV     DX,OFFSET CRLF  ; 改行復帰
                CALL    MESSAGE
                                        ;
                POP     DI
                POP     DX
                POP     AX
                POP     CX
                RET
                                        ;
                                        ;
BIN_DSP:                                ; IN...AX=表示文字ｺｰﾄﾞ
```

```
                PUSH    DX
                                                ;
                PUSH    AX
                MOV     DL, AL
                MOV     AH, 2
                INT     21H
                POP     AX
                MOV     DL, AH
                MOV     AH, 2
                INT     21H
                                                ;
                POP     DX
                RET
                                                ;
                                                ;
; ==========================================
; =              I／Oパラメータ                =
; ==========================================
                                                ; = 初期設定 =
INIT            DW      1000H                   ; 初期設定ｺﾏﾝﾄﾞｺｰﾄﾞ
                DB      0                       ; ｱﾌﾟﾘｹｰｼｮﾝ･ｿﾌﾄﾓｰﾄﾞ
                DB      1                       ; 使用ﾎﾞｰﾄﾞ枚数
                                                ;
                                                ; = ﾎﾞｰﾄﾞ登録･起動 =
BOARD           DW      1100H                   ; ﾎﾞｰﾄﾞ登録･起動ｺﾏﾝﾄﾞｺｰﾄﾞ
                DB      0                       ; ﾎﾞｰﾄﾞ登録No.
                DB      27                      ; 使用ｾｸﾞﾒﾝﾄｱﾄﾞﾚｽNo.27(D0000H)
                DB      0                       ; 割り込みNo.
                DB      0
                                                ;
                                                ; = PCﾘﾝｸﾊﾟﾗﾒｰﾀ読出 =
PARRD           DW      1900H                   ; ｺﾏﾝﾄﾞｺｰﾄﾞ
                DB      0                       ; 登録ﾎﾞｰﾄﾞNO.
LKNO            DB      0                       ; 自局ﾕﾆｯﾄNO.
                DW      0                       ; 読出開始要素NO.
                DW      311                     ; 読出要素数
                DW      OFFSET PAR_BUF          ; 読出領域ｵﾌｾｯﾄ
                DW      SEG PAR_BUF             ; 読出領域ｾｸﾞﾒﾝﾄ
                DW      622                     ; 読出領域ｻｲｽﾞ
                                                ;
                                                ; = PC LINK ﾗﾝﾀﾞﾑWRITE =
PCLKWT          DW      1551H                   ; ﾗﾝﾀﾞﾑWRITE起動ｺﾏﾝﾄﾞｺｰﾄﾞ
                DB      0                       ; 登録ﾎﾞｰﾄﾞNO.
                DB      5
                DW      OFFSET TLR              ; ﾘﾚｰ送信ﾊﾞｯﾌｧOFFSET
WTLRBFSG        DW      SEG TLR                 ; ﾘﾚｰ送信ﾊﾞｯﾌｧSEGMENT
                DW      0
LTXPT           DW      0                       ; ﾘﾚｰ送信ﾎﾟｲﾝﾀｰ
LTXSIZ          DW      0                       ; ﾘﾚｰ送信ﾃﾞｰﾀ･ｻｲｽﾞ
                DW      OFFSET TLD              ; ﾃﾞｰﾀ送信ﾊﾞｯﾌｧOFFSET
WTLDBFSG        DW      SEG TLD                 ; ﾃﾞｰﾀ送信ﾊﾞｯﾌｧSEGMENT
                DW      0
DTXPT           DW      0                       ; ﾃﾞｰﾀ送信ﾎﾟｲﾝﾀｰ
DTXSIZ          DW      0                       ; ﾃﾞｰﾀ送信･ｻｲｽﾞ
                DW      0
                DW      0
                DW      OFFSET WTSTB_F          ; 動作ﾓｰﾄﾞ
                DW      SEG WTSTB_F
                DW      OFFSET WTSTC_F          ; 運転状態
                DW      SEG WTSTC_F
                DB      255                     ; ﾀｲﾐﾝｸﾞ=ﾌﾗｸﾞﾀｲﾌﾟ 指定
                DB      0
                DW      4 DUP(0)
WTF             DW      0                       ; 完了ﾌﾗｸﾞ
                                                ;
                                                ; = PC LINK ﾗﾝﾀﾞﾑREAD =
PCLKRD          DW      1451H                   ; ﾗﾝﾀﾞﾑREAD起動ｺﾏﾝﾄﾞｺｰﾄﾞ
                DB      0                       ; 登録ﾎﾞｰﾄﾞNO.
                DB      5                               ;
                DW      OFFSET LR               ; ﾘﾚｰ受信ﾊﾞｯﾌｧOFFSET
                DW      SEG LR                  ; ﾘﾚｰ受信ﾊﾞｯﾌｧSEGMENT
                DW      256                     ; ﾘﾚｰﾊﾞｯﾌｧ･ｻｲｽﾞ
LRXPT           DW      0                       ; ﾘﾚｰ読出ﾎﾟｲﾝﾀｰ
LRXSIZ          DW      0                       ; ﾘﾚｰ読出ｻｲｽﾞ
                DW      OFFSET LD               ; ﾃﾞｰﾀ受信ﾊﾞｯﾌｧOFFSET
                DW      SEG LD                  ; ﾃﾞｰﾀ受信ﾊﾞｯﾌｧSEGMENT
```

パソコン機種別変更箇所
（左記はＰＣ９８シリーズの場合）

```
                DW      4096                    ; ﾃﾞｰﾀﾊﾞｯﾌｧ･ｻｲｽﾞ
DRXPT           DW      0                       ; ﾃﾞｰﾀ読出ﾎﾟｲﾝﾀｰ
DRXSIZ          DW      0                       ; ﾃﾞｰﾀ読出ｻｲｽﾞ
                DW      OFFSET STA_F            ; 通信保証領域
                DW      SEG STA_F
                DW      OFFSET STB_F            ; PC動作ﾓｰﾄﾞ領域
                DW      SEG STB_F
                DW      OFFSET STC_F            ; PC運転状態領域
                DW      SEG STC_F
                DB      255                     ; ﾀｲﾐﾝｸﾞ=ﾌﾗｸﾞﾀｲﾌﾟ 指定
                DB      0
                DW      4 DUP(0)
RDF             DW      0                       ; 完了ﾌﾗｸﾞ
                                                ;
                                                ; = ｽﾃｰﾀｽ読出 =
STRD            DW      1300H                   ; ｺﾏﾝﾄﾞｺｰﾄﾞ
                DB      0                       ; 登録ﾎﾞｰﾄﾞNO.
                DB      0
                DW      616                     ; 読出開始ﾚｼﾞｽﾀNo.
                DW      16                      ; 読出ﾃﾞｰﾀ数
                DW      OFFSET PCST             ; 読出ﾃﾞｰﾀ格納OFFSET
                DW      SEG PCST                ;                 SEGMENT


; ============================================
; =          ﾒｯｾｰｼﾞ ／ ﾊﾞｯﾌｧ領域             =
; ============================================
                                                ;
MSG1            DB      '初期設定===>$'
MSG1_OK         DB      'ＯＫ！'
                DB      CR,LF,'$'

MSG3            DB      'ｴﾗｰ.....ｺｰﾄﾞ='
MSG3_1          DB      0,0,0,0
                DB      CR,LF,'$'

MSG4            DB      '使用ボード登録 & 起動===>$'
MSG4_OK         DB      'ＯＫ！'
                DB      CR,LF,'$'

PARRD_MSG       DB       'PC LINK ﾊﾟﾗﾒｰﾀ読出===>$'
PARRD_OK_MSG    DB      'ＯＫ！',CR,LF,'$'

PCRNWT_ST_MSG DB        'PC LINKﾗﾝﾀﾞﾑWRITE登録===>$'
PCRNWT_STOK_MSG         DB      'ＯＫ！',CR,LF,'$'

PCRNRD_ST_MSG DB        'PC LINKﾗﾝﾀﾞﾑREAD登録===>$'
PCRNRD_STOK_MSG         DB      'ＯＫ！',CR,LF,'$'

PCRNWT_MSG      DB      'PC LINK ﾗﾝﾀﾞﾑWRITE実施'
                DB      CR,LF,'$'

PCRNRD_MSG      DB      'PC LINK ﾗﾝﾀﾞﾑREAD実施'
                DB      CR,LF,'$'

PCSTRD_MSG      DB      'PC LINKｽﾃｰﾀｽ読出===>$'
PCSTRD_OK_MSG DB        'ＰＣ ＬＩＮＫ起動'
                DB      CR,LF,'$'
PCSTRD_BD_MSG DB        'ＰＣ ＬＩＮＫ停止中'
                DB      CR,LF,'$'

WTER_MSG        DB      'PC LINKﾗﾝﾀﾞﾑWRITE異常===>$'
RDER_MSG        DB      'PC LINKﾗﾝﾀﾞﾑREAD異常===>$'

DSP_STA         DB      '[ 通信保証    ]=$'
DSP_STB         DB      '[ PC動作ﾓｰﾄﾞ  ]=$'
DSP_STC         DB      '[ PC運転状態  ]=$'

P1              DB      'ﾘﾚｰﾘﾝｸ送信ﾎﾟｲﾝﾀｰ  =$'
P2              DB      'ﾘﾚｰﾘﾝｸ送信ｻｲｽﾞ     =$'
P3              DB      'ﾃﾞｰﾀﾘﾝｸ送信ﾎﾟｲﾝﾀｰ=$'
P4              DB      'ﾃﾞｰﾀﾘﾝｸ送信ｻｲｽﾞ    =$'
P5              DB      'ﾘﾚｰﾘﾝｸ領域先頭     =$'
P6              DB      'ﾘﾚｰﾘﾝｸ領域ｻｲｽﾞ     =$'
P7              DB      'ﾃﾞｰﾀﾘﾝｸ領域先頭    =$'
P8              DB      'ﾃﾞｰﾀﾘﾝｸ領域ｻｲｽﾞ    =$'
```

```
CRLF            DB      CR,LF,'$'

LR_MSG          DB      '[ リレーリンク  ]=$'
LD_MSG          DB      '[ データリンク ]=$'

LR              DW      256 DUP(0)              ; リレーリンク受信バッファ
LD              DW      4096 DUP(0)             ; データリンク受信バッファ
TLR             DW      256 DUP(0)              ; リレーリンク送信バッファ
TLD             DW      4096 DUP(0)             ; データリンク送信バッファ

PAR_BUF         DW      311 DUP(0)              ; パラメータ用
PCST            DW      16  DUP(0)              ; ステータス読出用

STA_F           DB      8 DUP(0)                ; 通信保証用
STB_F           DB      8 DUP(0)                ; PC動作モード用
STC_F           DB      8 DUP(0)                ; PC運転状態用

D_TB            DW      8 DUP(0)                ; パラメータ内容表示用

WTSTB_F         DB      1                       ; 自局動作モード
WTSTC_F         DB      1                       ; 自局運転状態

LRDT            DW      0FFFFH
LDDT            DW      7777H

STACK           DW      256 DUP(0)              ; スタック領域
STPT            DB      0


                CODE    ENDS
                END     START
```

ＰＣ／ＡＴ互換機

```
BOARD           DW      1100H                   ; ボード登録・起動コマンドコード
                DB      0                       ; ボード登録No.
                DB      8                       ; 使用セグメントアドレスNo.
                DB      10                      ; 割り込みNo.
                DB      0
```

ＦＭＲシリーズ

```
BOARD           DW      1100H                   ; ボード登録・起動コマンドコード
                DB      0                       ; ボード登録No.
                DB      7                       ; 基板I/OポートアドレスNo.
                DW      7800H                   ; コントロール・ステータスI/Oポートアドレス
                DB      5                       ; 割り込みNo.
                DB      0
```

Cプログラム（ソフトウェア割り込みint60）

```
/*************************************************************/
/*                PC LINK RD/WT                              */
/*                     (PCLK.C)                              */
/*************************************************************/
/*                    PC98シリーズ                           */
/*     <PC LINK 領域ﾏｯﾌﾟ>                                    */
/*       ﾘﾚｰﾘﾝｸ(LR)   0-09....1番機                          */
/*                   10-19....ﾊﾟｿｺﾝI/Fﾎﾞｰﾄﾞ                 */
/*       ﾃﾞｰﾀﾘﾝｸ(LD)  0-09....1番機                          */
/*                   10-19....ﾊﾟｿｺﾝI/Fﾎﾞｰﾄﾞ                 */
/*                                                           */
/*        ﾘﾚｰﾘﾝｸ(LR)10-19 及びﾃﾞｰﾀﾘﾝｸ(LD)10-19 を送信し       */
/*        ﾘﾚｰﾘﾝｸ(LR)  0-09及びﾃﾞｰﾀﾘﾝｸ(LD)  0-09を受信します  */
/*                                                           */
/*   (注) ﾘﾝｸﾎﾞｰﾄﾞ 及び1番機に対して事前に、上記内容の       */
/*        PC LINKﾊﾟﾗﾒｰﾀの設定が必要です。                     */
/*                                                           */
/*       使用コマンド：WRITE=1541H(ﾀｲﾑｱｳﾄ付き)               */
/*                     READ =1441H(起動付き読出)             */
/*************************************************************/

#include  <dos.h>
#include  <stdio.h>
#include  <stdlib.h>

#define PC_N  1                     /* 局番 1           */

static union REGS inregs;
static union REGS outregs;
struct SREGS segregs;
static char far *p1;
static unsigned char dat[48];
unsigned int lr[256];               /* ﾘﾚｰﾘﾝｸ送受信ﾊﾞｯﾌｧ  */
unsigned int ld[4098];              /* ﾃﾞｰﾀﾘﾝｸ送受信ﾊﾞｯﾌｧ */

void main( void )
{
       int    i;
       int    err;
       long   lwork;
       char   cwork[5];
       char   c_dmy1[4];
       char   *dmy;
       unsigned int lrdt,lddt;

/* -------------------------*/
/* ------ 初期設定処理 ------*/
/* -------------------------*/

       dat[0] = 0x00;        /* 初期設定ｺﾏﾝﾄﾞｺｰﾄﾞ=1000H  */
       dat[1] = 0x10;
       dat[2] = 0;           /* ｱﾌﾟﾘｹｰｼｮﾝ･ｿﾌﾄﾓｰﾄﾞ=0       */
       dat[3] = 1;           /* 使用ボード枚数=1          */

       printf("初期設定===>");

       if((err = mew_send()) != 0)
       {
              printf("ｴﾗｰ.....ｺｰﾄﾞ= %x¥n", err);
              exit(-1);
       }
       printf("ＯＫ！¥n");

/* ------------------------------------*/
/* ------ 使用ボード登録 及び起動 ------*/
/* ------------------------------------*/
```

```
        dat[0] = 0x00;          /* 使用ボード登録及び起動    */
        dat[1] = 0x11;          /* ｺﾏﾝﾄﾞｺｰﾄﾞ=1100H           */
        dat[2] = 0x00;          /* 登録ボードNo.=0           */
        dat[3] = 27;            /* 使用ｾｸﾞﾒﾝﾄｱﾄﾞﾚｽNo=27     */
        dat[4] = 0;             /* 割り込みNo.=0             */

        printf("使用ボード登録 & 起動===>");

        if((err = mew_send()) != 0)
        {
                printf("ｴﾗｰ.....ｺｰﾄﾞ= %x¥n", err);
                exit(-1);
        }
        printf("ＯＫ！¥n");

/* ---------------------------*/
/* ------ PC LINK WRITE ------*/
/* ---------------------------*/

        lrdt = 0xFFFF;
        lddt = 0x7777;
LP:
        if(lrdt == 0xFFFF)
                lrdt = 0;
        else
                lrdt = lrdt + 0x1111;       /* ﾘﾚｰﾘﾝｸ送信ﾃﾞｰﾀ更新    */

        if(lddt == 0xFFFF)
                lddt = 0;
        else
                lddt = lddt + 0x1111;       /* ﾃﾞｰﾀﾘﾝｸ送信ﾃﾞｰﾀ更新 */

        for(i = 10 ; i < 20 ; i++)
        {
                lr[i] = lrdt;
                ld[i] = lddt;
        }

        dat[0] = 0x41;              /* PC LINK WRITE       */
        dat[1] = 0x15;              /* ｺﾏﾝﾄﾞｺｰﾄﾞ=1541H     */
        dat[2] = 0x00;              /* 登録ボードno.=0   */
        dat[3] = 0x05;
        p1 = (char far *)&lr[10];           /* ﾘﾚｰﾘﾝｸ             */
        *((unsigned *)&dat[4]) = FP_OFF(p1);        /* 送信ﾊﾞｯﾌｧｵﾌｾｯﾄ    */
        *((unsigned *)&dat[6]) = FP_SEG(p1);        /* 送信ﾊﾞｯﾌｧｾｸﾞﾒﾝﾄ   */
        *((unsigned *)&dat[10])= 20;        /* 書き込みﾎﾟｲﾝﾀｰ    */
        *((unsigned *)&dat[12])= 10;        /* 書込ｻｲｽﾞ(ﾜｰﾄﾞ)    */

        p1 = (char far *)&ld[10];           /* ﾃﾞｰﾀﾘﾝｸ            */
        *((unsigned *)&dat[14]) = FP_OFF(p1);       /* 送信ﾊﾞｯﾌｧｵﾌｾｯﾄ    */
        *((unsigned *)&dat[16]) = FP_SEG(p1);       /* 送信ﾊﾞｯﾌｧｾｸﾞﾒﾝﾄ   */
        *((unsigned *)&dat[20])= 20;        /* 書き込みﾎﾟｲﾝﾀｰ    */
        *((unsigned *)&dat[22])= 10;        /* 書込ｻｲｽﾞ(ﾜｰﾄﾞ)    */

        dat[24] = 1;                        /* 動作ﾓｰﾄﾞ(RUN)      */
        dat[25] = 1;                        /* 運転状態(正常)    */

        printf("ＰＣ　ＬＩＮＫ送信===>");

        if((err = mew_send()) != 0)
        {
                printf("ｴﾗｰ.....ｺｰﾄﾞ= %x ¥n", err);
        }
        else
        {
                printf("ＯＫ！¥n");
        }

/* ---------------------------*/
/* ------ PC LINK READ ------*/
/* ---------------------------*/


        dat[0] = 0x41;              /* PC LINK READ        */
        dat[1] = 0x14;              /* ｺﾏﾝﾄﾞｺｰﾄﾞ=1441H     */
```

パソコン機種別変更箇所
（左記はＰＣ９８シリーズの場合）

```
        dat[2] = 0x00;                  /* 登録ボードno.=0 */
        dat[3] = 0x5;
        p1 = (char far *)&lr[0];        /* リレーリンク          */
        *((unsigned *)&dat[4]) = FP_OFF(p1);        /* 受信ﾊﾞｯﾌｧｵﾌｾｯﾄ  */
        *((unsigned *)&dat[6]) = FP_SEG(p1);        /* 受信ﾊﾞｯﾌｧｾｸﾞﾒﾝﾄ */
        *((unsigned *)&dat[8]) = 256;               /* 受信ﾊﾞｯﾌｧｻｲｽﾞ    */
        *((unsigned *)&dat[10])= 0;                 /* 読み出しﾎﾟｲﾝﾀｰ  */
        *((unsigned *)&dat[12])= 10;                /* 読出ｻｲｽﾞ(ﾜｰﾄﾞ)  */

        p1 = (char far *)&ld[0];            /* ﾃﾞｰﾀﾘﾝｸ          */
        *((unsigned *)&dat[14]) = FP_OFF(p1);       /* 受信ﾊﾞｯﾌｧｵﾌｾｯﾄ  */
        *((unsigned *)&dat[16]) = FP_SEG(p1);       /* 受信ﾊﾞｯﾌｧｾｸﾞﾒﾝﾄ */
        *((unsigned *)&dat[18]) = 4096;             /* 受信ﾊﾞｯﾌｧｻｲｽﾞ    */
        *((unsigned *)&dat[20])= 0;                 /* 読み出しﾎﾟｲﾝﾀｰ  */
        *((unsigned *)&dat[22])= 10;                /* 読出ｻｲｽﾞ(ﾜｰﾄﾞ)  */

        printf("ＰＣ　ＬＩＮＫ受信===>");

        if((err = mew_send()) != 0)
        {
                printf("ｴﾗｰ.....ｺｰﾄﾞ= %x¥n", err);
        }
        else
        {
                printf("ＯＫ！¥n");

                printf("[ 通信保証    ]=");          /* 通信保証表示 */
                for(i = 0 ; i < 8 ; i++)
                {
                        printf(" %x",dat[24+i]);
                }
                printf("¥n");

                printf("[ PC動作ﾓｰﾄﾞ ]=");          /* PC動作ﾓｰﾄﾞ表示 */
                for(i = 0 ; i < 8 ; i++)
                {
                        printf(" %x",dat[32+i]);
                }
                printf("¥n");

                printf("[ PC運転状態 ]=");          /* PC運転状態表示 */
                for(i = 0 ; i < 8 ; i++)
                {
                        printf(" %x",dat[40+i]);
                }
                printf("¥n");

                for(i = 0 ; i < 10 ; i++)           /* 受信ﾃﾞｰﾀ表示 */
                {
                        printf("LR%d=%x",i,lr[i]);
                        printf("          LD%d=%x¥n",i,ld[i]);
                }
                printf("¥n");
        }
}


/* ---------------------------*/
/* ------ ﾘﾝｸﾘｽﾄ呼び出し------*/
/* ---------------------------*/
int mew_send(void)
{

        p1 = (char far *)&dat[0];

        inregs.x.dx = FP_OFF(p1);
        segregs.ds = FP_SEG(p1);

        int86x( 0x60, &inregs, &outregs, &segregs);

        return(outregs.x.ax);
}
```

```
/*************************************************************/
/*              PC LINK ランダムRD/WT                        */
/*                    (PCLK2.C)                              */
/*************************************************************/
/*                  PC98シリーズ                             */
/*    読出し/書込タイミング指定タイプ                        */
/*                                                           */
/*    ﾊﾟﾘｺﾝI/Fﾎﾞｰﾄﾞ に設定されているﾊﾟﾗﾒｰﾀでPC LINKを        */
/*    行います。                                             */
/*    また、実行中にﾊﾟﾗﾒｰﾀが変更された場合、ﾊﾟﾗﾒｰﾀの読出    */
/*    を自動的に行い、変更された内容でPC LINKを再開します   */
/*                                                           */
/*       使用コマンド：ﾗﾝﾀﾞﾑWRITE=1551H(起動)                */
/*                     ﾗﾝﾀﾞﾑREAD =1451H(起動)                */
/*                     ﾊﾟﾗﾒｰﾀREAD=1900H                      */
/*                     ｽﾃｰﾀｽREAD =1300H                      */
/*************************************************************/

#include  <dos.h>
#include  <stdio.h>
#include  <stdlib.h>

#define PC_N  1                         /* 局番 1         */

static union REGS inregs;
static union REGS outregs;
struct SREGS segregs;
static char far *p1;
static unsigned char dat[48];
static unsigned char rdat[48];              /* ﾗﾝﾀﾞﾑREAD登録用I/Oﾊﾟﾗﾒｰﾀﾃｰﾌﾞﾙ  */
unsigned int lr[256];                       /* ﾘﾚｰﾘﾝｸ受信用      */
unsigned int ld[4096];                      /* ﾃﾞｰﾀﾘﾝｸ受信用     */
unsigned int tlr[256];                      /* ﾘﾚｰﾘﾝｸ送信用      */
unsigned int tld[4096];                     /* ﾃﾞｰﾀﾘﾝｸ送信用     */
unsigned int par[311];                      /* PC LINK ﾊﾟﾗﾒｰﾀ用 */
unsigned char sta[8];                       /* 通信保証用        */
unsigned char stb[8];                       /* PC動作ﾓｰﾄﾞ用      */
unsigned char stc[8];                       /* PC運転状態用      */
unsigned int pcst[16];                      /* ｽﾃｰﾀｽﾘｰﾄﾞ用       */

void main( void )
{
        int     i;
        int     err,stcf,stbf,jkpt,wtf,rdf;
        long    lwork;
        char    cwork[5];
        char    c_dmy1[4];
        char    *dmy;
        unsigned int lrdt,lddt;
        unsigned int ltxpt,ltxsiz,dtxpt,dtxsiz,lrxpt,lrxsiz,drxpt,drxsiz;

/* ---------------------------*/
/* ------ 初期設定処理 ------*/
/* ---------------------------*/

        dat[0] = 0x00;          /* 初期設定ｺﾏﾝﾄﾞ･ｺｰﾄﾞ=1000H  */
        dat[1] = 0x10;
        dat[2] = 0;             /* ｱﾌﾟﾘｹｰｼｮﾝ･ｿﾌﾄﾓｰﾄﾞ=0        */
        dat[3] = 1;             /* 使用ボード枚数=1            */

        printf("初期設定===>");

        if((err = mew_send()) != 0)
        {
                printf("ｴﾗｰ.....ｺｰﾄﾞ=  %x¥n",  err);
                exit(-1);
        }
        printf("ＯＫ！¥n");

/* ------------------------------------*/
/* ------ 使用ボード登録 及び起動 ------*/
/* ------------------------------------*/
```

```
        dat[0] = 0x00;          /* 使用ボード登録及び起動    */
        dat[1] = 0x11;          /* ｺﾏﾝﾄﾞｺｰﾄﾞ=1100H           */
        dat[2] = 0x00;          /* 登録ボードNo.=0           */
        dat[3] = 27;            /* 使用ｾｸﾞﾒﾝﾄｱﾄﾞﾚｽNo=27     */
        dat[4] = 0;             /* 割り込みNo.=0             */

        printf("使用ボード登録 & 起動===>");

        if((err = mew_send()) != 0)
        {
                printf("ｴﾗｰ.....ｺｰﾄﾞ= %x¥n", err);
                exit(-1);
        }
        printf("ＯＫ！¥n");

        lrdt = 0xFFFF;                  /* 送信ﾃﾞｰﾀ初期値(ﾘﾚｰﾘﾝｸ用)   */
        ldat = 0x7777;                  /*                (ﾃﾞｰﾀﾘﾝｸ用) */

/* ---------------------------------*/
/* ------ PC LINK ﾊﾟﾗﾒｰﾀ読出 ------*/
/* ---------------------------------*/

PARRD:
        dat[0] = 0x00;
        dat[1] = 0x19;          /* PC LINK ﾊﾟﾗﾒｰﾀ読出ｺﾏﾝﾄﾞｺｰﾄﾞ=1900H  */
        dat[2] = 0x00;          /* 登録ﾎﾞｰﾄﾞNo.          */
        dat[4] = 0x00;
        dat[5] = 0x00;          /* 読出開始要素No.=0  */
        dat[6] = 0x18;
        dat[7] = 0x1;           /* 読出要素数=280     */
        p1 = (char far *)&par[0];
        *((unsigned *)&dat[8]) = FP_OFF(p1);        /* 読出ﾃﾞｰﾀ格納ｱﾄﾞﾚｽ ｵﾌｾｯﾄ   */
        *((unsigned *)&dat[10]) = FP_SEG(p1);       /*                      ｾｸﾞﾒﾝﾄ */
        *((unsigned *)&dat[12]) = 311;              /* 読出ﾃﾞｰﾀ格納ｴﾘｱｻｲｽﾞ       */

        printf("PC LINK ﾊﾟﾗﾒｰﾀ読出===>");

        if((err = mew_send()) != 0)
        {
                printf("ｴﾗｰ.....ｺｰﾄﾞ= %x ¥n", err);
                exit(-1);
        }
        else
        {
                printf("ＯＫ！¥n");

                jkpt   = dat[3] * 4 + 16;   /* 自局送信ﾃﾞｰﾀ格納位置   */
                ltxpt  = par[jkpt];         /* ﾘﾚｰﾘﾝｸ送信ﾎﾟｲﾝﾀｰ       */
                ltxsiz = par[jkpt+1] / 2;   /* ﾘﾚｰﾘﾝｸ送信ｻｲｽﾞ(ﾜｰﾄﾞ)   */
                dtxpt  = par[jkpt+2];       /* ﾃﾞｰﾀﾘﾝｸ送信ﾎﾟｲﾝﾀｰ       */
                dtxsiz = par[jkpt+3] / 2;   /* ﾃﾞｰﾀﾘﾝｸ送信ｻｲｽﾞ(ﾜｰﾄﾞ)  */
                lrxpt  = par[15];           /* ﾘﾚｰﾘﾝｸ領域先頭ﾎﾟｲﾝﾀｰ   */
                lrxsiz = par[16] / 2;       /* ﾘﾚｰﾘﾝｸ領域ｻｲｽﾞ(ﾜｰﾄﾞ)   */
                drxpt  = par[18];           /* ﾃﾞｰﾀﾘﾝｸ領域先頭ﾎﾟｲﾝﾀｰ  */
                drxsiz = par[19] / 2;       /* ﾃﾞｰﾀﾘﾝｸ領域ｻｲｽﾞ(ﾜｰﾄﾞ)  */
                printf("ﾘﾚｰﾘﾝｸ送信ﾎﾟｲﾝﾀｰ  =%x¥n",ltxpt);
                printf("ﾘﾚｰﾘﾝｸ送信ｻｲｽﾞ     =%x¥n",ltxsiz);
                printf("ﾃﾞｰﾀﾘﾝｸ送信ﾎﾟｲﾝﾀｰ=%x¥n",dtxpt);
                printf("ﾃﾞｰﾀﾘﾝｸ送信ｻｲｽﾞ    =%x¥n",dtxsiz);
                printf("ﾘﾚｰﾘﾝｸ領域先頭     =%x¥n",lrxpt);
                printf("ﾘﾚｰﾘﾝｸ領域ｻｲｽﾞ     =%x¥n",lrxsiz);
                printf("ﾃﾞｰﾀﾘﾝｸ領域先頭    =%x¥n",drxpt);
                printf("ﾃﾞｰﾀﾘﾝｸ領域ｻｲｽﾞ    =%x¥n",drxsiz);
        }

/* ------------------------------------*/
/* ------ PC LINK ﾗﾝﾀﾞﾑWRITE登録 ------*/
/* ------------------------------------*/

        dat[0] = 0x51;          /* PC LINK ﾗﾝﾀﾞﾑWRITE  */
        dat[1] = 0x15;          /* ｺﾏﾝﾄﾞｺｰﾄﾞ=1551H     */
        dat[2] = 0x00;          /* 登録ボードno.=0   */
        dat[3] = 0x05;
        p1 = (char far *)&tlr[0];                   /* ﾘﾚｰﾘﾝｸ             */
        *((unsigned *)&dat[4]) = FP_OFF(p1);        /* 送信ﾊﾞｯﾌｧｵﾌｾｯﾄ    */
```

パソコン機種別変更箇所
（左記はＰＣ９８シリーズの場合）

```
        *((unsigned *)&dat[6])  = FP_SEG(p1);       /* 送信ﾊﾞｯﾌｧｾｸﾞﾒﾝﾄ   */
        *((unsigned *)&dat[10])  = ltxpt;           /* 書き込みﾎﾟｲﾝﾀｰ    */
        *((unsigned *)&dat[12])  = ltxsiz;    /* 書込ｻｲｽﾞ(ﾜｰﾄﾞ)    */

        p1 = (char far *)&tld[0];             /* ﾃﾞｰﾀﾘﾝｸ            */
        *((unsigned *)&dat[14])  = FP_OFF(p1);      /* 送信ﾊﾞｯﾌｧｵﾌｾｯﾄ    */
        *((unsigned *)&dat[16])  = FP_SEG(p1);      /* 送信ﾊﾞｯﾌｧｾｸﾞﾒﾝﾄ   */
        *((unsigned *)&dat[20])  = dtxpt;           /* 書き込みﾎﾟｲﾝﾀｰ    */
        *((unsigned *)&dat[22])  = dtxsiz;          /* 書込ｻｲｽﾞ(ﾜｰﾄﾞ)    */

        p1 = (char far *)&stbf;                     /* 動作ﾓｰﾄﾞ          */
        *((unsigned *)&dat[28])  = FP_OFF(p1);      /* ﾊﾞｯﾌｧｵﾌｾｯﾄ        */
        *((unsigned *)&dat[30])  = FP_SEG(p1);      /* ﾊﾞｯﾌｧｾｸﾞﾒﾝﾄ       */

        p1 = (char far *)&stcf;                     /* 運転状態          */
        *((unsigned *)&dat[32])  = FP_OFF(p1);      /* ﾊﾞｯﾌｧｵﾌｾｯﾄ        */
        *((unsigned *)&dat[34])  = FP_SEG(p1);      /* ﾊﾞｯﾌｧｾｸﾞﾒﾝﾄ       */

        dat[36] = 255;                        /* 書込ﾀｲﾐﾝｸﾞ=ﾌﾗｸﾞ    */

        printf("PC  LINKﾗﾝﾀﾞﾑWRITE登録===>");

        if((err = mew_send())  != 0)
        {
              printf("ｴﾗｰ.....ｺｰﾄﾞ=  %x ¥n", err);
              exit(-1);
        }
        else
        {
              printf("ＯＫ！¥n");
              dat[46] = 0;
              dat[47] = 0;
        }


/* ----------------------------------*/
/* ------ PC LINK ﾗﾝﾀﾞﾑREAD登録 ------*/
/* ----------------------------------*/


        rdat[0] = 0x51;       /* PC LINK ﾗﾝﾀﾞﾑREAD  */
        rdat[1] = 0x14;       /* ｺﾏﾝﾄﾞｺｰﾄﾞ=1451H    */
        rdat[2] = 0x00;       /* 登録ボードno.=0    */
        rdat[3] = 0x5;
        p1 = (char far *)&lr[0];                    /* ﾘﾚｰﾘﾝｸ            */
        *((unsigned *)&rdat[4])  = FP_OFF(p1);      /* 受信ﾊﾞｯﾌｧｵﾌｾｯﾄ    */
        *((unsigned *)&rdat[6])  = FP_SEG(p1);      /* 受信ﾊﾞｯﾌｧｾｸﾞﾒﾝﾄ   */
        *((unsigned *)&rdat[8])  = 256;             /* 受信ﾊﾞｯﾌｧｻｲｽﾞ      */
        *((unsigned *)&rdat[10])  = 0;              /* 読出ﾎﾟｲﾝﾀｰ        */
        *((unsigned *)&rdat[12])  = lrxsiz;         /* 読出ｻｲｽﾞ(ﾜｰﾄﾞ)    */

        p1 = (char far *)&ld[0];                    /* ﾃﾞｰﾀﾘﾝｸ            */
        *((unsigned *)&rdat[14])  = FP_OFF(p1);     /* 受信ﾊﾞｯﾌｧｵﾌｾｯﾄ    */
        *((unsigned *)&rdat[16])  = FP_SEG(p1);     /* 受信ﾊﾞｯﾌｧｾｸﾞﾒﾝﾄ   */
        *((unsigned *)&rdat[18])  = 4096;           /* 受信ﾊﾞｯﾌｧｻｲｽﾞ      */
        *((unsigned *)&rdat[20])  = 0;              /* 読出ﾎﾟｲﾝﾀｰ        */
        *((unsigned *)&rdat[22])  = drxsiz;         /* 読出ｻｲｽﾞ(ﾜｰﾄﾞ)    */

        p1 = (char far *)&sta[0];                   /* 通信保証          */
        *((unsigned *)&rdat[24])  = FP_OFF(p1);     /* ﾊﾞｯﾌｧｵﾌｾｯﾄ        */
        *((unsigned *)&rdat[26])  = FP_SEG(p1);     /* ﾊﾞｯﾌｧｾｸﾞﾒﾝﾄ       */

        p1 = (char far *)&stb[0];    /* 動作ﾓｰﾄﾞ          */
        *((unsigned *)&rdat[28])  = FP_OFF(p1);      /* ﾊﾞｯﾌｧｵﾌｾｯﾄ        */
        *((unsigned *)&rdat[30])  = FP_SEG(p1);      /* ﾊﾞｯﾌｧｾｸﾞﾒﾝﾄ       */

        p1 = (char far *)&stc[0];                   /* 運転状態          */
        *((unsigned *)&rdat[32])  = FP_OFF(p1);     /* ﾊﾞｯﾌｧｵﾌｾｯﾄ        */
        *((unsigned *)&rdat[34])  = FP_SEG(p1);     /* ﾊﾞｯﾌｧｾｸﾞﾒﾝﾄ       */

        rdat[36] = 255;                             /* 読出ﾀｲﾐﾝｸﾞ=ﾌﾗｸﾞ   */

        printf("PC  LINK ﾗﾝﾀﾞﾑREAD登録===>");

        if((err = mew_send2())  != 0)
        {
```

```
                printf("ｴﾗｰ.....ｺｰﾄﾞ= %x¥n", err);
                exit(-1);
        }
        else
        {
                printf("ＯＫ！¥n");
                rdat[46] = 1;
                rdat[47] = 0;
        }

/* ------------------------------*/
/* ------ PC LINK WRITE/READ ------*/
/* ------------------------------*/

        do
        {
                wtf = dat[46] + dat[47] * 0x100;
                rdf = rdat[46] + rdat[47] * 0x100;

                if(wtf == 0)
                {
                        printf("PC LINK ﾗﾝﾀﾞﾑWRITE実施¥n");
                        if(lrdt == 0xFFFF)
                                lrdt = 0;
                        else
                                lrdt = lrdt + 0x1111;

                        if(lddt == 0xFFFF)
                                lddt = 0;
                        else
                                lddt = lddt + 0x1111;

                        for( i = 0 ; i < ltxsiz ; i++)
                        {
                                tlr[i] = lrdt;
                        }
                        for( i = 0 ; i < dtxsiz ; i++)
                        {
                                tld[i] = lddt;
                        }

                        if(stbf == 1)                   /* 動作ﾓｰﾄﾞ      */
                                stbf = 0;               /* prog          */
                        else
                                stbf = 1;               /* run           */

                        stcf = 1;                       /* 運転状態=正常 */
                        dat[46] = 1;                    /* PC LINK 送信要求 */
                        dat[47] = 0;
                }

                if(rdf == 0)
                {
                        printf("PC LINK ﾗﾝﾀﾞﾑREAD実施¥n");
                        printf("[ 通信保証   ]=");  /* 通信保証表示 */
                        for(i = 0 ; i < 8 ; i++)
                        {
                                printf(" %x",sta[i]);
                        }
                        printf("¥n");

                        printf("[ PC動作ﾓｰﾄﾞ ]=");  /* PC動作ﾓｰﾄﾞ表示 */
                        for(i = 0 ; i < 8 ; i++)
                        {
                                printf(" %x",stb[i]);
                        }
                        printf("¥n");

                        printf("[ PC運転状態 ]=");  /* PC運転状態表示 */
                        for(i = 0 ; i < 8 ; i++)
                        {
                                printf(" %x",stc[i]);
                        }
                        printf("¥n");

                        printf("[ ﾘﾚｰﾘﾝｸ  ]=");          /* 受信ﾃﾞｰﾀ表示 */
                        for(i = 0 ; i < lrxsiz ; i++)
```

```
                        {
                                printf(" %x",lr[i]);
                        }
                        printf("¥n");

                        printf("[ ﾃﾞｰﾀﾘﾝｸ ]=");
                        for(i = 0 ; i < drxsiz ; i++)
                        {
                                printf(" %x",ld[i]);
                        }

                        printf("¥n");
                        rdat[46] = 1;                   /* PC LINK受信要求 */
                        rdat[47] = 0;
                }
        }
        while((wtf<2) && (rdf<2));

/* ------------------------------*/
/* ------ PC LINK ｽﾃｰﾀｽ読出 ------*/
/* ------------------------------*/

PCSTRD:
        dat[0] = 0x00;
        dat[1] = 0x13;          /* PC LINK ｽﾃｰﾀｽ読出ｺﾏﾝﾄﾞｺｰﾄﾞ=1300H  */
        dat[2] = 0x00;          /* 登録ﾎﾞｰﾄﾞNo.                  */
        dat[3] = 0x00;
        dat[4] = 0x68;
        dat[5] = 0x2;           /* 読出開始ﾚｼﾞｽﾀｰNo.=616        */
        dat[6] = 0x10;
        dat[7] = 0x00;          /* 読出ﾃﾞｰﾀ数=16                */
        p1 = (char far *)&pcst[0];
        *((unsigned *)&dat[8])= FP_OFF(p1);         /* 読出ﾃﾞｰﾀ格納ｱﾄﾞﾚｽ ｵﾌｾｯﾄ  */
        *((unsigned *)&dat[10])= FP_SEG(p1);        /*                      ｾｸﾞﾒﾝﾄ */

        do
        {
                printf("PC LINKｽﾃｰﾀｽ読出===>");

                if((err = mew_send()) != 0)
                {
                        printf("ｴﾗｰ.....ｺｰﾄﾞ= %x ¥n", err);
                        exit(-1);
                }
                else
                {
                        if(pcst[14] == 0)
                        {
                                printf("ＰＣ　ＬＩＮＫ停止中¥n");
                        }
                        else
                        {
                                printf("ＰＣ　ＬＩＮＫ起動¥n");
                        }
                }
        }
        while(pcst[14] == 0);

        goto PARRD;
}


/* --------------------------*/
/* ------ ﾘﾝｸｿﾌﾄ呼び出し------*/
/* --------------------------*/
int mew_send(void)
{
        p1 = (char far *)&dat[0];
        inregs.x.dx = FP_OFF(p1);
        segregs.ds = FP_SEG(p1);
        int86x( 0x60, &inregs, &outregs, &segregs);
        return(outregs.x.ax);
}

int mew_send2(void)
```

```
    {
        p1 = (char far *)&rdat[0];
        inregs.x.dx  = FP_OFF(p1);
        segregs.ds = FP_SEG(p1);
        int86x( 0x60, &inregs, &outregs, &segregs);
        return(outregs.x.ax);
    }
```

パソコン機種別変更箇所

ＰＣ／ＡＴ互換機

```
        dat[0] = 0x00;          /* 使用ボード登録及び起動    */
        dat[1] = 0x11;          /* ｺﾏﾝﾄﾞｺｰﾄﾞ=1100H           */
        dat[2] = 0x00;          /* 登録ボードNo.=0           */
        dat[3] = 8;             /* 使用ｾｸﾞﾒﾝﾄｱﾄﾞﾚｽNo.=8      */
        dat[4] = 10;            /* 割り込みNo.=10     */
```

ＦＭＲシリーズ

```
        dat[0] = 0x00;          /* 使用ボード登録及び起動    */
        dat[1] = 0x11;          /* ｺﾏﾝﾄﾞｺｰﾄﾞ=1100H           */
        dat[2] = 0x00;          /* 登録ボードNo.=0           */
        dat[3] = 7;             /* 基板I/OﾎﾟｰﾄｱﾄﾞﾚｽNo.=7     */
        dat[4] = 0x00;          /* ｽﾃｰﾀｽ･ｺﾝﾄﾛｰﾙI/Oﾎﾟｰﾄｱﾄﾞﾚｽ */
        dat[5] = 0x78;          /* =7800H                    */
        dat[6] = 5;             /* 割り込みNo.=5             */
```

3－3 コンピュータリンクの機能コマンド

3-3-1 MEWTOCOL-COM WRITE→READ

コンピュータリンク通信／コマンド書き込み→レスポンス読み出し（コマンドコード&H156x）

機　能

指定バッファに格納されたMEWTOCOL-COMコマンドをリンクボードに書き込み送信要求を行い、受信したレスポンスを指定バッファへ読み出します。

解　説

・MEWTOCOL-COM WRITEコマンドと、同READコマンドを一連で実行します。
・送信できる最大数は、I/Oパラメータテーブルに設定した受信バッファサイズです。
・機能コマンドを実行するには、各種プログラミング言語でI/Oパラメータのセグメントアドレスとオフセットアドレスをds:dxレジスタにセットし、割り込みを実行するユーザープログラムを作成してください。
・コマンド実行後の結果（リターン値）は、axレジスタに格納されます。
　0：正常終了時
　その他：エラーコード（「3－9」参照）

登録送受信要求タイプ、登録送信要求タイプ、登録受信要求タイプの機能コマンドには、同時に使用できる（実行状態にある）コマンド数に制限があります。同時に使用できるコマンド数は、MEWNET-Hリンクソフト（デバイスドライバ）のCONFIG.SYSへの登録時のパラメータスイッチで、0～16まで指定できます。
CONFIG.SYSでの指定がない場合は、同時に実行できるコマンド数は4つまでです。CONFIG.SYSでの指定については、「1-1-2」をお読みください。

■コマンドタイプ

コード	機能
&H1560	送受信完了待ち（PC98のみKEY Break付き）
&H1561	送受信完了待ち（タイムアウト付き）
&H1565	登録送受信要求（タイムアウト付き）

注　意

・&H1560のKEY Breakは、ESCキーのみです。
・&H1561・&H1565は、タイムアウト付きです。
・タイムアウト値は、コントロールレジスタWRITEコマンドにて変更できます。
・コマンドタイプの詳細については、P.90～91ページをお読みください。

文　例

●MS-DOSファンクションコール（int21h）

```
;----- ｺﾝﾋﾟｭｰﾀﾘﾝｸﾑWRITE →READ ----
CL_WR:
        mov     bx,[FILHDL]
        mov     ah,44h
        mov     al,03h
        mov     dx,offset CL_WR
        int     21h
;------- I/Oﾊﾟﾗﾒｰﾀ ------
CL_WR   dw      1561h
        db      0,0
        dw      ofset WBUF,seg WBUF,?,13,0,?
        db      1,0,0,0,0,?,
                3,9,1,3,7,2,3,1
        dw      4 dup(?),ofset RBUF,seg RBUF,
                4007,?,?
WBUF    db      '#RDD0000001000'
RBUF    db      4007 dub(?)
```

●ソフトウェア割り込み（int60h）

```
;----- ｺﾝﾋﾟｭｰﾀﾘﾝｸWRITE →READ ----
CL_WR:
        mov     dx,offset CL_WR
        int     60h
;------- I/Oﾊﾟﾗﾒｰﾀ ------
CL_WR   dw      1561h
        db      0,0
        dw      ofset WBUF,seg WBUF,?,13,0,?
        db      1,0,0,0,0,?,
                3,9,1,3,7,2,3,1
        dw      4 dup(?),ofset RBUF,seg RBUF,
                4007,?,?
WBUF    db      '#RDD0000001000'
RBUF    db      4007 dub(?)
```

I/Oパラメータ

△：設定する［渡す値］
▼：実行後に格納される［得る値］

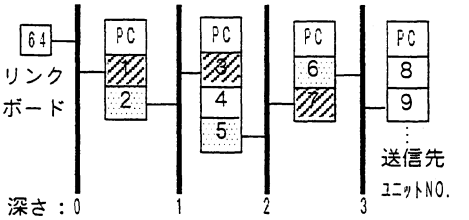
dsレジスタ ←I/Oパラメータの先頭セグメントアドレス
dxレジスタ ←I/Oパラメータの先頭オフセットアドレス

0	△［コマンドコード］&H156x
2	△［登録ボードNo.］0～3　使用するリンクボードの登録ボードNo.を設定します。
	△ 設定値：00固定で使用してください。
4	△［送信バッファアドレス（オフセット）］
6	△［送信バッファアドレス（セグメント）］
8	
A	△［送信データサイズ］送信バッファにセットしたコマンドデータのサイズ。
	2048バイトを越える場合は、複数フレームに分割して送信します。
C	△［フレームタイプ］0＝MEWNET-H用（ヘッダ＝<）／1＝MEWNET-P用（ヘッダ＝%）
	（1フレーム最大2048バイトで送信します）　（1フレーム最大118バイトで送信します）
E	
10	△［制御コード］00：階層リンク不使用　01：階リンク使用
	△［送信先CH］00固定で使用してください。
12	
14	
16	△［深さ］0～3（制御コードで階層リンク使用時にリンクの深さを指定します。）
	△［送信先ユニットNo.］1～64（送信先ユニットNo.）
18	△［伝送経路］制御コードで階層リンク使用時に中継局No.→中継リンクの順で指定します。
1A	
1C	
1E	
20	
22	
24	
26	△［受信バッファアドレス（オフセット）］
28	△［受信バッファアドレス（セグメント）］
2A	△［受信バッファサイズ］
2C	▼［受信データサイズ］受信バッファに格納したレスポンスデータのサイズ（バイト数）。
2E	▼［登録送受信要求完了フラグ］　格納値0：未完了／1：正常終了／その他：エラーコード &H1565実行時のみセットされます。

送信バッファには、コマンドデータ（#を含みBCCの直前まで）をセットしておきます。

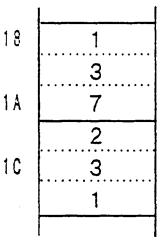
受信バッファには、送信先からのレスポンスデータ（$または!を含みBCCの直前まで）がセットされます。

伝送経路の指定



左図のネットワーク例では
・深さ⇒3
・送信先ユニットNo.⇒9
・中継局No.⇒1、3、7
・中継リンク⇒2、3、1

＊ネットワークの各層のリンクユニットが2台までの場合、深さ0を指定することもできます。



補　足

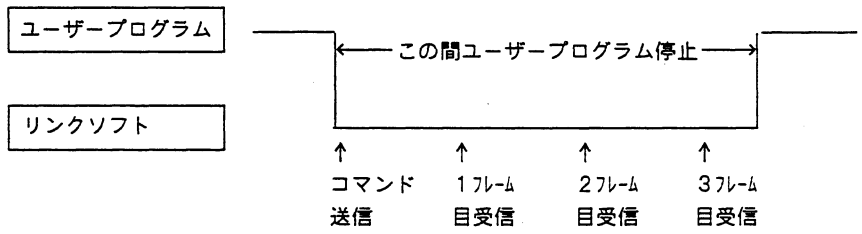
■コマンドタイプ別動作説明

コマンドコード	動作内容	動作詳細
&H１５６０	送受信完了まで待ちます。	・複数フレームの送受信が行えます。 ・ＰＣ９８シリーズのみKEY Break（ESCキー）が使用できます。
&H１５６１	指定タイムまで送受信の完了を待ちます。	・複数フレームの送受信が可能です。 ・タイムアウト指定は、コントロールレジスタWRITEコマンドで行います（初期値は10秒です）。
&H１５６５	登録送受信要求（タイムアウト付き）。	・複数フレームの送受信が可能です。 ・指定データの登録後、送信要求を行い、ユーザプログラムに処理を戻します（完了待ち無し）。 ・登録送受信要求タイプ、登録送信要求タイプ、登録受信要求タイプの機能コマンドには、同時に使用できる（実行状態にある）コマンド数に制限があります。同時に使用できるコマンド数は、MEWNET-Hリンクソフト（デバイスドライバ）のCONFIG.SYSへの登録時のパラメータスイッチで、0～16まで指定できます。CONFIG.SYSでの指定がない場合は、同時に実行できるコマンド数は4つまでです。CONFIG.SYSでの指定については、「1-1-2」をお読みください。 ・この登録は、全送信および、受信完了後、またはエラー発生後解除されます。 ・完了検知は、ユーザプログラム上で、I/Oパラメータの登録送受信要求完了フラグをモニタしてください。 ・登録タイプのコマンドを未完了状態で連続使用する場合、または他のコマンドと併用して使用する場合は、複数のI/Oパラメータテーブルを用意してください（受信バッファも複数用意してください）。

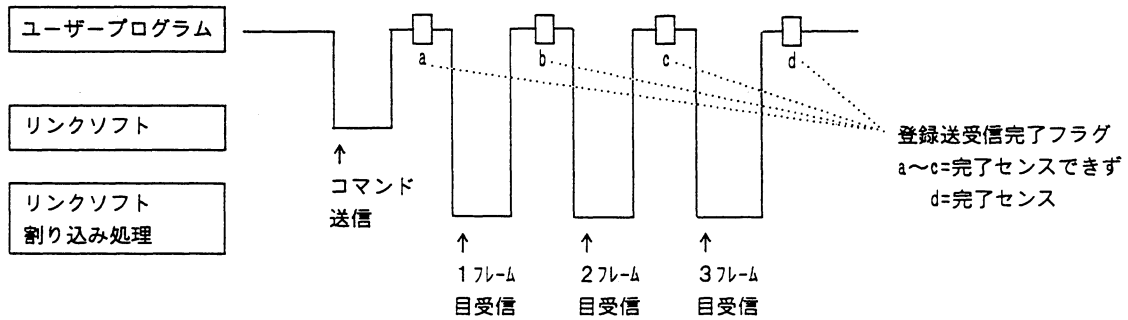
■各コマンドでの処理の違い

A.単一コマンド送信、複数レスポンス（3フレーム）受信時

・完了待ちタイプ（タイムアウト付き含む）　&H1560/&H1561

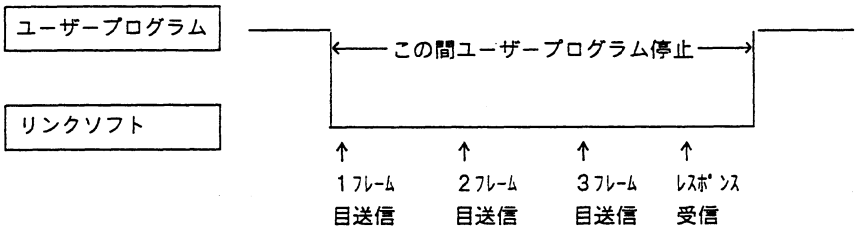


・登録送受信要求タイプ　&H1565

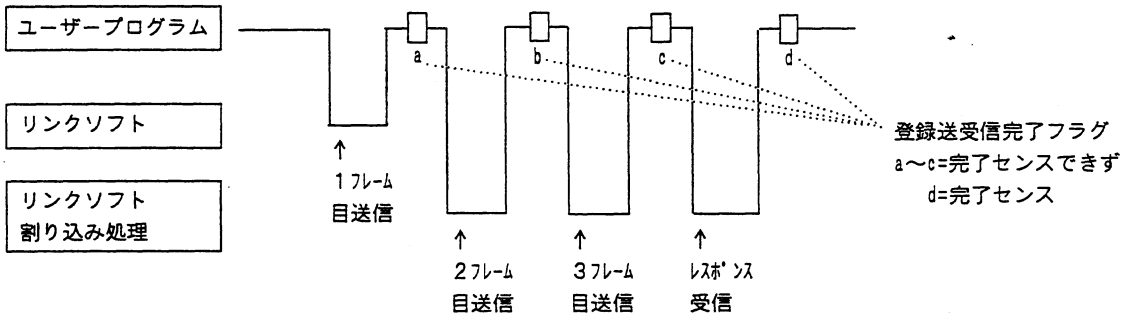


B.複数コマンド（3フレーム）送信、単一レスポンス受信時

・完了待ちタイプ（タイムアウト付き含む）　&H1560/&H1561

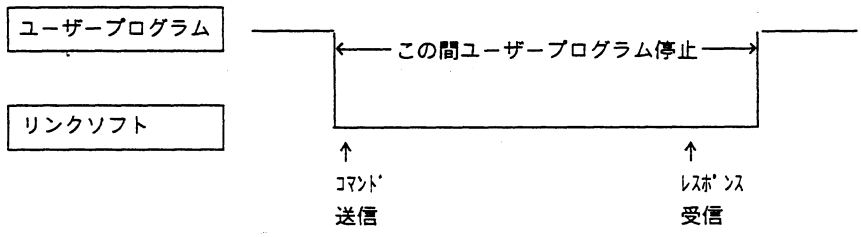


・登録送受信要求タイプ　&H1565

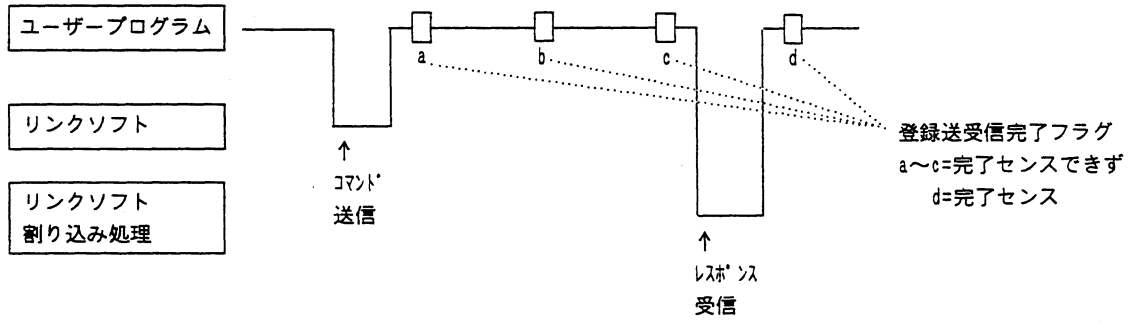


C.単一コマンド送信、単一レスポンス受信時

・完了待ちタイプ（タイムアウト付き含む）　&H1560/&H1561



・登録送受信要求タイプ　&H1565



3-3-2 MEWTOCOL-COM WRITE

コンピュータリンク通信／コマンド書き込み（コマンドコード&H152x）

機　能

指定バッファに格納されたMEWTOCOL-COMコマンドをリンクボードに書き込み、送信要求を行います。

解　説

・機能コマンドを実行するには、各種プログラミング言語でI/Oパラメータのセグメントアドレスとオフセットアドレスをds:dxレジスタにセットし、割り込みを実行するユーザープログラムを作成してください。
・コマンド実行後の結果（リターン値）は、axレジスタに格納されます。
　0：正常終了時
　その他：エラーコード（「3－9」参照）

登録送受信要求タイプ、登録送信要求タイプ、登録受信要求タイプの機能コマンドには、同時に使用できる（実行状態にある）コマンド数に制限があります。同時に使用できるコマンド数は、MEWNET-Hリンクソフト（デバイスドライバ）のCONFIG.SYSへの登録時のパラメータスイッチで、0～16まで指定できます。
CONFIG.SYSでの指定がない場合は、同時に実行できるコマンド数は4つまでです。CONFIG.SYSでの指定については、「1-1-2」をお読みください。

■コマンドタイプ

コード	機能
&H1520	送信完了待ち（PC98のみKEY Break付き）
&H1521	送信完了待ち（タイムアウト付き）
&H1522	送信要求のみ
&H1523	送信完了センス
&H1525	登録送信要求（タイムアウト付き）

注　意

・&H1520のKEY Breakは、[ESC]キーのみです。
・&H1521・&H1525は、タイムアウト付きです。
・タイムアウト値は、コントロールレジスタWRITEコマンドにて変更できます。
・コマンドタイプの詳細については、次ページをお読みください。

文　例

●MS-DOSファンクションコール（int21h）

```
;----- ｺﾝﾋﾟｭｰﾀﾘﾝｸWRITE ----
CL_W:
        mov     bx,[FILHDL]
        mov     ah,44h
        mov     al,03h
        mov     dx,offset CL_W
        int     21h
;------- I/Oﾊﾟﾗﾒｰﾀ ------
CL_W    dw      1521h
        db      0,0
        dw      offset WBUF,seg WBUF,?,13
        db      0,?,?,?,1,0,0,0,0,?,
                3,9,1,3,7,2,3,1
        dw      9 dup(?)
WBUF    db      '#RDD0000001000'
```

●ソフトウェア割り込み（int60）

```
;----- ｺﾝﾋﾟｭｰﾀﾘﾝｸWRITE→READ ----
CL_W:
        mov     dx,offset CL_W
        int     60h
;------- I/Oﾊﾟﾗﾒｰﾀ ------
CL_W    dw      1521h
        db      0,0
        dw      offset WBUF,seg WBUF,?,13
        db      0,?,?,?,1,0,0,0,0,?,
                3,9,1,3,7,2,3,1
        dw      9 dup(?)
WBUF    db      '#RDD0000001000'
```

I/Oパラメータ

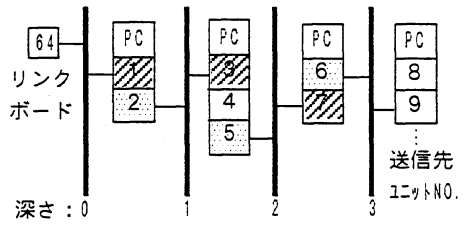
△：設定する［渡す値］
▼：実行後に格納される［得る値］

dsレジスタ ←I/Oパラメータの先頭セグメントアドレス
dxレジスタ ←I/Oパラメータの先頭オフセットアドレス

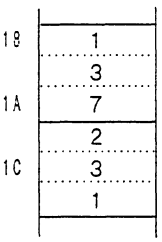
0	△［コマンドコード］&H152x
2	△［登録ボードNo.］0～3（使用するリンクボードの登録ボードNo.）
	△ 設定値：00固定で使用してください。
4	△［送信バッファアドレス（オフセット）］
6	△［送信バッファアドレス（セグメント）］
8	
A	△［送信データサイズ］送信バッファにセットしたコマンドデータのサイズ。
	2048バイトを越える場合は、複数フレームに分割して送信します。
C	△［フレームタイプ］0＝MEWNET-H用（ヘッダ＝<）／1＝MEWNET-P用（ヘッダ＝%）
	（1フレーム最大2048バイトで送信します） （1フレーム最大118バイトで送信します）
E	
10	△［制御コード］00：階層リンク不使用　01：階層リンク使用
	△［送信先CH］00固定で使用してください。
12	
14	
16	△［深さ］0～3（制御コードで階層リンク使用時にリンクの深さを指定します。）
	△［送信先ユニットNo.］1～64（送信先UNIT No.）
18	△［伝送経路］制御コードで階層リンク使用時に中継局No.→中継リンクの順で指定します。
1A	
1C	
1E	
20	
22	
24	
26	
28	
2A	
2C	
2E	

△［送信バッファアドレス（オフセット）］
△［送信バッファアドレス（セグメント）］
　└ 送信バッファには、コマンドデータ（#を含みBCCの直前まで）をセットしておきます。

伝送経路の指定



左図のネットワーク例では
・深さ⇒3
・送信先ユニットNo.⇒9
・中継局No.⇒1、3、7
・中継リンク⇒2、3、1
＊ネットワークの各層のリンクユニットが2台までの場合、深さ0を指定することもできます。



▨中継局No.（UNIT No.を指定）
　中継リンク（ＰＣから何ユニット目かを指定）

▼［登録送信要求完了フラグ］0：未完了／1：正常終了／その他：エラーコード
&H1525のみセットされます。

補　足

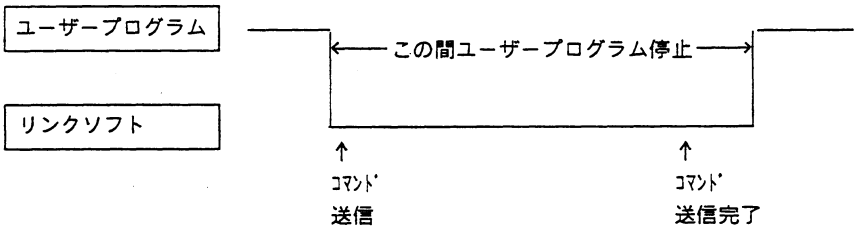
■コマンドタイプ別動作説明

コマンドコード	動作内容	動作詳細
&H1520	送信完了まで待ちます。	・複数フレームの送信ができます。 ・ＰＣ９８シリーズのみKEY Break（ESCキー）が有効です。
&H1521	指定タイムまで送信完了を待ちます。	・複数フレームの送信ができます。 ・タイムアウト指定は、コントロールレジスタWRITEコマンドで設定します（初期値は10秒です）。
&H1522	送信要求のみ。	・複数フレームの送信はできません。 ・送信要求だけをし、完了検知せずにユーザーソフトへ戻ります。
&H1523	送信完了センス。	・データの送信はしません。 ・コマンド&H1522で要求した送信の完了検知に使用します。 ・登録送信要求&H1525の完了検知には使用できません。
&H1525	登録送信要求（タイムアウト付き）。	・複数フレームの送信ができます。 ・指定データの登録後、送信要求を行い、ユーザプログラムに処理を戻します（完了待ち無し。） ・登録送受信要求タイプ、登録送信要求タイプ、登録受信要求タイプの機能コマンドには、同時に使用できる（実行状態にある）コマンド数に制限があります。同時に使用できるコマンド数は、MEWNET-Hリンクソフト（デバイスドライバ）のCONFIG.SYSへの登録時のパラメータスイッチで、0～16まで指定できます。CONFIG.SYSでの指定がない場合は、同時に実行できるコマンド数は4つまでです。CONFIG.SYSでの指定については、「1-1-2」をお読みください。 ・この登録は、全送信完了後、またはエラー発生後解除されます。 ・完了検知は、ユーザプログラム上で、I/Oパラメータの登録送信要求完了フラグをモニタしてください。 ・登録タイプのコマンドを未完了状態で連続使用する場合、または他のコマンドと併用して使用する場合は、複数のI/Oパラメータテーブルを用意してください（受信バッファも複数用意してください）。

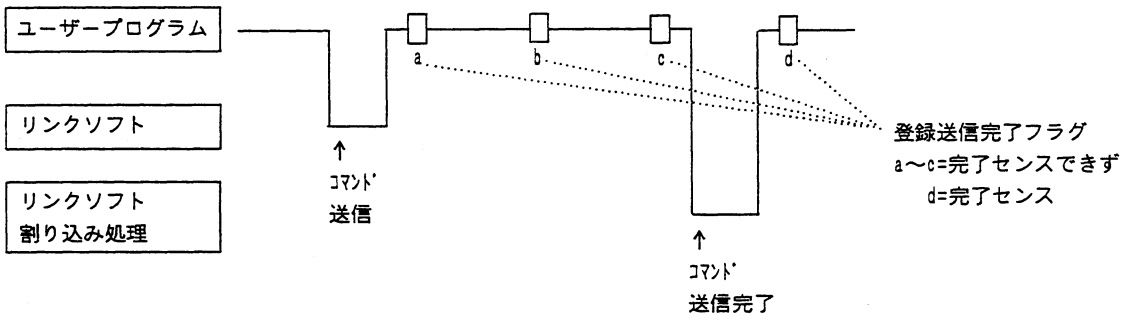
■各コマンドでの処理の違い

A.単一コマンド送信時

・完了待ちタイプ（タイムアウト付き含む）　&H1520/&H1521

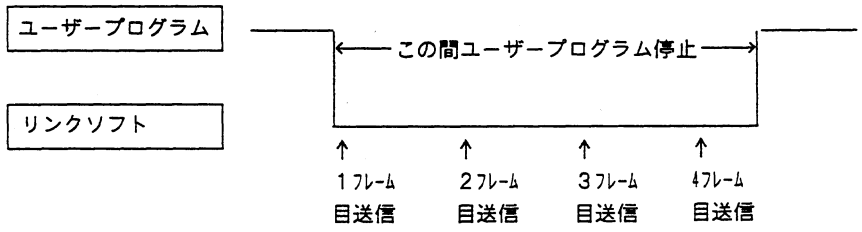


・登録送信要求タイプ　&H1525

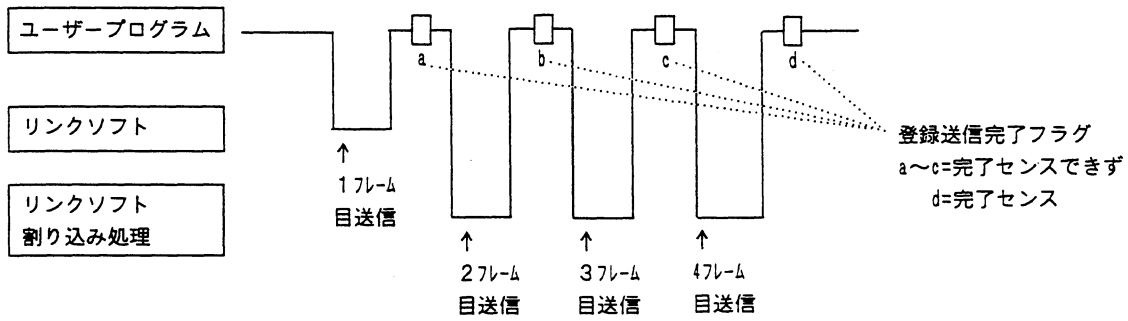


B.複数コマンド（4フレーム）送信時

・完了待ちタイプ（タイムアウト付き含む）　&H1520/1521



・登録送信要求タイプ　&H1525



3-3-3 MEWTOCOL-COM READ

コンピュータリンク通信／コマンド読み出し（コマンドコード&H142x）

機　能

リンクボードにMEWTOCOL-COMデータの受信要求を発行し、受信データを指定バッファへ格納します。

解　説

・機能コマンドを実行するには、各種プログラミング言語でI/Oパラメータのセグメントアドレスとオフセットアドレスをds:dxレジスタにセットし、割り込みを実行するユーザープログラムを作成してください。
・コマンド実行後の結果（リターン値）は、axレジスタに格納されます。
　0：正常終了時
　その他：エラーコード（「3－9」参照）

登録送受信要求タイプ、登録送信要求タイプ、登録受信要求タイプの機能コマンドには、同時に使用できる（実行状態にある）コマンド数に制限があります。同時に使用できるコマンド数は、MEWNET-Hリンクソフト（デバイスドライバ）のCONFIG.SYSへの登録時のパラメータスイッチで、0～16まで指定できます。
CONFIG.SYSでの指定がない場合は、同時に実行できるコマンド数は4つまでです。CONFIG.SYSでの指定については、「1-1-2」をお読みください。

■コマンドタイプ

コード	機能
&H1420	受信完了待ち（PC98のみKey Break付き）
&H1421	受信完了待ち（タイムアウト付き）
&H1422	受信完了センス （センス時データ読み出し付き）
&H1423	受信バッファクリア（空読み）
&H1444	登録受信要求（タイムアウト無し）
&H1425	登録受信要求（タイムアウト付き）

注　意

・&H1420のKEY Breakは、ESCキーのみです。
・&H1421・&H1425は、タイムアウト付きです。
　タイムアウト値は、コントロールレジスタWRITEコマンドにて変更できます。
・登録受信要求（タイムアウト無し）の取消には、登録受信タイムアウト無しタイプ解除（&H1460）を実行します。（「3－7」参照）
・コマンドタイプの詳細については、次ページをお読みください。

文　例

●MS-DOSファンクションコール（int21h）

```
;----- ｺﾝﾋﾟｭｰﾀﾘﾝｸREAD ----
CL_R:
        mov     bx, [FILHDL]
        mov     ah, 44h
        mov     al, 03h
        mov     dx, offset CL_R
        int     21h
;------- I/Oﾊﾟﾗﾒｰﾀ ------
CL_R    dw      1421h
        db      0, 0
        dw      offset RBUF, seg RBUF, 256,
        db      0, ?, ?, ?
        db      0, 0, 0, 0, 0, ?, 0, ?
        dw      12 dup(?)
RBUF    db      255 dub(?)
```

●ソフトウェア割り込み（int60h）

```
;----- ｺﾝﾋﾟｭｰﾀﾘﾝｸREAD ----
CL_R:
        mov     dx, offset CL_R
        int     60h
;------- I/Oﾊﾟﾗﾒｰﾀ ------
CL_R    dw      1421h
        db      0, 0
        dw      offset RBUF, seg RBUF, 256,
        db      0, ?, ?, ?
        db      0, 0, 0, 0, 0, ?, 0, ?
        dw      12 dup(?)
RBUF    db      255 dub(?)
```

I/Oパラメータ

△：設定する［渡す値］
▼：実行後に格納される［得る値］

dsレジスタ ←I/Oパラメータの先頭セグメントアドレス
dxレジスタ ←I/Oパラメータの先頭オフセットアドレス

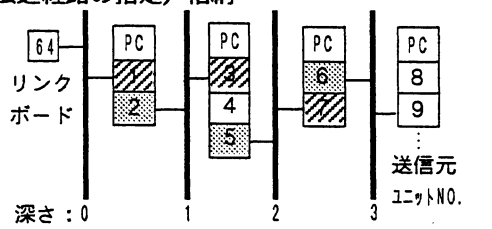
0	△［コマンドコード］&H142x
2	△［登録ボードNo.］0～3（使用するリンクボードの登録ボードNo.）
	△ 設定値：00固定で使用してください。
4	△［受信バッファアドレス（オフセット］） 受信バッファには、レスポンスデータ（$または!を含みBCCの直前まで）がセットされます。
6	△［受信バッファアドレス（セグメント）］
8	△［受信バッファサイズ］バイト数を指定します。
A	▼［受信データサイズ］受信バッファに格納したレスポンスデータのサイズ（バイト数）。
C	▼［フレームタイプ］0＝MEWNET-H用（ヘッダ＝<）／1＝MEWNET-P用（ヘッダ＝%）
E	
10	△［制御コード］送信元指定用 ▼［送信元CH］受信完了時に送信元CHが格納されます。
12	
14	
16	▼△［深さ］0～3（制御コードで階層リンク使用時のリンクの深さが指定／格納されます。）
	▼△［送信元ユニットNo.］1～64（送信元指定時にはUNIT No.をセットします。）
18	▼△［伝送経路］制御コードで階層リンク使用時に中継局No.→中継リンクの順で指定／格納されます。
1A	
1C	
1E	
20	
22	
24	
26	
28	
2A	
2C	▼［登録No.］登録受信要求（タイムアウト無し）タイプを解除するときに使用する登録No.
2E	▼［登録受信要求完了フラグ］0：未完了／1：正常終了／その他：エラーコード &H1425実行時のみセットされます。

■制御コード

	階層リンク	
	使用する	使用しない
送信元指定する	9	8
送信元指定しない	1	0

＊送信元を指定した場合、指定局からのデータのみREADします。

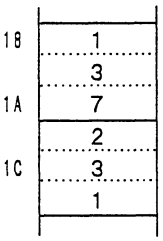
伝送経路の指定／格納



▨中継局No.（UNIT No.を指定）
▩中継リンク（ＰＣから何ユニット目かを指定）

左図のネットワーク例では
・深さ⇒3
・送信元ユニットNo.⇒9
・中継局No.⇒1、3、7
・中継リンク⇒2、3、1

＊ネットワークの各層のリンクユニットが2台までの場合、深さ0を指定することもできます。



上記送信元からのデータのみREADする場合
制御コード　=9
深さ　　=3
送信元ユニット　=9
伝送経路　中継局No.　=1、3、7
　　中継リンク　=2、3、1

補　足

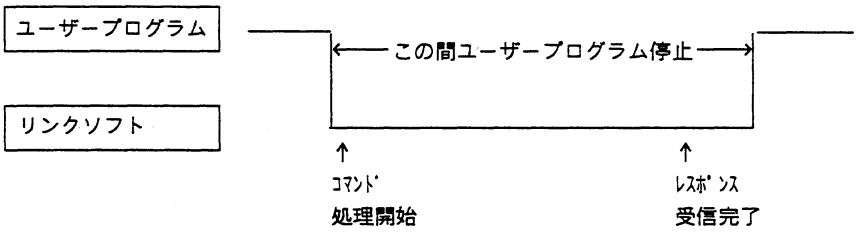
■コマンドタイプ別動作説明

コマンドコード	動作内容	動作詳細
&H1420	受信完了まで待ちます。	・複数フレームの受信ができます。 ・ＰＣ９８シリーズのみKEY Break（ESCキー）が有効です。
&H1421	指定タイムまで受信完了を待ちます。	・複数フレームの受信ができます。 ・タイムアウト指定は、コントロールレジスタWRITEコマンドで設定します（初期値は10秒です）。
&H1422	受信完了センス。	・複数フレームの受信はできません。 ・受信状態をチェックし戻ります。受信有り時データをREADします。 ・登録受信要求中は、使用できません。
&H1423	受信バッファクリア。	
&H1424 &H1425	登録受信要求。 タイムアウト付きおよびタイムアウト無し。	・複数フレームの受信ができます。 ・実行時に指定データの受信があれば、データをREADし、ユーザプログラムに処理を戻します（完了待ち無し）。このとき、READフレームが完結していなければ、READ指定内容を登録してユーザプログラムへ戻ります。 ・実行時に、指定データの受信が無ければ、指定内容を登録し、ユーザプログラムへ戻ります。 ・登録後、指定データの受信があれば、自動的にREADをします。 ・この登録は、指定データを全て受信完了後、またはエラー発生後解除されます。 ・登録送受信要求タイプ、登録送信要求タイプ、登録受信要求タイプの機能コマンドには、同時に使用できる（実行状態にある）コマンド数に制限があります。同時に使用できるコマンド数は、MEWNET-Hリンクソフト（デバイスドライバ）のCONFIG.SYSへの登録時のパラメータスイッチで、0～16まで指定できます。CONFIG.SYSでの指定がない場合は、同時に実行できるコマンド数は4つまでです。CONFIG.SYSでの指定については、「1-1-2」をお読みください。 ・完了検知は、ユーザプログラム上で、I/Oパラメータの登録受信要求完了フラグをモニタしてください。登録受信要求の完了をもって、登録が解除されます。 ・登録タイプのコマンドを未完了状態で連続使用する場合、または他のコマンドと併用して使用する場合は、複数のI/Oパラメータテーブルを用意してください（受信バッファも複数用意してください）。

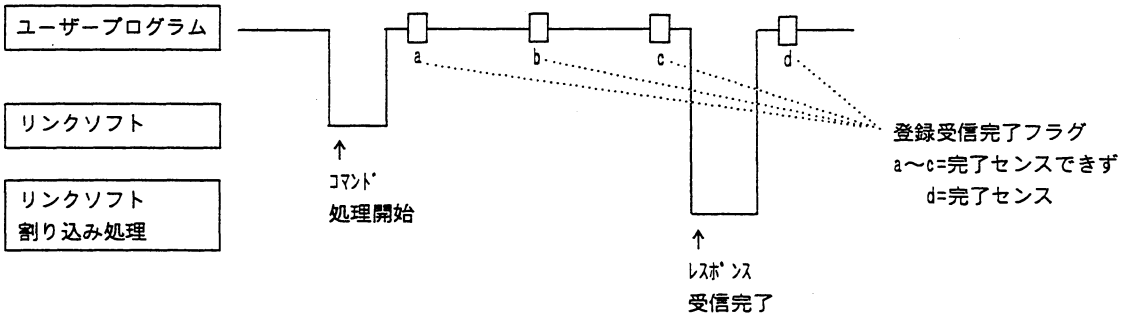
■各コマンドでの処理の違い

A.単一レスポンス受信時

・完了待ちタイプ（タイムアウト付き含む）　&H1420/&H1421

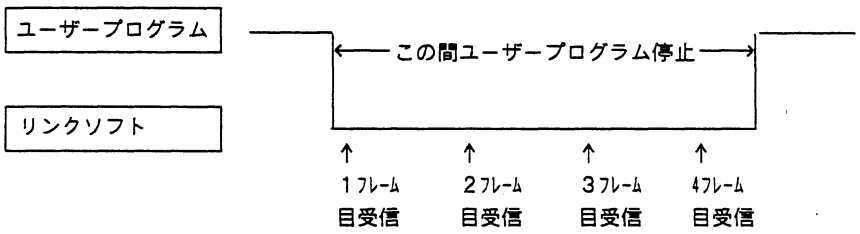


・登録受信要求タイプ　&H1424/&H1425

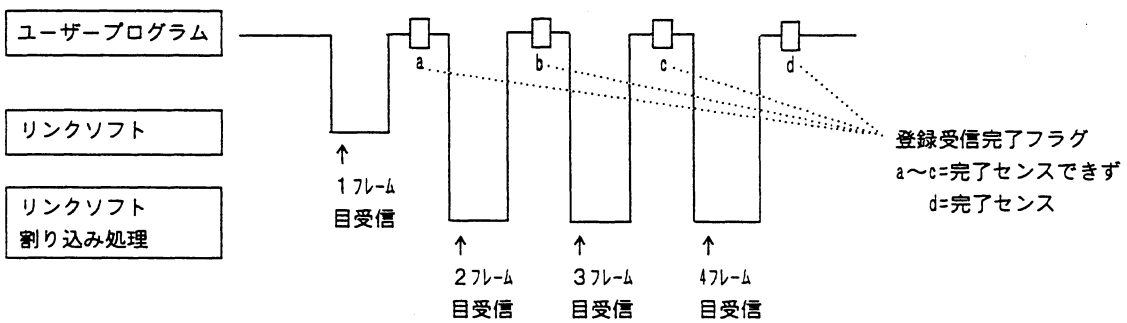


B.複数レスポンス（4フレーム）受信時

・完了待ちタイプ（タイムアウト付き含む）　&H1420/&H1421



・登録受信要求タイプ　&H1424/&H1425



3-3-4 コンピュータリンクのプログラム例

アセンブラプログラム（MS-DOSシステムコールint21）

```
; ********************************************************
; *              MEWTOCOL-COM                            *
; *                  <AMWCM.ASM>                         *
; ********************************************************
; *                  PC98シリーズ                        *
; *                                                      *
; *    1番機のPCのDT0から10ワード分読み出す              *
; *    単一フレーム:階層無し                             *
; *                                                      *
; *    使用コマンド：WRITE=1521H(ﾀｲﾑｱｳﾄ付き)              *
; *                  READ =1421H(ﾀｲﾑｱｳﾄ付き)            *
; ********************************************************
                                        ;
CODE    SEGMENT
        ASSUME CS:CODE,DS:CODE,ES:CODE,SS:CODE


CR              EQU     0DH             ; CRｺｰﾄﾞ
LF              EQU     0AH             ; LFｺｰﾄﾞ
                                        ;
                                        ;
START:
                MOV     AX,CS
                MOV     DS,AX
                MOV     ES,AX
                MOV     SS,AX
                MOV     AX,OFFSET STPT
                MOV     SP,AX
                                        ;
;  ---------------------------------------------
;                 初 期 設 定
;  ---------------------------------------------
INI:
                MOV     DX,OFFSET MSG1          ; 初期設定メッセージ表示
                CALL    MESSAGE
                MOV     DX,OFFSET INIT          ; 初期設定用I/Oﾊﾟﾗﾒｰﾀ ｱﾄﾞﾚｽ設定
                CALL    COMMAND                 ; ﾘﾝｸｿﾌﾄ呼出
                JNB     INI_OK
                JMP     DOS                     ; ｴﾗｰ時 MS-DOSへ戻る
                                        ;
INI_OK:
                MOV     DX,OFFSET MSG1_OK       ; 初期設定ＯＫﾒｯｾｰｼﾞ 表示
                CALL    MESSAGE
                                        ;
;  ---------------------------------------------
;        ボ ー ド 登 録 ・ 起 動
;  ---------------------------------------------
                                        ;
                MOV     DX,OFFSET MSG4          ; ﾎﾞｰﾄﾞ 登録･起動メッセージ表示
                CALL    MESSAGE
                MOV     DX,OFFSET BOARD         ; ﾎﾞｰﾄﾞ 登録用I/Oﾊﾟﾗﾒｰﾀ ｱﾄﾞﾚｽ
                CALL    COMMAND                 ; ﾘﾝｸｿﾌﾄ呼出
                JNB     BDST_OK
                JMP     DOS                     ; ｴﾗｰ時 MS-DOSへ戻る
                                        ;
BDST_OK:
                MOV     DX,OFFSET MSG4_OK       ; ﾎﾞｰﾄﾞ 登録&起動OKﾒｯｾｰｼﾞ 表示
                CALL    MESSAGE

;  ---------------------------------------------
;          MEWTOCOL-COM送信
;  ---------------------------------------------
LP:
                MOV     DX,OFFSET TX_MSG        ; 送信コマンド内容表示
                CALL    MESSAGE
                MOV     DX,OFFSET MEWWT         ; 送信用I/Oﾊﾟﾗﾒｰﾀ ｱﾄﾞﾚｽ
                CALL    COMMAND
                JB      LP                      ; ｴﾗｰ時 再送へ
                                        ;
                MOV     DX,OFFSET TX_OK_MSG     ; 送信ＯＫﾒｯｾｰｼﾞ 表示
                CALL    MESSAGE
                                        ;
;  ---------------------------------------------
```

```
;           ＭＥＷＴＯＣＯＬ－ＣＯＭ受信
; ----------------------------------------
                                        ;
                MOV     DX,OFFSET  RX_MSG       ; ﾚｽﾎﾟﾝｽ読出ﾒｯｾｰｼﾞ 表示
                CALL    MESSAGE
                MOV     DX,OFFSET  MEWRD        ; 受信用I/Oﾊﾟﾗﾒｰﾀ ｱﾄﾞﾚｽ
                CALL    COMMAND
                JB      LP                      ; ｴﾗｰ時 再送へ
                                        ;
                MOV     DX,OFFSET  RX_OK_MSG    ; ﾚｽﾎﾟﾝｽ読出ＯＫﾒｯｾｰｼﾞ 表示
                CALL    MESSAGE
                                        ;
                CALL    RXDATA_DSP              ; 読出内容表示
                                        ;
                JMP     LP                      ; MEWTOCOL-COM送信へ
                                        ;
                                        ;
; ========================================
; =           ＭＳ－ＤＯＳへ戻る処理           =
; ========================================
                                        ;
DOS:
                MOV     AH,4CH
                INT     21H
                                        ;
                                        ;
; ========================================
; =          メッセージの画面表示処理           =
; ========================================
                                        ;
MESSAGE:
                MOV     AH,9                    ; DXｱﾄﾞﾚｽから'$'迄を画面表示
                INT     21H
                RET
                                        ;
                                        ;
; ========================================
; =           リンクソフト呼出処理             =
; ========================================
                                        ;
COMMAND:
                INT     60H                     ; ｼｽﾃﾑｺｰﾙ
                CMP     AX,0                    ; AX>0時ｴﾗｰ
                JE      CMD2                    ;
                                        ;
                CALL    BI2ASC                  ; ｴﾗｰｺｰﾄﾞ ASCII変換
                MOV     DX,OFFSET  MSG3         ; ｴﾗｰﾒｯｾｰｼﾞ 画面表示
                CALL    MESSAGE
                STC
CMD2:
                RET
                                        ;
                                        ;
; ========================================
; =       エラーコードのＡＳＣＩＩ変換        =
; ========================================
                                        ;
BI2ASC:
                MOV     BL,AH
                MOV     AH,AL
                MOV     BH,BL
                AND     AL,0FH
                AND     BL,0FH
                MOV     CL,4
                SHR     AH,CL
                SHR     BH,CL
                ADD     BX,3030H
                ADD     AX,3030H
                CMP     AL,3AH
                JB      B0
                ADD     AL,7
B0:
                CMP     AH,3AH
                JB      B1
                ADD     AH,7
B1:
```

```
                CMP     BL,3AH
                JB      B2
                ADD     BL,7
B2:
                CMP     BH,3AH
                JB      B3
                ADD     BH,7
B3:
                MOV     [MSG3_1+0],BH
                MOV     [MSG3_1+1],BL
                MOV     [MSG3_1+2],AH
                MOV     [MSG3_1+3],AL
                RET
                                        ;
                                        ;
; ========================================
; =       ＢＩＮ==>ＡＳＣＩＩ変換       =
; ========================================
                                        ;
BIN_ASC:                                ; IN...AL=BINｺｰﾄﾞ
                PUSH    SI              ; OUT..AX=ASCIIｺｰﾄﾞ
                PUSH    BX
                PUSH    DX
                                        ;
                MOV     BL,AL
                AND     AL,0F0H         ; 上位４ビット
                SHR     AL,1
                SHR     AL,1
                SHR     AL,1
                SHR     AL,1
                LEA     SI,BINTOASC_TB  ; 変換テーブル
                MOV     AH,0
                ADD     SI,AX
                MOV     DH,BYTE PTR CS:[SI]
                                        ;
                MOV     AL,BL
                AND     AL,0FH          ; 下位４ビット
                LEA     SI,BINTOASC_TB  ; 変換テーブル
                MOV     AH,0
                ADD     SI,AX
                MOV     AL,BYTE PTR CS:[SI]
                MOV     AH,DH
                XCHG    AL,AH
                                        ;
                POP     DX
                POP     BX
                POP     SI
                RET
                                        ;
                                        ;
; ----------------------------------------
;       ＨＥＸ→ＡＳＣＩＩ　変換テーブル
; ----------------------------------------
                                        ;
BINTOASC_TB     DB      "0123456789ABCDEF"
                                        ;
                                        ;
; ========================================
; =              受信データ表示          =
; ========================================
                                        ;
RXDATA_DSP:
                CMP     BYTE PTR CS:[JBUF],24H      ; ﾚｽﾎﾟﾝｽOK?
                JE      RX_D_OK         ; YES===>RX_D_OK へ
                                        ; ...NO
                MOV     DX,OFFSET RX_BD_MSG ; ｴﾗｰﾚｽﾎﾟﾝｽ表示
                CALL    MESSAGE
                LEA     DI,JBUF
                MOV     AX,CS:[DI+1]
                CALL    BIN_DSP
                MOV     DX,OFFSET CRLF  ; 改行復帰
                CALL    MESSAGE
                RET
RX_D_OK:
                MOV     DX,OFFSET DSP_RXSIZE ; 受信ｻｲｽﾞ表示
                CALL    MESSAGE
```

```
                MOV     BX,CS:[RXSIZE]
                MOV     AL,BH
                CALL    BIN_ASC
                CALL    BIN_DSP
                MOV     AL,BL
                CALL    BIN_ASC
                CALL    BIN_DSP
                MOV     DX,OFFSET CRLF
                CALL    MESSAGE
                                        ;
                MOV     DX,OFFSET DSP_FRMTP     ; 受信ﾌﾚｰﾑﾀｲﾌﾟ表示
                CALL    MESSAGE
                MOV     AL,CS:[FRMTP]
                CALL    BIN_ASC
                CALL    BIN_DSP
                MOV     DX,OFFSET CRLF
                CALL    MESSAGE
                                        ;
                MOV     DX,OFFSET DSP_TXNO      ; 送信元No.表示
                CALL    MESSAGE
                MOV     AL,CS:[TXNO]
                CALL    BIN_ASC
                CALL    BIN_DSP
                MOV     DX,OFFSET CRLF
                CALL    MESSAGE
                                        ;
                XOR     CX,CX
                LEA     DI,JBUF+3
RX_D_LP:
                MOV     DX,OFFSET DSP_DT        ; "DT"表示
                CALL    MESSAGE
                MOV     AL,CL
                CALL    BIN_ASC
                CALL    BIN_DSP                 ; DTxxのxx表示
                MOV     DX,OFFSET DSP_E
                CALL    MESSAGE                 ; "="表示
                MOV     AX,DS:[DI+2]
                CALL    BIN_DSP                 ; ﾃﾞｰﾀ表示
                MOV     AX,DS:[DI]
                CALL    BIN_DSP                 ; ﾃﾞｰﾀ表示
                                                ;
                MOV     DX,OFFSET CRLF          ; 改行復帰
                CALL    MESSAGE
                                                ;
                ADD     DI,4                    ; ﾎﾟｲﾝﾀｰ更新
                INC     CL
                CMP     CL,10
                JB      RX_D_LP
                                        ;
                MOV     DX,OFFSET CRLF
                CALL    MESSAGE
                RET
                                        ;
                                        ;
BIN_DSP:                                ; IN...AX=表示文字ｺｰﾄﾞ
                PUSH    AX
                MOV     DL,AL
                MOV     AH,2
                INT     21H
                POP     AX
                MOV     DL,AH
                MOV     AH,2
                INT     21H
                RET
                                        ;
                                        ;
; ============================================
; =            I／Oパラメータ                =
; ============================================
                                        ; = 初期設定 =
INIT            DW      1000H           ; 初期設定ｺﾏﾝﾄﾞｺｰﾄﾞ
                DB      0               ; ｱﾌﾟﾘｹｰｼｮﾝ･ﾘﾌﾄﾓｰﾄﾞ
                DB      1               ; 使用ﾎﾞｰﾄﾞ枚数
                                        ;
```

```
                                        ; = ﾎﾞｰﾄﾞ登録･起動 =
BOARD       DW      1100H               ; ﾎﾞｰﾄﾞ登録･起動ｺﾏﾝﾄﾞｺｰﾄﾞ
            DB      0                   ; ﾎﾞｰﾄﾞ登録No.
            DB      27                  ; 使用ｾｸﾞﾒﾝﾄｱﾄﾞﾚｽNo.27(D0000H)
            DB      0                   ; 割り込みNo.
            DB      0
                                        ;
                                        ; = MEWTOCOL-COM送信 =
MEWWT       DW      1521H               ; MEWTOCOL-COM送信ｺﾏﾝﾄﾞｺｰﾄﾞ
            DB      0                   ; 登録ﾎﾞｰﾄﾞNO.
            DB      0
            DW      OFFSET SBUF         ; 送信ﾊﾞｯﾌｧOFFSET
MEWWT2      DW      SEG SBUF
            DW      0
            DW      14                  ; 送信ﾃﾞｰﾀ･ｻｲｽﾞ
            DB      0                   ; 送信ﾌﾚｰﾑﾀｲﾌﾟ
            DB      0,0,0
            DB      0                   ; 制御ｺｰﾄﾞ
            DB      0                   ; 送信先CH
            DB      0                   ; 送信先ID
            DB      0                   ; 送信元CH
            DB      0                   ; 送信元ID
            DB      0
            DB      0                   ; Depth
            DB      1                   ; 送信先NO.
            DB      14 DUP(0)
            DW      4 DUP(0)
            DW      0                   ; 登録ﾌﾗｸﾞ用
                                        ;
                                        ;
                                        ; = MEWTOCOL-COM受信 =
MEWRD       DW      1421H               ; MEWTOCOL-COM受信ｺﾏﾝﾄﾞｺｰﾄﾞ
            DB      0                   ; 登録ﾎﾞｰﾄﾞNO.
            DB      0
            DW      OFFSET JBUF         ; 受信ﾊﾞｯﾌｧOFFSET
MEWRD2      DW      SEG JBUF
            DW      2048                ; 受信ﾊﾞｯﾌｧｻｲｽﾞ
RXSIZE      DW      0                   ; 受信ﾃﾞｰﾀｻｲｽﾞ用
FRMTP       DB      0                   ; 受信ﾌﾚｰﾑﾀｲﾌﾟ
            DB      0,0,0
            DB      0                   ; 制御ｺｰﾄﾞ
            DB      0                   ; 送信元CH
            DB      0                   ; 送信元ID
            DB      0                   ; 受信CH
            DB      0                   ; 受信ID
            DB      0
            DB      0                   ; Depth
TXNO        DB      0                   ; 送信元NO.
            DB      14 DUP(0)           ; 受信経路用
            DW      4 DUP(0)
            DW      0                   ; 登録ﾌﾗｸﾞ用
                                        ;
                                        ;
;  ============================================
;  =          メッセージ ／ バッファ領域          =
;  ============================================
                                        ;
MSG1        DB      '初期設定===>$'
MSG1_OK     DB      'ＯＫ！'
            DB      CR,LF,'$'

MSG3        DB      'ｴﾗｰ.....ｺｰﾄﾞ='
MSG3_1      DB      0,0,0,0
            DB      CR,LF,'$'

MSG4        DB      '使用ボード登録 & 起動===>$'
MSG4_OK     DB      'ＯＫ！'
            DB      CR,LF,'$'

TX_MSG      DB      'ｺﾏﾝﾄﾞ=#RDD0000000009',CR,LF
            DB      'ﾊﾟｹｯﾄ送信===>$'
TX_OK_MSG   DB      'ＯＫ！',CR,LF,'$'

RX_MSG      DB      'ﾚｽﾎﾟﾝｽ読出===>$'
RX_OK_MSG   DB      'ＯＫ！',CR,LF,'$'
```

パソコン機種別変更箇所
（左記はＰＣ９８シリーズの場合）

```
RX_BD_MSG       DB      'ｴﾗｰﾚｽﾎﾟﾝｽ ｺｰﾄﾞ=$'

DSP_RXSIZE      DB      '受信ﾃﾞｰﾀｻｲｽﾞ(Hex)=$'
DSP_FRMTP       DB      '受信ﾌﾚｰﾑﾀｲﾌﾟ      =$'
DSP_TXNO        DB      '送信元No.         =$'

DSP_DT          DB      'DT$'
DSP_E           DB      '=$'
CRLF            DB      CR,LF,'$'

SBUF            DB      '#RDD0000000009'            ; 送信ﾃﾞｰﾀ

JBUF            DB      2048 DUP(0)                 ; 受信ﾊﾞｯﾌｧ

STACK           DW      256 DUP(0)                  ; ｽﾀｯｸ領域
STPT            DB      0


                CODE    ENDS
                END     START
```

```
; ********************************************************
; *    MEWTOCOL-COM WRITE→READ                           *
; *                  <AMWCM2.ASM>                        *
; ********************************************************
; *                          PC98シリーズ                *
; *                                                      *
; *     1番機のPCのDT0から10ワード分読み出す             *
; *     単一フレーム:階層無し                            *
; *                                                      *
; *     使用コマンド:1565H(登録送受信要求ﾀｲﾑｱｳﾄ付き)      *
; ********************************************************
                                        ;
CODE    SEGMENT
        ASSUME CS:CODE,DS:CODE,ES:CODE,SS:CODE


CR              EQU     0DH             ; CRｺｰﾄﾞ
LF              EQU     0AH             ; LFｺｰﾄﾞ
                                        ;
                                        ;
START:
                MOV     AX,CS
                MOV     DS,AX
                MOV     ES,AX
                MOV     SS,AX
                MOV     AX,OFFSET STPT
                MOV     SP,AX
                                        ;
; ------------------------------------------
;               初 期 設 定
; ------------------------------------------
INI:
                MOV     DX,OFFSET MSG1      ; 初期設定ﾒｯｾｰｼﾞ表示
                CALL    MESSAGE
                MOV     DX,OFFSET INIT      ; 初期設定用I/Oﾊﾟﾗﾒｰﾀ ｱﾄﾞﾚｽ設定
                CALL    COMMAND             ; ﾘﾝｸｿﾌﾄ呼出
                JNB     INI_OK
                JMP     DOS                 ; ｴﾗｰ時 MS-DOSへ戻る
                                        ;
INI_OK:
                MOV     DX,OFFSET MSG1_OK   ; 初期設定OKﾒｯｾｰｼﾞ表示
                CALL    MESSAGE
                                        ;
; ------------------------------------------
```

```
;        ボ ー ド 登 録 ・ 起 動
; ---------------------------------------------
                                        ;
                MOV     DX,OFFSET MSG4          ; ﾎﾞｰﾄﾞ登録･起動メッセージ表示
                CALL    MESSAGE
                MOV     DX,OFFSET BOARD         ; ﾎﾞｰﾄﾞ登録用I/Oﾊﾟﾗﾒｰﾀ ｱﾄﾞﾚｽ
                CALL    COMMAND                 ; ﾘﾝｸｿﾌﾄ呼出
                JNB     BDST_OK
                JMP     DOS                     ; ｴﾗｰ時 MS-DOSへ戻る
                                        ;
BDST_OK:
                MOV     DX,OFFSET MSG4_OK       ; ﾎﾞｰﾄﾞ登録&起動OKﾒｯｾｰｼﾞ 表示
                CALL    MESSAGE

; ---------------------------------------------
;         MEWTOCOL-COM WRITE→READ登録
; ---------------------------------------------
LP:
                MOV     DX,OFFSET TX_MSG        ; 送信コマンド内容表示
                CALL    MESSAGE
                MOV     DX,OFFSET MEWWT         ; 送信用I/Oﾊﾟﾗﾒｰﾀ ｱﾄﾞﾚｽ
                CALL    COMMAND
                JB      LP                      ; ｴﾗｰ時 再登録へ
                                        ;
                MOV     DX,OFFSET TX_OK_MSG     ; 送信ＯＫﾒｯｾｰｼﾞ 表示
                CALL    MESSAGE
                                        ;
; ---------------------------------------------
;        MEWTOCOL-COM WRITE→READ完了ﾁｪｯｸ
; ---------------------------------------------
                                        ;
CKLP:
                MOV     BX,CS:[ENDF]
                CMP     BX,1                    ; 正常終了?
                JNE     CK_10                   ; NO ===>CK_10 へ
                                        ; ...YES
                CALL    RXDATA_DSP              ; 読出内容表示
                JMP     LP                      ; ｴﾗｰ時 登録へ
                                        ;
CK_10:
                CMP     BX,0                    ; 未完状態?
                JE      CKLP                    ; YES===>CK_LP へ
                                        ; ...NO
                MOV     DX,OFFSET RX_BD2_MSG    ; 異常発生表示
                CALL    MESSAGE
                MOV     AL,BH                   ; 異常ｺｰﾄﾞ 表示
                CALL    BIN_ASC
                CALL    BIN_DSP                 ; 上位
                MOV     AL,BL
                CALL    BIN_ASC
                CALL    BIN_DSP                 ; 下位
                MOV     DX,OFFSET CRLF          ; 改行復帰
                CALL    MESSAGE
                JMP     LP                      ; MEWTOCOL-COM送信へ
                                        ;
; ===============================================
; =          ＭＳ－ＤＯＳへ戻る処理            =
; ===============================================
                                        ;
DOS:
                MOV     AH,4CH
                INT     21H
                                        ;
                                        ;
; ===============================================
; =          メッセージの画面表示処理            =
; ===============================================
                                        ;
MESSAGE:
                MOV     AH,9                    ; DXｱﾄﾞﾚｽから'$'迄を画面表示
                INT     21H
                RET
                                        ;
                                        ;
; ===============================================
; =          リンクソフト呼出処理                =
```

```
; ============================================
                                        ;
COMMAND:
            INT     60H                 ; ｼｽﾃﾑｺｰﾙ
            CMP     AX,0                ; AX>0時ｴﾗｰ
            JE      CMD2                ;
                                        ;
            CALL    BI2ASC              ; ｴﾗｰｺｰﾄﾞ ASCII変換
            MOV     DX,OFFSET MSG3      ; ｴﾗｰﾒｯｾｰｼﾞ 画面表示
            CALL    MESSAGE
            STC
CMD2:
            RET
                                        ;
                                        ;
; ============================================
; =       エラーコードのＡＳＣＩＩ変換       =
; ============================================
                                        ;
BI2ASC:
            MOV     BL,AH
            MOV     AH,AL
            MOV     BH,BL
            AND     AL,0FH
            AND     BL,0FH
            MOV     CL,4
            SHR     AH,CL
            SHR     BH,CL
            ADD     BX,3030H
            ADD     AX,3030H
            CMP     AL,3AH
            JB      B0
            ADD     AL,7
B0:
            CMP     AH,3AH
            JB      B1
            ADD     AH,7
B1:
            CMP     BL,3AH
            JB      B2
            ADD     BL,7
B2:
            CMP     BH,3AH
            JB      B3
            ADD     BH,7
B3:
            MOV     [MSG3_1+0],BH
            MOV     [MSG3_1+1],BL
            MOV     [MSG3_1+2],AH
            MOV     [MSG3_1+3],AL
            RET
                                        ;
                                        ;
; ============================================
; =        ＢＩＮ==>ＡＳＣＩＩ変換         =
; ============================================
                                        ;
BIN_ASC:                                ; IN...AL=BINｺｰﾄﾞ
            PUSH    SI                  ; OUT..AX=ASCIIｺｰﾄﾞ
            PUSH    BX
            PUSH    DX
                                        ;
            MOV     BL,AL
            AND     AL,0F0H             ; 上位４ビット
            SHR     AL,1
            SHR     AL,1
            SHR     AL,1
            SHR     AL,1
            LEA     SI,BINTOASC_TB      ; 変換テーブル
            MOV     AH,0
            ADD     SI,AX
            MOV     DH,BYTE PTR CS:[SI]
                                        ;
            MOV     AL,BL
```

```
                AND     AL,0FH                  ; 下位４ビット
                LEA     SI,BINTOASC_TB          ; 変換テーブル
                MOV     AH,0
                ADD     SI,AX
                MOV     AL,BYTE PTR CS:[SI]
                MOV     AH,DH
                XCHG    AL,AH
                                                ;
                POP     DX
                POP     BX
                POP     SI
                RET
                                                ;
                                                ;
;  ---------------------------------------------
;        ＨＥＸ→ＡＳＣＩＩ　変換テーブル
;  ---------------------------------------------
                                                ;
BINTOASC_TB     DB      "0123456789ABCDEF"
                                                ;
                                                ;
;  =============================================
;  =                受信データ表示               =
;  =============================================
                                                ;
RXDATA_DSP:
                CMP     BYTE PTR CS:[JBUF],24H      ; ﾚｽﾎﾟﾝｽOK?
                JE      RX_D_OK                 ; YES===>RX_D_OK へ
                                                ; ...NO
                MOV     DX,OFFSET RX_BD_MSG     ; ｴﾗｰﾚｽﾎﾟﾝｽ表示
                CALL    MESSAGE
                LEA     DI,JBUF
                MOV     AX,CS:[DI+1]
                CALL    BIN_DSP
                MOV     DX,OFFSET CRLF          ; 改行復帰
                CALL    MESSAGE
                RET
RX_D_OK:
                MOV     DX,OFFSET DSP_RXSIZE    ; 受信ｻｲｽﾞ表示
                CALL    MESSAGE
                MOV     BX,CS:[RXSIZE]
                MOV     AL,BH
                CALL    BIN_ASC
                CALL    BIN_DSP
                MOV     AL,BL
                CALL    BIN_ASC
                CALL    BIN_DSP
                MOV     DX,OFFSET CRLF
                CALL    MESSAGE
                                                ;
                MOV     DX,OFFSET DSP_FRMTP     ; 受信ﾌﾚｰﾑﾀｲﾌﾟ表示
                CALL    MESSAGE
                MOV     AL,CS:[FRMTP]
                CALL    BIN_ASC
                CALL    BIN_DSP
                MOV     DX,OFFSET CRLF
                CALL    MESSAGE
                                                ;
                MOV     DX,OFFSET DSP_TXNO      ; 送信元No.表示
                CALL    MESSAGE
                MOV     AL,CS:[TXNO]
                CALL    BIN_ASC
                CALL    BIN_DSP
                MOV     DX,OFFSET CRLF
                CALL    MESSAGE
                                                ;
                XOR     CX,CX
                LEA     DI,JBUF+3
RX_D_LP:
                MOV     DX,OFFSET DSP_DT        ; "DT"表示
                CALL    MESSAGE
                MOV     AL,CL
                CALL    BIN_ASC
                CALL    BIN_DSP                 ; DTxxのxx表示
                MOV     DX,OFFSET DSP_E
                CALL    MESSAGE                 ; "="表示
```

```
                MOV     AX,DS:[DI+2]
                CALL    BIN_DSP                 ; ﾃﾞｰﾀ表示
                MOV     AX,DS:[DI]
                CALL    BIN_DSP                 ; ﾃﾞｰﾀ表示
                                                ;
                MOV     DX,OFFSET CRLF          ; 改行復帰
                CALL    MESSAGE
                                                ;
                ADD     DI,4                    ; ﾎﾟｲﾝﾀｰ更新
                INC     CL
                CMP     CL,10
                JB      RX_D_LP
                                                ;
                MOV     DX,OFFSET CRLF
                CALL    MESSAGE
                RET
                                                ;
                                                ;
BIN_DSP:                                        ; IN...AX=表示文字ｺｰﾄﾞ
                PUSH    AX
                MOV     DL,AL
                MOV     AH,2
                INT     21H
                POP     AX
                MOV     DL,AH
                MOV     AH,2
                INT     21H
                RET
                                                ;
                                                ;
;  ============================================
;  =            Ｉ／Ｏパラメータ              =
;  ============================================
                                                ; = 初期設定 =
INIT            DW      1000H                   ; 初期設定ｺﾏﾝﾄﾞｺｰﾄﾞ
                DB      0                       ; ｱﾌﾟﾘｹｰｼｮﾝ･ｿﾌﾄﾓｰﾄﾞ
                DB      1                       ; 使用ﾎﾞｰﾄﾞ枚数
                                                ;
                                                ; = ﾎﾞｰﾄﾞ登録･起動 =
BOARD           DW      1100H                   ; ﾎﾞｰﾄﾞ登録･起動ｺﾏﾝﾄﾞｺｰﾄﾞ
                DB      0                       ; ﾎﾞｰﾄﾞ登録No.
                DB      27                      ; 使用ｾｸﾞﾒﾝﾄｱﾄﾞﾚｽNo.7(D0000H)
                DB      0                       ; 割り込みNo.
                DB      0
                                                ;
                                                ; = MEWTOCOL-COM WRITE→READ =
MEWWT           DW      1565H                   ; ｺﾏﾝﾄﾞｺｰﾄﾞ 登録ﾀｲﾌﾟ
                DB      0                       ; 登録ﾎﾞｰﾄﾞNO.
                DB      0
                DW      OFFSET SBUF             ; 送信ﾊﾞｯﾌｧOFFSET
                DW      SEG SBUF
                DW      0
                DW      14                      ; 送信ﾃﾞｰﾀ･ｻｲｽﾞ
FRMTP           DB      0                       ; 送信ﾌﾚｰﾑﾀｲﾌﾟ
                DB      0,0,0
                DB      0                       ; 制御ｺｰﾄﾞ
                DB      0                       ; 送信先CH
                DB      0                       ; 送信先ID
                DB      0                       ; 送信元CH
                DB      0                       ; 送信元ID
                DB      0
                DB      0                       ; Depth
TXNO            DB      1                       ; 送信先NO.
                DB      14 DUP(0)
                DW      OFFSET JBUF             ; 受信ﾊﾞｯﾌｧOFFSET
                DW      SEG JBUF                ; 受信ﾊﾞｯﾌｧSEGMENT
                DW      2048                    ; 受信ﾊﾞｯﾌｧｻｲｽﾞ
RXSIZE          DW      0                       ; 受信ﾃﾞｰﾀｻｲｽﾞ
ENDF            DW      0                       ; 登録ﾌﾗｸﾞ用
                                                ;
                                                ;
;  ============================================
;  =        メッセージ ／ バッファ領域        =
;  ============================================
                                                ;
MSG1            DB      '初期設定===>$'
```

パソコン機種別変更箇所
（左記はＰＣ９８シリーズの場合）

```
MSG1_OK         DB      'ＯＫ！'
                DB      CR,LF,'$'

MSG3            DB      'ｴﾗｰ.....ｺｰﾄﾞ='
MSG3_1          DB      0,0,0,0
                DB      CR,LF,'$'

MSG4            DB      '使用ボード登録 & 起動===>$'
MSG4_OK         DB      'ＯＫ！'
                DB      CR,LF,'$'

TX_MSG          DB      'ｺﾏﾝﾄﾞ=#RDD0000000009',CR,LF
                DB      'MEWTOCOL-COM  WRITE→READ登録===>$'
TX_OK_MSG       DB      'ＯＫ！',CR,LF,'$'

RX_BD_MSG       DB      'ｴﾗｰﾚｽﾎﾟﾝｽ ｺｰﾄﾞ=$'
RX_BD2_MSG      DB      '異常発生.....ｺｰﾄﾞ=$'

DSP_RXSIZE      DB      '受信ﾃﾞｰﾀｻｲｽﾞ(Hex)=$'
DSP_FRMTP       DB      '受信ﾌﾚｰﾑﾀｲﾌﾟ      =$'
DSP_TXNO        DB      '送信元No.         =$'

DSP_DT          DB      'DT$'
DSP_E           DB      '=$'
CRLF            DB      CR,LF,'$'

SBUF            DB      '#RDD0000000009'            ; 送信ﾃﾞｰﾀ

JBUF            DB      2048 DUP(0)                 ; 受信ﾊﾞｯﾌｧ

STACK           DW      256 DUP(0)                  ; ｽﾀｯｸ領域
STPT            DB      0


                CODE    ENDS
                END     START
```

パソコン機種別変更箇所

ＰＣ／ＡＴ互換機

```
BOARD           DW      1100H               ; ﾎﾞｰﾄﾞ登録･起動ｺﾏﾝﾄﾞｺｰﾄﾞ
                DB      0                   ; ﾎﾞｰﾄﾞ登録No.
                DB      8                   ; 使用ｾｸﾞﾒﾝﾄｱﾄﾞﾚｽNo.
                DB      10                  ; 割り込みNo.
                DB      0
```

ＦＭＲシリーズ

```
BOARD           DW      1100H               ; ﾎﾞｰﾄﾞ登録･起動ｺﾏﾝﾄﾞｺｰﾄﾞ
                DB      0                   ; ﾎﾞｰﾄﾞ登録No.
                DB      7                   ; 基板I/OﾎﾟｰﾄｱﾄﾞﾚｽNo.
                DW      7800H               ; ｺﾝﾄﾛｰﾙ･ｽﾃｰﾀｽI/Oﾎﾟｰﾄｱﾄﾞﾚｽ
                DB      5                   ; 割り込みNo.
                DB      0
```

Cプログラム（ソフトウェア割り込みint60）

```
/*************************************************************/
/*                  MEWTOCOL－COM                            */
/*                      (MWCM.C)                             */
/*************************************************************/
/*                    PC98シリーズ                           */
/*                                                           */
/*     1番機のPCのDT0から10ワード分読み出す                  */
/*     単一フレーム:階層無し                                 */
/*                                                           */
/*     使用コマンド：WRITE=1521H(ﾀｲﾑｱｳﾄ付き)                  */
/*                   READ =1421H(ﾀｲﾑｱｳﾄ付き)                */
/*************************************************************/

#include  <dos.h>
#include  <stdio.h>
#include  <stdlib.h>

#define ECODE "D"                   /* DT          */
#define START_N 0                   /* DT 0 から   */
#defineEND_N  9                     /* DT 9 までﾘｰﾄ       */
#define PC_N  1                     /* 局番 1      */

static union REGS inregs;
static union REGS outregs;
struct SREGS segregs;
static char far *p1;
static unsigned char dat[48];
unsigned char sbuf[2048];           /* 送信ﾊﾞｯﾌｧ     */
unsigned char jbuf[2048];           /* 受信ﾊﾞｯﾌｧ     */

void main( void )
{
       int    i;
       int    err;
       long   lwork;
       char   cwork[5];
       char   c_dmy1[4];
       char   *dmy;
       int    data,rxsize;

/* ---------------------------*/
/* ------ 初期設定処理 ------*/
/* ---------------------------*/

       dat[0] = 0x00;        /* 初期設定ｺﾏﾝﾄﾞ･ｺｰﾄﾞ=1000H  */
       dat[1] = 0x10;
        dat[2] = 0;          /* ｱﾌﾟﾘｹｰｼｮﾝ･ｿﾌﾄﾓｰﾄﾞ=0        */
        dat[3] = 1;          /* 使用ボード枚数=1           */

       printf("初期設定===>");

       if((err = mew_send()) != 0)
       {
              printf("ｴﾗｰ.....ｺｰﾄﾞ=  %x¥n", err);
              exit(-1);
       }
       printf("ＯＫ！¥n");

/* ------------------------------------*/
/* ------ 使用ボード登録 及び起動 ------*/
/* ------------------------------------*/

       dat[0] = 0x00;        /* 使用ボード登録及び起動    */
       dat[1] = 0x11;        /* ｺﾏﾝﾄﾞｺｰﾄﾞ=1100H           */
       dat[2] = 0x00;        /* 登録ボードNo.=0           */
       dat[3] = 27;          /* 使用ｾｸﾞﾒﾝﾄｱﾄﾞﾚｽNo.=27     */
       dat[4] = 0;           /* 割り込みNo.=0             */

       printf("使用ボード登録 & 起動===>");

       if((err = mew_send()) != 0)
```

パソコン機種別変更箇所
（左記はＰＣ９８シリーズの場合）

```
        {
                printf("ｴﾗｰ.....ｺｰﾄﾞ=  %x¥n",  err);
                exit(-1);
        }
        printf("ＯＫ！¥n");

/* ------------------------------*/
/* ------ MEWTOCOL-COM WRITE ------*/
/* ------------------------------*/

TXLP:
        memset( sbuf,  0,  sizeof( sbuf ) );
        sprintf( sbuf,  "#RD%s%05d%05d", ECODE, START_N, END_N  );
        printf("ｺﾏﾝﾄﾞ=%s¥n", sbuf);
        p1 = (char far *)&sbuf[0];

        dat[0]  = 0x21;         /* MEWTOCOL-COM WRITE */
        dat[1]  = 0x15;         /* ｺﾏﾝﾄﾞｺｰﾄﾞ=1521H    */
         dat[2]  = 0x00;        /* 登録ボードno.=0    */
         dat[3]  = 0x00;
        *((unsigned *)&dat[4])  = FP_OFF(p1);       /* 送信ﾊﾞｯﾌｧｵﾌｾｯﾄ    */
        *((unsigned *)&dat[6])  = FP_SEG(p1);       /* 送信ﾊﾞｯﾌｧｾｸﾞﾒﾝﾄ   */
        *((unsigned *)&dat[10])  = strlen(sbuf);    /* 送信ﾃﾞｰﾀｻｲｽﾞ       */
        dat[12]  = 0;                               /* 送信ﾌﾚｰﾑﾀｲﾌﾟ       */
        dat[16]  = 0;                               /* 書き込み制御ｺｰﾄﾞ  */
        dat[17]  = 0;           /* 送信先CH=0        */
        dat[18]  = 0;           /* 送信先ID=0        */
        dat[19]  = 0;           /* 送信元CH=0        */
        dat[20]  = 0;           /* 送信元ID=0        */
        dat[22]  = 0;           /* Depth=0(階層なし)*/
        dat[23]  = PC_N;        /* 送信先No.(PC局番0*/

        printf("ﾊﾟｹｯﾄ送信===>");

        if((err = mew_send())  != 0)
        {
                printf("ｴﾗｰ.....ｺｰﾄﾞ=  %x ¥n",  err);
                goto TXLP;
        }

        printf("ＯＫ！¥n");


/* ------------------------------*/
/* ------ MEWTOCOL-COM READ ------*/
/* ------------------------------*/

        p1 = (char far *)&jbuf[0];

        dat[0]  = 0x21;         /* MEWTOCOL-COM READ */
        dat[1]  = 0x14;         /* ｺﾏﾝﾄﾞｺｰﾄﾞ=1421H    */
        dat[2]  = 0x00;         /* 登録ボードno.=0    */
        dat[3]  = 0x00;         /* 受信ch no.         */
        *((unsigned *)&dat[4])  = FP_OFF(p1);       /* 受信ﾊﾞｯﾌｧｵﾌｾｯﾄ   */
        *((unsigned *)&dat[6])  = FP_SEG(p1);       /* 受信ﾊﾞｯﾌｧｾｸﾞﾒﾝﾄ  */
        *((unsigned *)&dat[8])  = 2048;             /* 受信ﾊﾞｯﾌｧｻｲｽﾞ     */
        dat[16]  = 0;                               /* 受信制御ｺｰﾄﾞ     */
        dat[19]  = 0;                               /* 受信CH No.=0     */

        printf("ﾚｽﾎﾟﾝｽ読出===>");

        if((err = mew_send())  != 0)
        {
                printf("ｴﾗｰ.....ｺｰﾄﾞ=  %x¥n",  err);
        }
        else
        {
                printf("ＯＫ！¥n");

                if( jbuf[0]  == '!' )
                {
                        printf( "ｴﾗｰﾚｽﾎﾟﾝｽ ｺｰﾄﾞ= %c%c¥n",  jbuf[1], jbuf[2]  );
                }
                else
                {
                        rxsize = dat[10]  + dat[11]  * 0x100;
                        printf("受信ﾃﾞｰﾀｻｲｽﾞ=%d¥n", rxsize);
```

```
                    printf("受信ﾌﾚｰﾑﾀｲﾌﾟ=%d¥n",dat[12]);
                    printf("送信元No.    =%d¥n",dat[23]);
                    for( i = 0 ; i < END_N - START_N + 1 ; i++)
                    {
                            cwork[ 2 ] = jbuf[i*4+3];
                            cwork[ 3 ] = jbuf[i*4+4];
                            cwork[ 0 ] = jbuf[i*4+5];
                            cwork[ 1 ] = jbuf[i*4+6];
                            cwork[ 4 ] = '¥0';
                            lwork = strtol( &cwork[ 0 ], &dmy, 16 );
                            data = (int)lwork;
                            printf("%s%d=%x¥n",ECODE,START_N+i,data);
                    }
                    printf("¥n");
            }
        }
        goto TXLP;
}


/* ---------------------------*/
/* ------ ﾘﾝｸｿﾌﾄ呼び出し------*/
/* ---------------------------*/
int mew_send(void)
{

        p1 = (char far *)&dat[0];

        inregs.x.dx  = FP_OFF(p1);
        segregs.ds  = FP_SEG(p1);

        int86x( 0x60, &inregs, &outregs, &segregs);
        return(outregs.x.ax);
}
```

```
/*************************************************************/
/*       MEWTOCOL－COM WRITE→READ                            */
/*                          (MWCM2.C)                        */
/*************************************************************/
/*                  PC９８シリーズ                           */
/*                                                           */
/*      １番機のＰＣのＤＴ０から１０ワード分読み出す         */
/*      単一フレーム:階層無し                                */
/*                                                           */
/*      使用コマンド：1565H(登録送受信要求ﾀｲﾑｱｳﾄ付き)         */
/*************************************************************/

#include  <dos.h>
#include  <stdio.h>
#include  <stdlib.h>

#define ECODE "D"                       /* DT              */
#define START_N 0                       /* DT 0 から       */
#defineEND_N  9                         /* DT 9 までﾘｰﾄﾞ    */
#define PC_N  1                         /* 局番 1          */

static union REGS inregs;
static union REGS outregs;
struct SREGS segregs;
static char far *p1;
static unsigned char dat[48];
unsigned char sbuf[2048];                   /* 送信ﾊﾞｯﾌｧ     */
unsigned char jbuf[2048];                   /* 受信ﾊﾞｯﾌｧ     */

void main( void )
{
        int    i;
```

```
        int     err;
        long    lwork;
        char    cwork[5];
        char    c_dmy1[4];
        char    *dmy;
        int     data,rxsize,endf;

/* -------------------------*/
/* ------ 初期設定処理 ------*/
/* -------------------------*/

        dat[0] = 0x00;          /* 初期設定ｺﾏﾝﾄﾞ･ｺｰﾄﾞ=1000H  */
        dat[1] = 0x10;
         dat[2] = 0;            /* ｱﾌﾟﾘｹｰｼｮﾝ･ｿﾌﾄﾓｰﾄﾞ=0        */
         dat[3] = 1;            /* 使用ボード枚数=1           */

        printf("初期設定===>");

        if((err = mew_send()) != 0)
        {
                printf("ｴﾗｰ.....ｺｰﾄﾞ=  %x¥n", err);
                exit(-1);
        }
        printf("ＯＫ！¥n");

/* ---------------------------------*/
/* ------ 使用ボード登録 及び起動 ------*/
/* ---------------------------------*/

        dat[0] = 0x00;          /* 使用ボード登録及び起動    */
        dat[1] = 0x11;          /* ｺﾏﾝﾄﾞｺｰﾄﾞ=1100H           */
        dat[2] = 0x00;          /* 登録ボードNo.=0           */
        dat[3] = 27;            /* 使用ｾｸﾞﾒﾝﾄｱﾄﾞﾚｽNo.=27     */
        dat[4] = 0;             /* 割り込みNo.=0             */

        printf("使用ボード登録 & 起動===>");

        if((err = mew_send()) != 0)
        {
                printf("ｴﾗｰ.....ｺｰﾄﾞ=  %x¥n", err);
                exit(-1);
        }
        printf("ＯＫ！¥n");

/* ---------------------------------------*/
/* ------ MEWTOCOL-COM WRITE→READ登録 ------*/
/* ---------------------------------------*/

TXLP:
        memset( sbuf, 0, sizeof( sbuf ) );
        sprintf( sbuf, "#RD%s%05d%05d", ECODE, START_N, END_N  );
        printf("ｺﾏﾝﾄﾞ=%s¥n",sbuf);
        p1 = (char far *)&sbuf[0];

        dat[0] = 0x65;          /* MEWTOCOL-COM WRITE→READ */
        dat[1] = 0x15;          /* ｺﾏﾝﾄﾞｺｰﾄﾞ=1565H          */
         dat[2] = 0x00;         /* 登録ボードno.=0          */
         dat[3] = 0x00;
        *((unsigned *)&dat[4]) = FP_OFF(p1);        /* 送信ﾊﾞｯﾌｧｵﾌｾｯﾄ   */
        *((unsigned *)&dat[6]) = FP_SEG(p1);        /* 送信ﾊﾞｯﾌｧｾｸﾞﾒﾝﾄ  */
        *((unsigned *)&dat[10]) = strlen(sbuf);     /* 送信ﾃﾞｰﾀｻｲｽﾞ      */
        dat[12] = 0;            /* 送信ﾌﾚｰﾑﾀｲﾌﾟ      */
        dat[16] = 0;            /* 書き込み制御ｺｰﾄﾞ  */
        dat[17] = 0;            /* 送信先CH=0        */
        dat[18] = 0;            /* 送信先ID=0        */
        dat[19] = 0;            /* 送信元CH=0        */
        dat[20] = 0;            /* 送信元ID=0        */
        dat[22] = 0;            /* Depth=0(階層なし)*/
        dat[23] = PC_N;         /* 送信先No.(PC局番0*/

        p1 = (char far *)&jbuf[0];
        *((unsigned *)&dat[38]) = FP_OFF(p1);       /* 受信ﾊﾞｯﾌｧｵﾌｾｯﾄ   */
        *((unsigned *)&dat[40]) = FP_SEG(p1);       /* 受信ﾊﾞｯﾌｧｾｸﾞﾒﾝﾄ  */
        *((unsigned *)&dat[42]) = 2048;             /* 受信ﾊﾞｯﾌｧｻｲｽﾞ     */


        printf("MEWTOCOL-COM  WRITE→READ登録===>");

        if((err = mew_send()) != 0)
        {
                printf("ｴﾗｰ.....ｺｰﾄﾞ=  %x ¥n", err);
                goto TXLP;
        }

        printf("ＯＫ！¥n");


/* -----------------------------------------*/
```

パソコン機種別変更箇所
（左記はＰＣ９８シリーズの場合）

```
/* ------ MEWTOCOL-COM WRITE→READ完了ﾁｪｯｸ ------*/
/* ----------------------------------------*/

        do
        {
                endf = dat[46] + dat[47] * 0x100;
                if(endf == 1)
                {
                        if( jbuf[0] == '!' )
                        {
                                printf( "ｴﾗｰﾚｽﾎﾟﾝｽ ｺｰﾄﾞ= %c%c¥n", jbuf[1],jbuf[2]
);
                        }
                        else
                        {
                                rxsize = dat[44] + dat[45] * 0x100;
                                printf("受信ﾃﾞｰﾀｻｲｽﾞ=%d¥n",rxsize);
                                printf("受信ﾌﾚｰﾑﾀｲﾌﾟ=%d¥n",dat[12]);
                                printf("送信元No.    =%d¥n",dat[23]);
                                for( i = 0 ; i < END_N - START_N + 1 ; i++)
                                {
                                        cwork[ 2 ] = jbuf[i*4+3];
                                        cwork[ 3 ] = jbuf[i*4+4];
                                        cwork[ 0 ] = jbuf[i*4+5];
                                        cwork[ 1 ] = jbuf[i*4+6];
                                        cwork[ 4 ] = '¥0';
                                        lwork = strtol( &cwork[ 0 ], &dmy, 16 );
                                        data = (int)lwork;
                                        printf("%s%d=%x¥n",ECODE,START_N+i,data);
                                }
                                printf("¥n");
                        }
                }
                else
                {
                        if(endf != 0)
                        {
                                printf("異常発生.....ｺｰﾄﾞ= %x ¥n",endf);
                        }
                }
        }
        while(endf == 0);

        goto TXLP;
}


/* ---------------------------*/
/* ------ ﾘﾝｸｿﾌﾄ呼び出し------*/
/* ---------------------------*/
int mew_send(void)
{

        p1 = (char far *)&dat[0];

        inregs.x.dx  = FP_OFF(p1);
        segregs.ds = FP_SEG(p1);

        int86x( 0x60, &inregs, &outregs, &segregs);
        return(outregs.x.ax);
}
```

パソコン機種別変更箇所

ＰＣ／ＡＴ互換機

```
        dat[0] = 0x00;          /* 使用ボード登録及び起動    */
        dat[1] = 0x11;          /* ｺﾏﾝﾄﾞｺｰﾄﾞ=1100H           */
        dat[2] = 0x00;          /* 登録ボードNo.=0           */
        dat[3] = 8;             /* 使用ｾｸﾞﾒﾝﾄｱﾄﾞﾚｽNo.=8      */
        dat[4] = 10;            /* 割り込みNo.=10            */
```

ＦＭＲシリーズ

```
        dat[0] = 0x00;          /* 使用ボード登録及び起動    */
        dat[1] = 0x11;          /* ｺﾏﾝﾄﾞｺｰﾄﾞ=1100H           */
        dat[2] = 0x00;          /* 登録ボードNo.=0           */
        dat[3] = 7;             /* 基板I/OﾎﾟｰﾄｱﾄﾞﾚｽNo.=7     */
        dat[4] = 0x00;          /* ｽﾃｰﾀｽ･ｺﾝﾄﾛｰﾙI/Oﾎﾟｰﾄｱﾄﾞﾚｽ */
        dat[5] = 0x78;          /* =7800H                    */
        dat[6] = 5;             /* 割り込みNo.=5             */
```

3－4 シリアル伝送の機能コマンド

3-4-1 シリアル伝送機能の使用登録／解除

シリアル伝送／シリアル伝送機能の使用登録／解除（コマンドコード&H180x）

機　能

シリアル伝送機能の使用をリンクボードへ登録、または登録の解除を行います。

解　説

・リンクボードのシリアル伝送機能の動作モードを次から指定します。
　モード　0：多重接続マスタ
　　2：単一接続
・機能コマンドを実行するには、各種プログラミング言語でI/Oパラメータのセグメントアドレスとオフセットアドレスをds:dxレジスタにセットし、割り込みを実行するユーザープログラムを作成してください。
・コマンド実行後の結果（リターン値）は、axレジスタに格納されます。
　0：正常終了時
　その他：エラーコード（「3－9」参照）

■コマンドタイプ

コード	機能
&H1800	全登録解除
&H1801	登録解除（ボード番号指定）
&H1802	登録

注　意

・本コマンドを使用して登録を行わないと、他のシリアル伝送機能コマンドは使用できません。
・動作モードには、多重接続マスタ、単一接続が指定できます。
・&H1800では、I/Oパラメータの動作モード以降の指定は不要です。
・動作モードが多重接続マスタの場合は、I/Oパラメータの通信先ユニットNo.以降の指定は不要です。

文　例

●MS-DOSファンクションコール（int21h）

```
;----- シリアル登録 ----
RS_0:
        mov     bx, [FILHDL]
        mov     ah, 44h
        mov     al, 3
        mov     dx, offset RS_0
        int     21h
;------- I/Oﾊﾟﾗﾒｰﾀ ------
RS_0    dw      1802h
        db      0, 0, 1, 10, 1
        dw      0
        db      9, 7, 0, 3, 1, 3, 7, 2, 3, 1
        dw      14 dup(?)
        db      ?
```

●ソフトウェア割り込み（int60h）

```
;----- シリアル登録 ----
RS_0:
        mov     dx, offset RS_0
        int     60h
;------- I/Oﾊﾟﾗﾒｰﾀ ------
RS_0    dw      1802h
        db      0, 0, 1, 10, 1
        dw      0
        db      9, 7, 0, 3, 1, 3, 7, 2, 3, 1
        dw      14 dup(?)
        db      ?
```

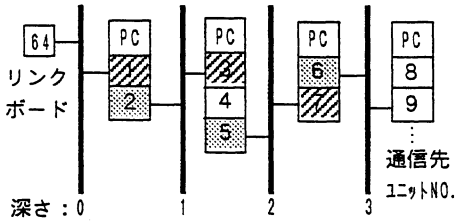
I/Oパラメータ

△：設定する［渡す値］
▼：実行後に格納される［得る値］

dsレジスタ ←I/Oパラメータの先頭セグメントアドレス
dxレジスタ ←I/Oパラメータの先頭オフセットアドレス

0	△［コマンドコード］設定値：&H180x
2	△［登録ボードNo.］0～3　使用するリンクボードの登録ボードNo.を設定します。
	△ 設定値：00固定で使用してください。
4	△［動作モード］0：多重接続マスタ／2：単一接続
	△［タイムアウト値］データおよび制御コマンド送受信時のタイムアウト値（0.5×n秒）
6	△［再送指定］伝送異常時の再送信指定　0：再送信しない／1：再送信する
	△［再送回数］再送信する場合の再送信回数（0：無限）
8	単一接続時のみ設定します
	△［通信先ユニットNo.］1～64（通信先のUNIT No.）
A	△［通信先CH No.］通信先がリンクユニット・ボードのRS232Cの場合：7／その他の場合：F
	△ 設定値：00固定で使用してください。
C	△［深さ］0～3（階層間リンクのリンクの深さ）
E	△［伝送経路］中継局No.→中継リンクの順で指定します。
10	
12	
14	
16	
18	
1A	
1C	
1E	
20	
22	
24	
26	
28	
2A	
2C	
2E	

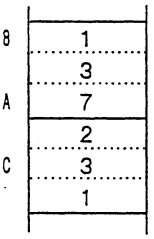
伝送経路の指定



▨中継局No.（UNIT No.を指定）
▩中継リンク（ＰＣから何ユニット目かを指定）

左図のネットワーク例では
・深さ⇒3
・通信先ユニットNo.⇒9
・中継局No.⇒1、3、7
・中継リンク⇒2、3、1

＊ネットワークの各層のリンクユニットが2台までの場合、深さ0を指定することもできます。



⇦システムで使用しますのでI/Oパラメータは必ず左記の通り確保しておいてください。

3-4-2 多重接続マスタ用制御コマンドWRITE/READ

シリアル伝送／多重接続モード制御コマンド処理（コマンドコード&H181x）

機　能

多重接続マスタ用制御コマンドのＷＲＩＴＥ→ＲＥＡＤを一連で行います。

解　説

・リンクボードを多重接続マスターとして使用する場合、本コマンドを使用して制御コマンドの送信、および通信相手局からのレスポンスの受信を行います。
・機能コマンドを実行するには、各種プログラミング言語でI/Oパラメータのセグメントアドレスとオフセットアドレスをds:dxレジスタにセットし、割り込みを実行するユーザープログラムを作成してください。
・コマンド実行後の結果（リターン値）は、axレジスタに格納されます。
　0：正常終了時
　その他：エラーコード（「３－９」参照）

登録送受信要求タイプ、登録送信要求タイプ、登録受信要求タイプの機能コマンドには、同時に使用できる（実行状態にある）コマンド数に制限があります。同時に使用できるコマンド数は、MEWNET-Hリンクソフト（デバイスドライバ）のCONFIG.SYSへの登録時のパラメータスイッチで、0～16まで指定できます。
CONFIG.SYSでの指定がない場合は、同時に実行できるコマンド数は4つまでです。CONFIG.SYSでの指定については、「1-1-2」をお読みください。

■コマンドタイプ

コード	機能
&H1810	完了待ち（PC98のみKEY Break付き）
&H1811	完了待ち（タイムアウト付き）
&H1815	登録送受信要求（タイムアウト付き）

注　意

・&H1810のKEY Breakは、ESCキーのみです。
・&H1811（完了待ち）は、レスポンス受信完了、またはタイムアウトまで待ちます。
　タイムアウト値は、&1802（シリアル伝送機能登録／解除）にて設定します。
・&H1815（登録送受信要求）は、制御コマンドの送信要求のみを行って、ユーザープログラムへ制御を戻します。
・&H1815（登録送信要求）コマンド実行後のaxレジスタのリターン値には、送信要求までの処理結果が返ります。axレジスタのリターン値=0で、完了フラグ=0の時は、まだ処理を完了していません。処理完了は、完了フラグ≠0で検知してください。
・送信バッファに格納する制御コマンドのフォーマットについては、４章をお読みください。

文　例

●MS-DOSファンクションコール（int21h）

```
;----- シリアル制御コマンド ----
RS_ST:
        mov     bx,[FILHDL]
        mov     ah,44h
        mov     al,3
        mov     dx,offset RS_ST
        int     21h
;------- I/Oﾊﾟﾗﾒｰﾀ ------
RS_ST   dw      1811h
        db      0,0,
        dw      offset WBUF,seg WBUF,?,9
        dw      13 dup(0)
        dw      offset RBUF,seg RBUF,256,0,0
RBUF    db      256 dub(?)
WBUF    db      '<AA#S2'
```

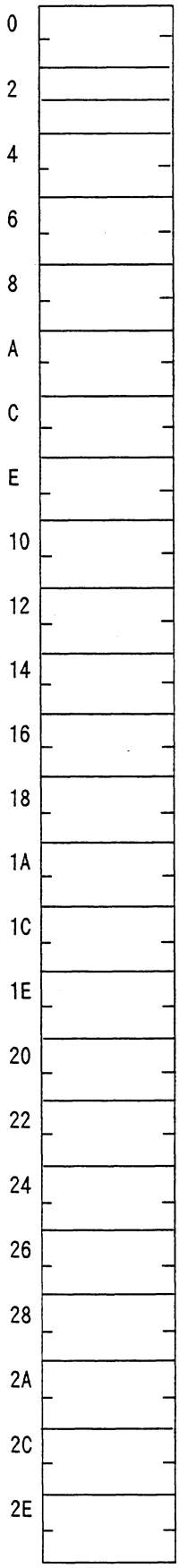
●ソフトウェア割り込み（int60h）

```
;----- シリアル制御コマンド ----
RS_ST:
        mov     dx,offset RS_ST
        int     60h
;------- I/Oﾊﾟﾗﾒｰﾀ ------
RS_ST   dw      1811h
        db      0,0,
        dw      offset WBUF,seg WBUF,?,9
        dw      13 dup(0)
        dw      offset RBUF,seg RBUF,256,0,0
RBUF    db      256 dub(?)
WBUF    db      '<AA#S2'
```

I/Oパラメータ

△：設定する［渡す値］
▼：実行後に格納される［得る値］

dsレジスタ ←I/Oパラメータの先頭セグメントアドレス
dxレジスタ ←I/Oパラメータの先頭オフセットアドレス



0 △［コマンドコード］設定値：&H181x

2 △［登録ボードNo.］格納値：0～3　使用するリンクボードの登録ボードNo.を設定します。
△ 設定値：00固定で使用してください。

4 △［送信バッファアドレス（オフセット）］
6 △［送信バッファアドレス（セグメント）］
　送信バッファには、コマンドデータ（#を含みBCCの直前まで）をセットしておきます。

8

A △［送信データサイズ］送信バッファにセットしたコマンドデータのサイズ（バイト数）。

C

E

10

12

14

16

18

1A

1C

1E

20

22

24

26 △［受信バッファアドレス（オフセット）］
28 △［受信バッファアドレス（セグメント）］
　受信バッファには、送信先からのレスポンスデータ（$または!を含みBCCの直前まで）が格納されます。（ただし、スレーブ局へのデータ受信要求でデータ有り時を除く。）

2A △［受信バッファサイズ］バイト数を指定します。

2C ▼［受信データサイズ］受信バッファに格納したレスポンスデータのサイズ（バイト数）。

2E ▼［完了フラグ］0：未完了／1：正常終了／その他：エラーコード
&H1815のみセットされます。

3-4-3 シリアルデータ送信

シリアル伝送／シリアルデータ送信（コマンドコード&H182x）

機　能

シリアルデータの送信を行います。

解　説

・シリアル伝送のすべての動作モードで、シリアルデータの送信を行います。
・機能コマンドを実行するには、各種プログラミング言語でI/Oパラメータのセグメントアドレスとオフセットアドレスをds:dxレジスタにセットし、割り込みを実行するユーザープログラムを作成してください。
・コマンド実行後の結果（リターン値）は、axレジスタに格納されます。
　0：正常終了時
　その他：エラーコード（「3－9」参照）

■コマンドタイプ

コード	機能
&H1820	送信完了待ち（PC98のみKEY Break付き）
&H1821	送信完了待ち（タイムアウト付き）
&H1822	送信要求（要求のみで完了まで待たない）
&H1823	完了センス（送信要求の完了検知用）
&H1826	強制停止（再送／スレーブ時の送信中断用）

注　意

・送信要求の動作は次の通りです。
　単一接続時：登録時に設定された通信局に対して送信を行います。
　多重接続マスタ時：多重マスタ用制御コマンドで指定された通信局（スレーブ）に対して送信します。通信局の指定の無い場合は、エラー（149H）が返ります。
・前回要求したデータの送信が未完了時は、新たにデータの送信要求はできません（エラー130H）。
・次データの送信が可能かどうかは、完了センスタイプ（&H1823）で判定します。コマンド実行後のリターン値（axレジスタの値）が0の時、次データ送信可能。完了センスタイプでは、完了検知できなかった場合、エラー（134H）が返ります。

文　例

●MS-DOSファンクションコール（int21h）

```
;----- シリアル送信 ----
RS_W:
        mov     bx,[FILHDL]
        mov     ah,44h
        mov     al,3
        mov     dx,offset RS_W
        int     21h
;------- I/Oﾊﾟﾗﾒｰﾀ ------
RS_W    dw      1821h
        db      0,0
        dw      offset WBUF,seg WBUF,?,4,
                18 dup(?)
WBUF    db      'ABCD'
```

●ソフトウェア割り込み（int60h）

```
;----- シリアル送信 ----
PCRRD:
        mov     dx,offset RS_W
        int     60h
;------- I/Oﾊﾟﾗﾒｰﾀ ------
RS_W    dw      1821h
        db      0,0
        dw      offset WBUF,seg WBUF,?,4,
                18 dup(?)
WBUF    db      'ABCD'
```

I/Oパラメータ

△：設定する［渡す値］
▼：実行後に格納される［得る値］

dsレジスタ ←I/Oパラメータの先頭セグメントアドレス
dxレジスタ ←I/Oパラメータの先頭オフセットアドレス

0		△［コマンドコード］設定値：&H182x
2		△［登録ボードNo.］格納値：0～3　使用するリンクボードの登録ボードNo.を設定します。
		△ 設定値：00固定で使用してください。
4		△［送信バッファアドレス（オフセット）］
6		△［送信バッファアドレス（セグメント）］
8		
A		△［送信データサイズ］送信バッファにセットしたデータのサイズ（バイト数）。
C		
E		
10		
12		
14		
16		
18		
1A		
1C		
1E		
20		
22		
24		
26		
28		
2A		
2C		
2E		

送信するデータをセットしておきます。

注　意

・完了センスおよび強制停止タイプでは、送信バッファアドレス以降の指定は不要です。

⇦システムで使用しますのでI/Oパラメータは必ず左記の通り確保しておいてください。

3-4-4 シリアルデータ受信

シリアル伝送／シリアルデータ受信（コマンドコード&H1830）

機　能

単一接続において、シリアルデータの受信を行います。

解　説

・シリアル伝送機能を単一接続モードで使用する場合、本コマンドでデータの受信を行います。多重接続マスタモードで使用する場合のデータ受信には、多重接続マスタ用制御コマンドWRITE→READを使用します。
・機能コマンドを実行するには、各種プログラミング言語でI/Oパラメータのセグメントアドレスとオフセットアドレスをds:dxレジスタにセットし、割り込みを実行するユーザープログラムを作成してください。
・コマンド実行後の結果（リターン値）は、axレジスタに格納されます。
　0：正常終了時
　その他：エラーコード（「3－9」参照）

■コマンドタイプ

コード	機能
&H1820	受信完了待ち（PC98のみKEY Break付き）
&H1821	受信完了待ち（タイムアウト付き）
&H1823	完了センス（受信要求の完了検知用）
&H1826	受信バッファクリア

注　意

・受信データがある場合は、指定されたバッファにデータを格納します。
・受信データが無い場合は、エラー（133H）が返ります。
・多重接続マスタの受信は、多重接続マスタ用制御コマンド（S2）にて行います。

文　例

●MS-DOSファンクションコール（int21h）

```
;----- PCﾘﾝｸﾗﾝﾀﾞﾑREAD ----
RS_R:
        mov     bx,[FILHDL]
        mov     ah,44h
        mov     al,3
        mov     dx,offset RS_R
        int     21h
;------- I/Oﾊﾟﾗﾒｰﾀ ------
RS_R    dw      1831h
        db      0,0
        dw      offset RBUF,seg RBUF,256,0
        dw      18 dup(?)
RBUF    dw      256 dub(?)
```

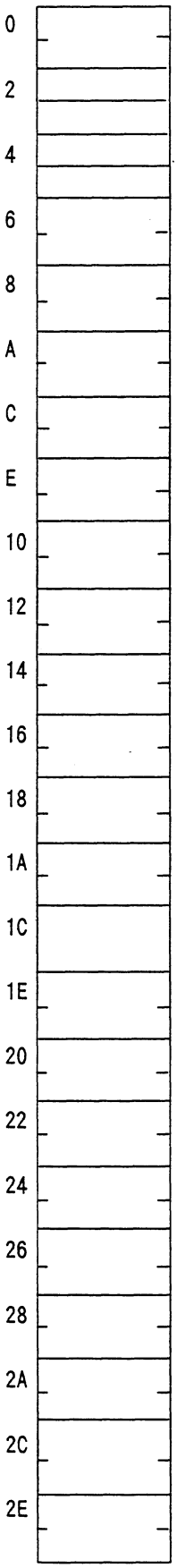
●ソフトウェア割り込み（int60h）

```
;----- PCﾘﾝｸﾗﾝﾀﾞﾑREAD ----
RS_R:
        mov     dx,offset RS_R
        int     60h
;------- I/Oﾊﾟﾗﾒｰﾀ ------
RS_R    dw      1831h
        db      0,0
        dw      offset RBUF,seg RBUF,256,0
        dw      18 dup(?)
RBUF    dw      256 dub(?)
```

I/Oパラメータ

△：設定する［渡す値］
▼：実行後に格納される［得る値］

dsレジスタ ←I/Oパラメータの先頭セグメントアドレス
dxレジスタ ←I/Oパラメータの先頭オフセットアドレス



△［コマンドコード］設定値：&H1830

△［登録ボードNo.］格納値：0～3　使用するリンクボードの登録ボードNo.を設定します。
△ 設定値：00固定で使用してください。

△［受信バッファアドレス（オフセット）］

△［受信バッファアドレス（セグメント）］

△［受信バッファアサイズ］バイト数を指定します。

▼［受信データサイズ］受信バッファに格納したデータのサイズ（バイト数）。

⇦システムで使用しますのでI/Oパラメータは必ず左記の通り確保しておいてください。

3-4-5 シリアルデータ伝送のプログラム例

アセンブラプログラム（MS-DOSシステムコールint21）

```
; ********************************************************
; *              シリアル伝送機能　単一接続              *
; *                    <ABSSG.ASM>                       *
; ********************************************************
; *                      ＰＣ９８シリーズ                *
; *    マスター処理:                                     *
; *    ６４番機(ﾊﾟｿｺﾝI/Fﾎﾞｰﾄﾞ) へ文字データを送信します。*
; *    ６４番機から返信があれば発行データを変更して      *
; *    いきます。                                        *
; *    ６４番機をABSSG2ﾀｲﾌﾟで起動させて下さい。          *
; *                                                      *
; *     使用コマンド：登録 =1802H(単一接続)              *
; *                   WRITE=1821H(ﾀｲﾑｱｳﾄ付き)           *
; *                   READ =1832H(完了ｾﾝｽﾀｲﾌﾟ)          *
; ********************************************************
                                        ;
CODE    SEGMENT
        ASSUME CS:CODE,DS:CODE,ES:CODE,SS:CODE


CR              EQU     0DH             ; CRｺｰﾄﾞ
LF              EQU     0AH             ; LFｺｰﾄﾞ
                                        ;
                                        ;
START:
                MOV     AX,CS
                MOV     DS,AX
                MOV     ES,AX
                MOV     SS,AX
                MOV     AX,OFFSET STPT
                MOV     SP,AX
                                        ;
;  ---------------------------------------------
;                 初　期　設　定
;  ---------------------------------------------
INI:
                MOV     DX,OFFSET MSG1          ; 初期設定メッセージ表示
                CALL    MESSAGE
                MOV     DX,OFFSET INIT          ; 初期設定用I/Oﾊﾟﾗﾒｰﾀ ｱﾄﾞﾚｽ設定
                CALL    COMMAND                 ; ﾘﾝｸｿﾌﾄ呼出
                JNB     INI_OK
                JMP     DOS                     ; ｴﾗｰ時 MS-DOSへ戻る
                                        ;
INI_OK:
                MOV     DX,OFFSET MSG1_OK       ; 初期設定ＯＫﾒｯｾｰｼﾞ 表示
                CALL    MESSAGE
                                        ;
;  ---------------------------------------------
;         ボ　ー　ド　登　録　・　起　動
;  ---------------------------------------------
                                        ;
                MOV     DX,OFFSET MSG4          ; ﾎﾞｰﾄﾞ登録･起動メッセージ表示
                CALL    MESSAGE
                MOV     DX,OFFSET BOARD                 ; ﾎﾞｰﾄﾞ登録用I/Oﾊﾟﾗﾒｰﾀ ｱﾄﾞﾚｽ
                CALL    COMMAND                 ; ﾘﾝｸｿﾌﾄ呼出
                JNB     BDST_OK
                JMP     DOS                     ; ｴﾗｰ時 MS-DOSへ戻る
                                        ;
BDST_OK:
                MOV     DX,OFFSET MSG4_OK       ; ﾎﾞｰﾄﾞ登録&起動OKﾒｯｾｰｼﾞ 表示
                CALL    MESSAGE
                                        ;
;  ---------------------------------------------
;                 単一接続モード登録
;  ---------------------------------------------
                                        ;
BSSGST:
                MOV     DX,OFFSET BSSG_MSG      ; 登録メッセージ
                CALL    MESSAGE
```

```
                MOV     DX,OFFSET  BSSG_ST      ; 登録I/Oﾊﾟﾗﾒｰﾀｱﾄﾞﾚｽ
                CALL    COMMAND
                JNB     BSSGST_OK
                JMP     DOS                     ; ｴﾗｰ時 MS-DOSへ戻る
                                                ;
BSSGST_OK:
                MOV     DX,OFFSET  BSSG_OK_MSG ; 登録OKﾒｯｾｰｼﾞ 表示
                CALL    MESSAGE
                                                ;
; ---------------------------------------------
;           シリアルデータ ＷＲＩＴＥ
; ---------------------------------------------
                                                ;
                MOV     CS:[DATANO],0
TXLP:
                MOV     BX,CS:[DATANO]          ; BX=送信ﾃﾞｰﾀNo.
                SHL     BX,1
                SHL     BX,1
                LEA     DI,TX_DATA_TB
                MOV     AX,CS:[DI+BX]           ; AX=送信ﾃﾞｰﾀ･ｻｲｽﾞ
                MOV     CS:[TXSIZE],AX
                MOV     AX,CS:[DI+BX+2]         ; AX=送信ﾊﾞｯﾌｧOFFSET
                MOV     CS:[TXBUFF_OFF],AX
TXLP2:
                MOV     DX,OFFSET  TX_DATA_MSG ; 発行ﾃﾞｰﾀ表示
                CALL    MESSAGE
                MOV     DX,CS:[TXBUFF_OFF]
                CALL    MESSAGE
                                                ;
                MOV     DX,OFFSET  TX_MSG       ; 文字ﾃﾞｰﾀ発行ﾒｯｾｰｼﾞ 表示
                CALL    MESSAGE
                MOV     DX,OFFSET  MEWWT              ; WRITEI/Oﾊﾟﾗﾒｰﾀ ｱﾄﾞﾚｽ
                CALL    COMMAND
                JB      TXLP2                   ; ｴﾗｰ時 再送
                                                ;
                MOV     DX,OFFSET  TX_OK_MSG    ; ＯＫﾒｯｾｰｼﾞ 表示
                CALL    MESSAGE
                                                ;
; ---------------------------------------------
;           シリアルデータ ＲＥＡＤ
; ---------------------------------------------
                                                ;
                MOV     DX,OFFSET  RX_MSG       ; 文字型ﾃﾞｰﾀ受信ｾﾝｽﾒｯｾｰｼﾞ 表示
                CALL    MESSAGE
                MOV     DX,OFFSET  MEWRD        ; READ用I/Oﾊﾟﾗﾒｰﾀ ｱﾄﾞﾚｽ
                CALL    COMMAND
                JNB     RD_OK                   ; 受信センス===>RD_OK へ
                                                ; ...NO
                CMP     AX,133H                 ; 受信ｾﾝｽ出来ず?
                JNE     RD_ED                   ; NO ===>RD_ED へ
                                                ; ...YES
                MOV     DX,OFFSET  RX_BD_MSG    ; 受信ｾﾝｽ出来ずﾒｯｾｰｼﾞ 表示
                CALL    MESSAGE
                JMP     RD_ED
                                                ;
RD_OK:                                          ; == 受信センス時 ==
                MOV     DX,OFFSET  RX_OK_MSG    ; ＯＫﾒｯｾｰｼﾞ 表示
                CALL    MESSAGE
                                                ;
                CALL    RXDATA_DSP              ; 受信内容表示
                MOV     AX,CS:[DATANO]
                INC     AX
                CMP     AX,4
                JB      RD_10
                XOR     AX,AX
RD_10:
                MOV     CS:[DATANO],AX          ; 送信ﾃﾞｰﾀNO.更新
RD_ED:
                MOV     DX,OFFSET  CRLF         ; 改行復帰
                CALL    MESSAGE
                JMP     TXLP                    ; 文字型WRITEへ
                                                ;
                                                ;
; ===============================================
; =           ＭＳ－ＤＯＳへ戻る処理           =
; ===============================================
```

```
                                        ;
DOS:
                MOV     AH,4CH
                INT     21H
                                        ;
                                        ;
; ============================================
; =          メッセージの画面表示処理          =
; ============================================
                                        ;
MESSAGE:
                MOV     AH,9                    ; DXｱﾄﾞﾚｽから'$'迄を画面表示
                INT     21H
                RET
                                        ;
                                        ;
; ============================================
; =          リンクソフト呼出処理              =
; ============================================
                                        ;
COMMAND:
                INT     60H                     ; ｼｽﾃﾑｺｰﾙ
                CMP     AX,0                    ; 正常終了?
                JE      CMD_ED                  ; YES===>CMD_ED へ
                                                ;
                CMP     AX,133H                 ; 受信ｾﾝｽ出来ず?
                JE      CMD_ER                  ; YES===>CMD_ER へ
                                                ; ...NO
                PUSH    AX                      ; ｴﾗｰｺｰﾄﾞ待避
                CALL    BI2ASC                  ; ｴﾗｰｺｰﾄﾞ ASCII変換
                MOV     DX,OFFSET MSG3          ; ｴﾗｰﾒｯｾｰｼﾞ画面表示
                CALL    MESSAGE
                POP     AX                      ; ｴﾗｰｺｰﾄﾞ復帰
CMD_ER:
                STC
CMD_ED:
                RET
                                        ;
                                        ;
; ============================================
; =        エラーコードのＡＳＣＩＩ変換       =
; ============================================
                                        ;
BI2ASC:
                MOV     BL,AH
                MOV     AH,AL
                MOV     BH,BL
                AND     AL,0FH
                AND     BL,0FH
                MOV     CL,4
                SHR     AH,CL
                SHR     BH,CL
                ADD     BX,3030H
                ADD     AX,3030H
                CMP     AL,3AH
                JB      B0
                ADD     AL,7
B0:
                CMP     AH,3AH
                JB      B1
                ADD     AH,7
B1:
                CMP     BL,3AH
                JB      B2
                ADD     BL,7
B2:
                CMP     BH,3AH
                JB      B3
                ADD     BH,7
B3:
                MOV     [MSG3_1+0],BH
                MOV     [MSG3_1+1],BL
                MOV     [MSG3_1+2],AH
                MOV     [MSG3_1+3],AL
                RET
                                        ;
```

```
                                        ;
; ========================================
; =        ＢＩＮ==>ＡＳＣＩＩ変換        =
; ========================================
                                        ;
BIN_ASC:                                ; IN...AL=BINｺｰﾄﾞ
                PUSH    SI              ; OUT..AX=ASCIIｺｰﾄﾞ
                PUSH    BX
                PUSH    DX
                                        ;
                MOV     BL,AL
                AND     AL,0F0H         ; 上位４ビット
                SHR     AL,1
                SHR     AL,1
                SHR     AL,1
                SHR     AL,1
                LEA     SI,BINTOASC_TB  ; 変換テーブル
                MOV     AH,0
                ADD     SI,AX
                MOV     DH,BYTE PTR CS:[SI]
                                        ;
                MOV     AL,BL
                AND     AL,0FH          ; 下位４ビット
                LEA     SI,BINTOASC_TB  ; 変換テーブル
                MOV     AH,0
                ADD     SI,AX
                MOV     AL,BYTE PTR CS:[SI]
                MOV     AH,DH
                XCHG    AL,AH
                                        ;
                POP     DX
                POP     BX
                POP     SI
                RET
                                        ;
                                        ;
; ----------------------------------------
;       ＨＥＸ→ＡＳＣＩＩ　変換テーブル
; ----------------------------------------
                                        ;
BINTOASC_TB     DB      "0123456789ABCDEF"
                                        ;
                                        ;
; ========================================
; =              受信データ表示           =
; ========================================
                                        ;
RXDATA_DSP:
                MOV     DX,OFFSET DSP_RXSIZE ; 受信ｻｲｽﾞ表示
                CALL    MESSAGE
                MOV     BX,CS:[RXSIZE]
                LEA     DI,JBUF
                MOV     BYTE PTR CS:[DI+BX],CR
                MOV     BYTE PTR CS:[DI+BX+1],LF
                MOV     BYTE PTR CS:[DI+BX+2],'$'
                MOV     AL,BH
                CALL    BIN_ASC
                CALL    BIN_DSP
                MOV     AL,BL
                CALL    BIN_ASC
                CALL    BIN_DSP
                MOV     DX,OFFSET CRLF
                CALL    MESSAGE
                                        ;
                MOV     DX,OFFSET DSP_RXNO  ; 送信元No.表示
                CALL    MESSAGE
                MOV     AL,CS:[RXNO]
                CALL    BIN_ASC
                CALL    BIN_DSP
                MOV     DX,OFFSET CRLF
                CALL    MESSAGE
                                        ;
                MOV     DX,OFFSET DSP_DATA  ; 受信データ表示
                CALL    MESSAGE
                MOV     DX,OFFSET JBUF
                CALL    MESSAGE
```

```
                                        ;
                RET
                                        ;
                                        ;
BIN_DSP:                                ; IN...AX=表示文字ｺｰﾄﾞ
                PUSH    AX
                MOV     DL,AL
                MOV     AH,2
                INT     21H
                POP     AX
                MOV     DL,AH
                MOV     AH,2
                INT     21H
                RET
                                        ;
                                        ;
; ========================================
; =          Ｉ／Ｏパラメータ           =
; ========================================
                                        ; = 初期設定 =
INIT            DW      1000H           ; 初期設定ｺﾏﾝﾄﾞｺｰﾄﾞ
                DB      0               ; ｱﾌﾟﾘｹｰｼｮﾝ･ｿﾌﾄﾓｰﾄﾞ
                DB      1               ; 使用ﾎﾞｰﾄﾞ枚数
                                        ;
                                        ; = ﾎﾞｰﾄﾞ登録･起動 =
BOARD           DW      1100H           ; ﾎﾞｰﾄﾞ登録･起動ｺﾏﾝﾄﾞｺｰﾄﾞ
                DB      0               ; ﾎﾞｰﾄﾞ登録No.
                DB      27              ; 使用ｾｸﾞﾒﾝﾄｱﾄﾞﾚｽNo.27(D0000H)
                DB      0               ; 割り込みNo.
                DB      0
                                        ; = ｼﾘｱﾙ伝送単一接続登録 =
BSSG_ST         DW      1802H           ; ｺﾏﾝﾄﾞｺｰﾄﾞ
                DB      0               ; 登録ﾎﾞｰﾄﾞNo.
                DB      0               ; ﾕｰｻﾞｰID No.
                DB      2               ; 動作ﾓｰﾄﾞ(=単一接続)
                DB      0AH             ; ﾀｲﾑｱｳﾄ値(10*0.5=5s)
                DB      1               ; 再送指定(有り)
                DW      0               ; 再送回数(無限)
                DB      64              ; 通信先ﾕﾆｯﾄNo.
                DB      0FH             ; 通信先CH No.(I/Fﾎﾞｰﾄﾞﾊﾞｽ側)
                DB      0               ; 通信先ID No.
                DB      0               ; 通信先階層(無し)
                DB      6 DUP(0)
                                        ;
                                        ;
                                        ; = ｼﾘｱﾙﾃﾞｰﾀWRITE =
MEWWT           DW      1821H           ; ｺﾏﾝﾄﾞｺｰﾄﾞ
                DB      0               ; 登録ﾎﾞｰﾄﾞNO.
                DB      0               ; ﾕｰｻﾞｰID NO.
TXBUFF_OFF      DW      0               ; 送信ﾊﾞｯﾌｧOFFSET
                DW      SEG SBUF0
                DW      0
TXSIZE          DW      0               ; 送信ﾃﾞｰﾀ･ｻｲｽﾞ
                DW      18 DUP(0)       ; ｼｽﾃﾑ使用領域用確保
                                        ;
                                        ;
                                        ; = ｼﾘｱﾙﾃﾞｰﾀREAD =
MEWRD           DW      1832H           ; ｺﾏﾝﾄﾞｺｰﾄﾞ(完了ｾﾝｽﾀｲﾌﾟ)
                DB      0               ; 登録ﾎﾞｰﾄﾞNO.
                DB      0               ; ﾕｰｻﾞｰID NO.
                DW      OFFSET JBUF     ; 受信ﾊﾞｯﾌｧOFFSET
MEWRD2          DW      SEG JBUF
                DW      2048            ; 受信ﾊﾞｯﾌｧｻｲｽﾞ
RXSIZE          DW      0               ; 受信ﾃﾞｰﾀｻｲｽﾞ用
                DW       5 DUP(0)       ; ｼｽﾃﾑ使用領域用確保
                DB      0               ;
RXNO            DB      0               ; 送信元No.
                DW      12 DUP(0)       ; ｼｽﾃﾑ使用領域用確保
                                        ;
                                        ;
; ========================================
; =        メッセージ ／ バッファ領域    =
; ========================================
                                        ;
MSG1            DB      '初期設定===>$'
MSG1_OK         DB      'ＯＫ！'
```

パソコン機種別変更箇所
（左記はＰＣ９８シリーズの場合）

```
                DB      CR,LF,'$'

MSG3            DB      'ｴﾗｰ.....ｺｰﾄﾞ='
MSG3_1          DB      0,0,0,0
                DB      CR,LF,'$'

MSG4            DB      '使用ボード登録 & 起動===>$'
MSG4_OK         DB      'ＯＫ！'
                DB      CR,LF,'$'

BSSG_MSG        DB      'シリアル伝送単一接続登録===>$'
BSSG_OK_MSG     DB      'ＯＫ！',CR,LF,'$'

TX_DATA_MSG     DB      '発行データ＝$'
TX_MSG          DB      'シリアルデータ発行===>$'
TX_OK_MSG       DB      'ＯＫ！',CR,LF,'$'

RX_MSG          DB      'シリアルデータ受信センス===>$'
RX_OK_MSG       DB      'ＯＫ！',CR,LF,'$'
RX_BD_MSG       DB      '受信センス出来ず！！',CR,LF,'$'

DSP_RXSIZE      DB      '受信ﾃﾞｰﾀｻｲｽﾞ (Hex)=$'
DSP_RXNO        DB      '送信元No.          =$'
DSP_DATA        DB      '受信データ         =$'

CRLF            DB      CR,LF,'$'

SBUF0           DB      'AAAAAAAAAA',CR,LF,'$'                          ; 送信ﾃﾞｰﾀ0
SBUF1           DB      'BBBBBBBBBBBBBBB',CR,LF,'$'                     ; 送信ﾃﾞｰﾀ1
SBUF2           DB      'CCCCCCCCCCCCCCCCCCC',CR,LF,'$'                 ; 送信ﾃﾞｰﾀ2
SBUF3           DB      'DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD',CR,LF,'$'    ; 送信ﾃﾞｰﾀ3

                                                ; 送信ﾃﾞｰﾀ ﾃｰﾌﾞﾙ
TX_DATA_TB      DW      10                      ; ﾃﾞｰﾀ0... 送信ｻｲｽﾞ
                DW      OFFSET SBUF0            ;          送信ﾊﾞｯﾌｧ
                DW      15                      ; ﾃﾞｰﾀ1... 送信ｻｲｽﾞ
                DW      OFFSET SBUF1            ;          送信ﾊﾞｯﾌｧ
                DW      20                      ; ﾃﾞｰﾀ2... 送信ｻｲｽﾞ
                DW      OFFSET SBUF2            ;          送信ﾊﾞｯﾌｧ
                DW      30                      ; ﾃﾞｰﾀ3... 送信ｻｲｽﾞ
                DW      OFFSET SBUF3            ;          送信ﾊﾞｯﾌｧ

JBUF            DB      2048 DUP(0)             ; 受信ﾊﾞｯﾌｧ

DATANO          DW      0                       ; 送信ﾃﾞｰﾀNo.

STACK           DW      256 DUP(0)              ; ｽﾀｯｸ領域
STPT            DB      0


CODE    ENDS
        END     START
```

```
; ********************************************************
; *              シリアル伝送機能　単一接続              *
; *                    <ABSSG2.ASM>                      *
; ********************************************************
; *                  ＰＣ９８シリーズ                    *
; *     スレーブ処理:                                    *
; *      ６３番機からのデータを受信したら応答を返します。*
; *      （受信各文字に＋２０ｈした文字を返却します。）  *
; *     本プログラムを起動するパソコンＩ／Ｆボードは     *
; *     ６４番機として下さい。                           *
; *     ６３番機側をABSSGﾀｲﾌﾟで起動させて下さい。        *
; *                                                      *
; *     使用コマンド：登録 =1802H(単一接続)              *
; *                    WRITE=1821H(ﾀｲﾑｱｳﾄ付き)          *
; *                    READ =1832H(完了ｾﾝｽﾀｲﾌﾟ)         *
; ********************************************************
                                        ;
CODE    SEGMENT
        ASSUME CS:CODE,DS:CODE,ES:CODE,SS:CODE


CR              EQU     0DH             ; CRｺｰﾄﾞ
LF              EQU     0AH             ; LFｺｰﾄﾞ
                                        ;
                                        ;
START:
                MOV     AX,CS
                MOV     DS,AX
                MOV     ES,AX
                MOV     SS,AX
                MOV     AX,OFFSET STPT
                MOV     SP,AX
                                        ;
; ----------------------------------------------
;                 初　期　設　定
; ----------------------------------------------
INI:
                MOV     DX,OFFSET MSG1          ; 初期設定メッセージ表示
                CALL    MESSAGE
                MOV     DX,OFFSET INIT          ; 初期設定用I/Oﾊﾟﾗﾒｰﾀ ｱﾄﾞﾚｽ設定
                CALL    COMMAND                 ; ﾘﾝｸﾘｿﾌﾄ呼出
                JNB     INI_OK
                JMP     DOS                     ; ｴﾗｰ時 MS-DOSへ戻る
                                        ;
INI_OK:
                MOV     DX,OFFSET MSG1_OK       ; 初期設定ＯＫﾒｯｾｰｼﾞ 表示
                CALL    MESSAGE
                                        ;
; ----------------------------------------------
;        ボ　ー　ド　登　録　・　起　動
; ----------------------------------------------
                                        ;
                MOV     DX,OFFSET MSG4          ; ﾎﾞｰﾄﾞ登録･起動メッセージ表示
                CALL    MESSAGE
                MOV     DX,OFFSET BOARD         ; ﾎﾞｰﾄﾞ登録用I/Oﾊﾟﾗﾒｰﾀ ｱﾄﾞﾚｽ
                CALL    COMMAND                 ; ﾘﾝｸﾘｿﾌﾄ呼出
                JNB     BDST_OK
                JMP     DOS                     ; ｴﾗｰ時 MS-DOSへ戻る
                                        ;
BDST_OK:
                MOV     DX,OFFSET MSG4_OK       ; ﾎﾞｰﾄﾞ登録&起動OKﾒｯｾｰｼﾞ 表示
                CALL    MESSAGE
                                        ;
; ----------------------------------------------
;               単一接続モード登録
; ----------------------------------------------
                                        ;
BSSGST:
                MOV     DX,OFFSET BSSG_MSG      ; 登録メッセージ
                CALL    MESSAGE
                MOV     DX,OFFSET BSSG_ST       ; 登録I/Oﾊﾟﾗﾒｰﾀｱﾄﾞﾚｽ
                CALL    COMMAND
                JNB     BSSGST_OK
                JMP     DOS                     ; ｴﾗｰ時 MS-DOSへ戻る
                                        ;
BSSGST_OK:
                MOV     DX,OFFSET BSSG_OK_MSG ; 登録OKﾒｯｾｰｼﾞ 表示
```

```
                CALL    MESSAGE
                                                ;
;  ---------------------------------------------
;               シリアルデータ READ
;  ---------------------------------------------
                                                ;
RXLP:
                MOV     DX,OFFSET RX_MSG        ; ｼﾘｱﾙﾃﾞｰﾀ受信ｾﾝｽﾒｯｾｰｼﾞ表示
                CALL    MESSAGE
                MOV     DX,OFFSET MEWRD         ; READ用I/Oﾊﾟﾗﾒｰﾀ ｱﾄﾞﾚｽ
                CALL    COMMAND
                JNB     RD_OK                   ; 受信センス===>RD_OK へ
                                                ; ...NO
                CMP     AX,133H                 ; 受信ｾﾝｽ出来ず?
                JNE     RXLP                    ; NO ===>RXLP へ
                                                ; ...YES
                MOV     DX,OFFSET RX_BD_MSG     ; 受信ｾﾝｽ出来ずﾒｯｾｰｼﾞ表示
                CALL    MESSAGE
                JMP     RXLP
                                                ;
RD_OK:                                          ; == 受信センス時 ==
                MOV     DX,OFFSET RX_OK_MSG     ; ＯＫﾒｯｾｰｼﾞ表示
                CALL    MESSAGE
                                                ;
                CALL    RXDATA_DSP              ; 受信内容表示
                                                ;
                MOV     CX,CS:[RXSIZE]
                LEA     SI,JBUF
                LEA     DI,SBUF
RD_LP:
                MOV     AL,CS:[SI]              ; AL=受信ﾃﾞｰﾀ
                ADD     AL,20H                  ; 受信ﾃﾞｰﾀ+20H
                MOV     CS:[DI],AL              ; <==送信ﾊﾞｯﾌｧへ転送
                INC     SI
                INC     DI
                LOOP    RD_LP
                                                ;
                MOV     BYTE PTR CS:[DI],CR
                MOV     BYTE PTR CS:[DI+1],LF
                MOV     BYTE PTR CS:[DI+2],'$'
                                                ;
;  ---------------------------------------------
;               シリアルデータ WRITE
;  ---------------------------------------------
                                                ;
                MOV     AX,CS:[RXSIZE]
                MOV     CS:[TXSIZE],AX          ; 送信ﾃﾞｰﾀ･ｻｲｽﾞ ｾｯﾄ
                MOV     DX,OFFSET TX_DATA_MSG   ; 発行ﾃﾞｰﾀ表示
                CALL    MESSAGE
                MOV     DX,OFFSET SBUF
                CALL    MESSAGE
                                                ;
                MOV     DX,OFFSET TX_MSG        ; ｼﾘｱﾙﾃﾞｰﾀ発行ﾒｯｾｰｼﾞ表示
                CALL    MESSAGE
                MOV     DX,OFFSET MEWWT                 ; WRITEI/Oﾊﾟﾗﾒｰﾀ ｱﾄﾞﾚｽ
                CALL    COMMAND
                JB      TX_ED                   ; ｴﾗｰ時 再送
                                                ;
                MOV     DX,OFFSET TX_OK_MSG     ; ＯＫﾒｯｾｰｼﾞ表示
                CALL    MESSAGE
TX_ED:
                MOV     DX,OFFSET CRLF          ; 改行復帰
                CALL    MESSAGE
                JMP     RXLP
                                                ;
;  =============================================
;  =            ＭＳ－ＤＯＳへ戻る処理            =
;  =============================================
                                                ;
DOS:
                MOV     AH,4CH
                INT     21H
                                                ;
                                                ;
;  =============================================
```

```
; =          メッセージの画面表示処理          =
; ============================================
                                        ;
MESSAGE:
            MOV     AH, 9               ; DXｱﾄﾞﾚｽから'$'迄を画面表示
            INT     21H
            RET
                                        ;
                                        ;
; ============================================
; =          リンクソフト呼出処理          =
; ============================================
                                        ;
COMMAND:
            INT     60H                 ; ｼｽﾃﾑｺｰﾙ
            CMP     AX, 0               ; 正常終了?
            JE      CMD_ED              ; YES===>CMD_ED へ
                                        ;
            CMP     AX, 133H            ; 受信ｾﾝｽ出来ず?
            JE      CMD_ER              ; YES===>CMD_ER へ
                                        ; ...NO
            PUSH    AX                  ; ｴﾗｰｺｰﾄﾞ待避
            CALL    BI2ASC              ; ｴﾗｰｺｰﾄﾞ ASCII変換
            MOV     DX, OFFSET MSG3     ; ｴﾗｰﾒｯｾｰｼﾞ画面表示
            CALL    MESSAGE
            POP     AX                  ; ｴﾗｰｺｰﾄﾞ復帰
CMD_ER:
            STC
CMD_ED:
            RET
                                        ;
                                        ;
; ============================================
; =        エラーコードのＡＳＣＩＩ変換        =
; ============================================
                                        ;
BI2ASC:
            MOV     BL, AH
            MOV     AH, AL
            MOV     BH, BL
            AND     AL, 0FH
            AND     BL, 0FH
            MOV     CL, 4
            SHR     AH, CL
            SHR     BH, CL
            ADD     BX, 3030H
            ADD     AX, 3030H
            CMP     AL, 3AH
            JB      B0
            ADD     AL, 7
B0:
            CMP     AH, 3AH
            JB      B1
            ADD     AH, 7
B1:
            CMP     BL, 3AH
            JB      B2
            ADD     BL, 7
B2:
            CMP     BH, 3AH
            JB      B3
            ADD     BH, 7
B3:
            MOV     [MSG3_1+0], BH
            MOV     [MSG3_1+1], BL
            MOV     [MSG3_1+2], AH
            MOV     [MSG3_1+3], AL
            RET
                                        ;
                                        ;
; ============================================
; =        ＢＩＮ==>ＡＳＣＩＩ変換        =
; ============================================
                                        ;
BIN_ASC:                                ; IN...AL=BINｺｰﾄﾞ
            PUSH    SI                  ; OUT..AX=ASCIIｺｰﾄﾞ
```

```
                PUSH    BX
                PUSH    DX
                                        ;
                MOV     BL,AL
                AND     AL,0F0H         ; 上位４ビット
                SHR     AL,1
                SHR     AL,1
                SHR     AL,1
                SHR     AL,1
                LEA     SI,BINTOASC_TB  ; 変換テーブル
                MOV     AH,0
                ADD     SI,AX
                MOV     DH,BYTE PTR CS:[SI]
                                        ;
                MOV     AL,BL
                AND     AL,0FH          ; 下位４ビット
                LEA     SI,BINTOASC_TB  ; 変換テーブル
                MOV     AH,0
                ADD     SI,AX
                MOV     AL,BYTE PTR CS:[SI]
                MOV     AH,DH
                XCHG    AL,AH
                                        ;
                POP     DX
                POP     BX
                POP     SI
                RET
                                        ;
                                        ;
;  ---------------------------------------------
;         ＨＥＸ→ＡＳＣＩＩ　変換テーブル
;  ---------------------------------------------
                                        ;
BINTOASC_TB     DB      "0123456789ABCDEF"
                                        ;
                                        ;
;  =============================================
;  =                受信データ表示               =
;  =============================================
                                        ;
RXDATA_DSP:
                MOV     DX,OFFSET DSP_RXSIZE ; 受信ｻｲｽﾞ表示
                CALL    MESSAGE
                MOV     BX,CS:[RXSIZE]
                LEA     DI,JBUF
                MOV     BYTE PTR CS:[DI+BX],CR
                MOV     BYTE PTR CS:[DI+BX+1],LF
                MOV     BYTE PTR CS:[DI+BX+2],'$'
                MOV     AL,BH
                CALL    BIN_ASC
                CALL    BIN_DSP
                MOV     AL,BL
                CALL    BIN_ASC
                CALL    BIN_DSP
                MOV     DX,OFFSET CRLF
                CALL    MESSAGE
                                        ;
                MOV     DX,OFFSET DSP_RXNO   ; 送信元No.表示
                CALL    MESSAGE
                MOV     AL,CS:[RXNO]
                CALL    BIN_ASC
                CALL    BIN_DSP
                MOV     DX,OFFSET CRLF
                CALL    MESSAGE
                                        ;
                MOV     DX,OFFSET DSP_DATA   ; 受信データ表示
                CALL    MESSAGE
                MOV     DX,OFFSET JBUF
                CALL    MESSAGE
                                        ;
                RET
                                        ;
                                        ;
BIN_DSP:                                ; IN...AX=表示文字ｺｰﾄﾞ
                PUSH    AX
                MOV     DL,AL
```

```
                MOV     AH,2
                INT     21H
                POP     AX
                MOV     DL,AH
                MOV     AH,2
                INT     21H
                RET
                                        ;
                                        ;
;  ============================================
;  =              I／Oパラメータ                =
;  ============================================
                                        ; = 初期設定 =
INIT            DW      1000H           ; 初期設定ｺﾏﾝﾄﾞｺｰﾄﾞ
                DB      0               ; ｱﾌﾟﾘｹｰｼｮﾝ･ｿﾌﾄﾓｰﾄﾞ
                DB      1               ; 使用ﾎﾞｰﾄﾞ枚数
                                        ;
                                        ; = ﾎﾞｰﾄﾞ登録･起動 =
BOARD           DW      1100H           ; ﾎﾞｰﾄﾞ登録･起動ｺﾏﾝﾄﾞｺｰﾄﾞ
                DB      0               ; ﾎﾞｰﾄﾞ登録No.
                DB      27              ; 使用ｾｸﾞﾒﾝﾄｱﾄﾞﾚｽNo.27(D0000H)
                DB      0               ; 割り込みNo.
                DB      0
                                        ; = ｼﾘｱﾙ伝送単一接続登録 =
BSSG_ST         DW      1802H           ; ｺﾏﾝﾄﾞｺｰﾄﾞ
                DB      0               ; 登録ﾎﾞｰﾄﾞNo.
                DB      0               ; ﾕｰｻﾞｰID No.
                DB      2               ; 動作ﾓｰﾄﾞ(=単一接続)
                DB      0AH             ; ﾀｲﾑｱｳﾄ値(10*0.5=5s)
                DB      1               ; 再送指定(有り)
                DW      0               ; 再送回数(無限)
                DB      63              ; 通信先ﾕﾆｯﾄNo.
                DB      0FH             ; 通信先CH No.(I/Fﾎﾞｰﾄﾞﾊﾞｽ側)
                DB      0               ; 通信先ID No.
                DB      0               ; 通信先階層(無し)
                DB      6 DUP(0)
                                        ;
                                        ;
                                        ; = ｼﾘｱﾙﾃﾞｰﾀWRITE =
MEWWT           DW      1821H           ; ｺﾏﾝﾄﾞｺｰﾄﾞ
                DB      0               ; 登録ﾎﾞｰﾄﾞNO.
                DB      0               ; ﾕｰｻﾞｰID NO.
TXBUFF_OFF      DW      OFFSET SBUF     ; 送信ﾊﾞｯﾌｧOFFSET
                DW      SEG SBUF
                DW      0
TXSIZE          DW      0               ; 送信ﾃﾞｰﾀ･ｻｲｽﾞ
                DW      18 DUP(0)       ; ｼｽﾃﾑ使用領域用確保
                                        ;
                                        ;
                                        ; = ｼﾘｱﾙﾃﾞｰﾀREAD =
MEWRD           DW      1832H           ; ｺﾏﾝﾄﾞｺｰﾄﾞ(完了ｾﾝｽﾀｲﾌﾟ)
                DB      0               ; 登録ﾎﾞｰﾄﾞNO.
                DB      0               ; ﾕｰｻﾞｰID NO.
                DW      OFFSET JBUF     ; 受信ﾊﾞｯﾌｧOFFSET
MEWRD2          DW      SEG JBUF
                DW      2048            ; 受信ﾊﾞｯﾌｧｻｲｽﾞ
RXSIZE          DW      0               ; 受信ﾃﾞｰﾀｻｲｽﾞ用
                DW       5 DUP(0)       ;  ｼｽﾃﾑ使用領域用確保
                DB      0               ;
RXNO            DB      0               ; 送信元No.
                DW      12 DUP(0)       ; ｼｽﾃﾑ使用領域用確保
                                        ;
                                        ;
;  ============================================
;  =          メッセージ ／ バッファ領域          =
;  ============================================
                                        ;
MSG1            DB      '初期設定===>$'
MSG1_OK         DB      'ＯＫ！'
                DB      CR,LF,'$'

MSG3            DB      'ｴﾗｰ.....ｺｰﾄﾞ='
MSG3_1          DB      0,0,0,0
                DB      CR,LF,'$'
```

パソコン機種別変更箇所
（左記はＰＣ９８シリーズの場合）

```
MSG4            DB      '使用ボード登録 & 起動===>$'
MSG4_OK         DB      'ＯＫ！'
                DB      CR,LF,'$'

BSSG_MSG        DB      'シリアル伝送単一接続登録===>$'
BSSG_OK_MSG     DB      'ＯＫ！',CR,LF,'$'

TX_DATA_MSG     DB      '発行データ＝$'
TX_MSG          DB      'シリアルデータ発行===>$'
TX_OK_MSG       DB      'ＯＫ！',CR,LF,'$'

RX_MSG          DB      'シリアルデータ受信センス===>$'
RX_OK_MSG       DB      'ＯＫ！',CR,LF,'$'
RX_BD_MSG       DB      '受信センス出来ず！！',CR,LF,'$'

DSP_RXSIZE      DB      '受信ﾃﾞｰﾀｻｲｽﾞ(Hex)=$'
DSP_RXNO        DB      '送信元No.          =$'
DSP_DATA        DB      '受信データ         =$'

CRLF            DB      CR,LF,'$'

SBUF            DB      2048 DUP(0)                     ; 送信ﾊﾞｯﾌｧ
JBUF            DB      2048 DUP(0)                     ; 受信ﾊﾞｯﾌｧ

STACK           DW      256 DUP(0)                      ; ｽﾀｯｸ領域
STPT            DB      0


                CODE    ENDS
                END     START
```

```
; ********************************************************
; *                シリアル伝送機能　多重接続            *
; *                      <ABSML.ASM>                     *
; ********************************************************
; *                    ＰＣ９８シリーズ                  *
; *     マスター処理:                                    *
; *     各スレーブ局（１～３番機）に対して、下記処理を   *
; *     実施します。                                     *
; *        ・各スレーブ局受信データの読出しを行います。  *
; *        ・受信データが有れば、同一データを返却します。*
; *                                                      *
; *     使用コマンド：登録 =1802H(多重ﾏｽﾀｰ接続ﾓｰﾄﾞ)      *
; *                    WRITE=1811H(ﾀｲﾑｱｳﾄ付き)          *
; *                    READ =1821H(ﾀｲﾑｱｳﾄ付き)          *
; ********************************************************
                                                ;
CODE    SEGMENT
        ASSUME CS:CODE,DS:CODE,ES:CODE,SS:CODE


CR              EQU     0DH                     ; CRｺｰﾄﾞ
LF              EQU     0AH                     ; LFｺｰﾄﾞ
                                                ;
                                                ;
START:
                MOV     AX,CS
                MOV     DS,AX
                MOV     ES,AX
                MOV     SS,AX
                MOV     AX,OFFSET STPT
                MOV     SP,AX
                                                ;
; ---------------------------------------------
;                 初　期　設　定
; ---------------------------------------------
INI:
                MOV     DX,OFFSET MSG1          ; 初期設定メッセージ表示
                CALL    MESSAGE
```

```
                MOV     DX,OFFSET INIT          ; 初期設定用I/Oﾊﾟﾗﾒｰﾀ ｱﾄﾞﾚｽ設定
                CALL    COMMAND                 ; ﾘﾝｸｿﾌﾄ呼出
                JNB     INI_OK
                JMP     DOS                     ; ｴﾗｰ時 MS-DOSへ戻る
                                                ;
INI_OK:
                MOV     DX,OFFSET OK_MSG        ; 初期設定ＯＫﾒｯｾｰｼﾞ 表示
                CALL    MESSAGE
                                                ;
;  ---------------------------------------------
;          ボ ー ド 登 録 ・ 起 動
;  ---------------------------------------------
                                                ;
                MOV     DX,OFFSET MSG4          ; ﾎﾞｰﾄﾞ登録･起動メッセージ表示
                CALL    MESSAGE
                MOV     DX,OFFSET BOARD         ; ﾎﾞｰﾄﾞ登録用I/Oﾊﾟﾗﾒｰﾀ ｱﾄﾞﾚｽ
                CALL    COMMAND                 ; ﾘﾝｸｿﾌﾄ呼出
                JNB     BDST_OK
                JMP     DOS                     ; ｴﾗｰ時 MS-DOSへ戻る
                                                ;
BDST_OK:
                MOV     DX,OFFSET OK_MSG        ; ﾎﾞｰﾄﾞ登録&起動OKﾒｯｾｰｼﾞ 表示
                CALL    MESSAGE
                                                ;
;  ---------------------------------------------
;          多重接続（マスター）モード登録
;  ---------------------------------------------
                                                ;
BSSGST:
                MOV     DX,OFFSET BSSG_MSG      ; 登録メッセージ
                CALL    MESSAGE
                MOV     DX,OFFSET BSSG_ST       ; 登録I/Oﾊﾟﾗﾒｰﾀｱﾄﾞﾚｽ
                CALL    COMMAND
                JNB     BSSGST_OK
                JMP     DOS                     ; ｴﾗｰ時 MS-DOSへ戻る
                                                ;
BSSGST_OK:
                MOV     DX,OFFSET OK_MSG        ; 登録OKﾒｯｾｰｼﾞ 表示
                CALL    MESSAGE
                                                ;
                MOV     BYTE PTR CS:[TXNO],3    ; 接続局No.用初期値
                                                ;
                                                ;
;  ---------------------------------------------
;          制御コマンド発行（スレーブ局指定）
;  ---------------------------------------------
                                                ;
LP:
                MOV     DX,OFFSET CRLF          ; 改行復帰
                CALL    MESSAGE
                INC     BYTE PTR CS:[TXNO]      ; 接続局変更
                CMP     BYTE PTR CS:[TXNO],4    ; 4番機?
                JNE     CNTCM_10                ; NO ===>CNTCM_10 へ
                                                ; ...YES
                MOV     BYTE PTR CS:[TXNO],1    ; 1番機に戻す
CNTCM_10:
                MOV     BL,CS:[TXNO]
                XOR     BH,BH
                DEC     BX
                SHL     BX,1
                SHL     BX,1
                SHL     BX,1
                LEA     DI,TXDTTB
                MOV     DX,OFFSET SRVST_MSG             ; ｽﾚｰﾌﾞ局接続ﾒｯｾｰｼﾞ 表示
                CALL    MESSAGE
                MOV     DL,CS:[DI+BX]                   ; DL=局番
                CALL    BIN_DSP_CHR                     ; 局番表示
                MOV     DX,OFFSET SRVST_MSG2
                CALL    MESSAGE
                MOV     AX,CS:[DI+BX+2]                 ; 接続用ｺﾏﾝﾄﾞ格納位置
                MOV     CS:[CNTOFF],AX
                MOV     WORD PTR CS:[CNTSIZE],5         ; ﾃﾞｰﾀｻｲｽﾞ ｾｯﾄ
                CALL    CNTCMD_TX               ; 制御コマンド発行
                JNB     DTRD                    ; OK ===>DTRD へ
RSPER:                                          ; ...ｴﾗｰ
                JMP     LP
```

```
                                        ;
; ---------------------------------------
;       制御コマンド発行（データ読出指定）
; ---------------------------------------
                                        ;
DTRD:
            MOV     DX,OFFSET  RX_MSG   ; ﾃﾞｰﾀ読出ﾒｯｾｰｼﾞ 表示
            CALL    MESSAGE
            LEA     AX,RDDT
            MOV     CS:[CNTOFF],AX
            MOV     WORD PTR CS:[CNTSIZE],3     ; ﾃﾞｰﾀｻｲｽﾞ ｾｯﾄ
            CALL    CNTCMD_TX           ; 制御コマンド発行
            JB      RSPER               ; ｴﾗｰ===>RSPER へ
                                        ; ...OK
            CALL    RXDATA_DSP          ; 受信内容表示
                                        ;
            MOV     CX,CS:[RXSIZE]
            ADD     CX,3                ; CR,LF,$用 +3ﾊﾞｲﾄ
            LEA     SI,JBUF
            LEA     DI,SBUF
            CLD
            REP     MOVSB               ; 受信ﾊﾞｯﾌｧ==>送信ﾊﾞｯﾌｧ転送
                                        ;
                                        ;
; ---------------------------------------
;           シリアルデータ ＷＲＩＴＥ
; ---------------------------------------
                                        ;
            MOV     AX,CS:[RXSIZE]
            MOV     CS:[TXSIZE],AX      ; 送信ﾃﾞｰﾀ･ｻｲｽﾞ ｾｯﾄ
            MOV     DX,OFFSET  TX_DATA_MSG ; 発行ﾃﾞｰﾀ表示
            CALL    MESSAGE
            MOV     DX,OFFSET  SBUF
            CALL    MESSAGE
                                        ;
            MOV     DX,OFFSET  TX_MSG   ; ｼﾘｱﾙﾃﾞｰﾀ発行ﾒｯｾｰｼﾞ 表示
            CALL    MESSAGE
            MOV     DX,OFFSET  BSWT     ; WRITEI/Oﾊﾟﾗﾒｰﾀ ｱﾄﾞﾚｽ
            CALL    COMMAND
            JB      TX_ED               ; ｴﾗｰ時 再送
                                        ;
            MOV     DX,OFFSET  OK_MSG   ; ＯＫﾒｯｾｰｼﾞ 表示
            CALL    MESSAGE
TX_ED:
            MOV     DX,OFFSET  CRLF     ; 改行復帰
            CALL    MESSAGE
            JMP     LP
                                        ;
; ===========================================
; =    制御コマンド発行処理                 =      OUTPUT
; ===========================================      CY=0...ﾚｽﾎﾟﾝｽOK
                                                 ; CY=1...ﾚｽﾎﾟﾝｽ異常
CNTCMD_TX:
            MOV     DX,OFFSET  CNTCMD   ; 制御ｺﾏﾝﾄﾞ発行用I/Oﾊﾟﾗﾒｰﾀ
            CALL    COMMAND
            JB      CNTCMD_TX_ED
            CALL    RSPCD_CK            ; ﾚｽﾎﾟﾝｽｺｰﾄﾞ･ﾁｪｯｸ
            JB      CNTCMD_TX_ED
            MOV     DX,OFFSET  OK_MSG   ; ＯＫ！ﾒｯｾｰｼﾞ 表示
            CALL    MESSAGE
            CLC
                                        ;
CNTCMD_TX_ED:
            RET
                                        ;
                                        ;
; ===========================================
; =          ＭＳ－ＤＯＳへ戻る処理        =
; ===========================================
                                        ;
DOS:
            MOV     AH,4CH
            INT     21H
                                        ;
                                        ;
```

```
; ============================================
; =          メッセージの画面表示処理          =
; ============================================
                                        ;
MESSAGE:
        PUSH    DI
        PUSH    BX
                                        ;
        MOV     AH, 9                   ; DXｱﾄﾞﾚｽから'$'迄を画面表示
        INT     21H
                                        ;
        POP     BX
        POP     DI
        RET
                                        ;
                                        ;
; ============================================
; =          リンクソフト呼出処理            =
; ============================================
                                        ;
COMMAND:
        PUSH    DI
        PUSH    BX
                                        ;
        INT     60H                     ; ｼｽﾃﾑｺｰﾙ
        CMP     AX, 0                   ; 正常終了?
        JE      CMD_ED                  ; YES===>CMD_ED へ
                                        ;
        CMP     AX, 133H                ; 受信ｾﾝｽ出来ず?
        JE      CMD_ER                  ; YES===>CMD_ER へ
                                        ; ...NO
        PUSH    AX                      ; ｴﾗｰｺｰﾄﾞ待避
        CALL    BI2ASC                  ; ｴﾗｰｺｰﾄﾞ ASCII変換
        MOV     DX, OFFSET MSG3         ; ｴﾗｰﾒｯｾｰｼﾞ画面表示
        CALL    MESSAGE
        POP     AX                      ; ｴﾗｰｺｰﾄﾞ復帰
CMD_ER:
        STC
CMD_ED:
        POP     BX
        POP     DI
        RET
                                        ;
                                        ;
; ============================================
; =        エラーコードのＡＳＣＩＩ変換      =
; ============================================
                                        ;
BI2ASC:
        MOV     BL, AH
        MOV     AH, AL
        MOV     BH, BL
        AND     AL, 0FH
        AND     BL, 0FH
        MOV     CL, 4
        SHR     AH, CL
        SHR     BH, CL
        ADD     BX, 3030H
        ADD     AX, 3030H
        CMP     AL, 3AH
        JB      B0
        ADD     AL, 7
B0:
        CMP     AH, 3AH
        JB      B1
        ADD     AH, 7
B1:
        CMP     BL, 3AH
        JB      B2
        ADD     BL, 7
B2:
        CMP     BH, 3AH
        JB      B3
        ADD     BH, 7
B3:
        MOV     [MSG3_1+0], BH
```

```
                MOV     [MSG3_1+1],BL
                MOV     [MSG3_1+2],AH
                MOV     [MSG3_1+3],AL
                RET
                                        ;
                                        ;
; ========================================
; =       ＢＩＮ==>ＡＳＣＩＩ変換       =
; ========================================
                                        ;
BIN_ASC:                                        ; IN...AL=BINｺｰﾄﾞ
                PUSH    SI                      ; OUT..AX=ASCIIｺｰﾄﾞ
                PUSH    BX
                PUSH    DX
                                        ;
                MOV     BL,AL
                AND     AL,0F0H                 ; 上位４ビット
                SHR     AL,1
                SHR     AL,1
                SHR     AL,1
                SHR     AL,1
                LEA     SI,BINTOASC_TB          ; 変換テーブル
                MOV     AH,0
                ADD     SI,AX
                MOV     DH,BYTE PTR CS:[SI]
                                        ;
                MOV     AL,BL
                AND     AL,0FH                  ; 下位４ビット
                LEA     SI,BINTOASC_TB          ; 変換テーブル
                MOV     AH,0
                ADD     SI,AX
                MOV     AL,BYTE PTR CS:[SI]
                MOV     AH,DH
                XCHG    AL,AH
                                        ;
                POP     DX
                POP     BX
                POP     SI
                RET
                                        ;
                                        ;
; ----------------------------------------
;       ＨＥＸ→ＡＳＣＩＩ　変換テーブル
; ----------------------------------------
                                        ;
BINTOASC_TB     DB      "0123456789ABCDEF"
                                        ;
                                        ;
; ========================================
; =       受信レスポンスコードチェック   =
; ========================================
                                        ;
RSPCD_CK:
                CMP     BYTE PTR CS:[JBUF],21H          ; 異常ﾚｽﾎﾟﾝｽ ?
                JNE     RSPCD_OK                        ; NO===>RSPCD_OK へ
                                        ; ...YES
                MOV     DX,OFFSET ERR_RSP_MSG ; ｴﾗｰﾚｽﾎﾟﾝｽﾒｯｾｰｼﾞ 表示
                CALL    MESSAGE
                LEA     DI,JBUF
                MOV     AX,CS:[DI+1]
                CALL    BIN_DSP
                MOV     DX,OFFSET CRLF          ; 改行復帰
                CALL    MESSAGE
                STC
                JMP     RSPCD_ED
RSPCD_OK:
                CLC
RSPCD_ED:
                RET
                                        ;
                                        ;
; ========================================
; =           受信データ表示             =
; ========================================
                                        ;
RXDATA_DSP:
```

```
                MOV     DX,OFFSET  DSP_RXSIZE  ; 受信ｻｲｽﾞ表示
                CALL    MESSAGE
                MOV     BX,CS:[RXSIZE]
                LEA     DI,JBUF
                MOV     BYTE PTR CS:[DI+BX],CR
                MOV     BYTE PTR CS:[DI+BX+1],LF
                MOV     BYTE PTR CS:[DI+BX+2],'$'
                MOV     AL,BH
                CALL    BIN_ASC
                CALL    BIN_DSP
                MOV     AL,BL
                CALL    BIN_ASC
                CALL    BIN_DSP
                MOV     DX,OFFSET  CRLF
                CALL    MESSAGE
                                        ;
                MOV     DX,OFFSET  DSP_RXNO     ; 送信元No.表示
                CALL    MESSAGE
                MOV     AL,CS:[RXNO]
                CALL    BIN_ASC
                CALL    BIN_DSP
                MOV     DX,OFFSET  CRLF
                CALL    MESSAGE
                                        ;
                MOV     DX,OFFSET  DSP_DATA     ; 受信データ表示
                CALL    MESSAGE
                MOV     DX,OFFSET  JBUF
                CALL    MESSAGE
                                        ;
                RET
                                        ;
                                        ;
BIN_DSP:                                ; IN...AX=表示文字ｺｰﾄﾞ
                PUSH    AX
                MOV     DL,AL
                MOV     AH,2
                INT     21H
                POP     AX
                MOV     DL,AH
BIN_DSP_CHR:
                MOV     AH,2
                INT     21H
                RET
                                        ;
                                        ;
;  ============================================
;  =            I／Oパラメータ               =
;  ============================================
                                        ; = 初期設定 =
INIT            DW      1000H           ; 初期設定ｺﾏﾝﾄﾞｺｰﾄﾞ
                DB      0               ; ｱﾌﾟﾘｹｰｼｮﾝ･ｿﾌﾄﾓｰﾄﾞ
                DB      1               ; 使用ﾎﾞｰﾄﾞ枚数
                                        ;
                                        ; = ﾎﾞｰﾄﾞ登録･起動 =
BOARD           DW      1100H           ; ﾎﾞｰﾄﾞ登録･起動ｺﾏﾝﾄﾞｺｰﾄﾞ
                DB      0               ; ﾎﾞｰﾄﾞ登録No.
                DB      27              ; 使用ｾｸﾞﾒﾝﾄｱﾄﾞﾚｽNo.27(D0000H)
                DB      0               ; 割り込みNO.
                DB      0
                                        ; = ｼﾘｱﾙ伝送多重接続登録 =
BSSG_ST         DW      1802H           ; ｺﾏﾝﾄﾞｺｰﾄﾞ
                DB      0               ; 登録ﾎﾞｰﾄﾞNo.
                DB      0               ; ﾕｰｻﾞｰID No.
                DB      0               ; 動作ﾓｰﾄﾞ(=多重接続ﾏｽﾀｰ)
                DB      0AH             ; ﾀｲﾑｱｳﾄ値(10*0.5=5s)
                DB      1               ; 再送指定(有り)
                DW      0               ; 再送回数(無限)
                DB      10 DUP(0)
                                        ;
                                        ;
                                        ; = 制御コマンド =
CNTCMD          DW      1811H           ; ｺﾏﾝﾄﾞｺｰﾄﾞ
                DB      0               ; 登録ﾎﾞｰﾄﾞNO.
                DB      0               ; ﾕｰｻﾞｰID NO.
CNTOFF          DW      0               ; 送信ﾊﾞｯﾌｧOFFSET
```

パソコン機種別変更箇所
（左記はＰＣ９８シリーズの場合）

```
                DW      SEG JBUF                ; 送信ﾊﾞｯﾌｧsegment
                DW      0
CNTSIZE         DW      0                       ; 送信ﾃﾞｰﾀ･ｻｲｽﾞ
                DW      5 DUP(0)                ; ｼｽﾃﾑ使用領域用確保
                DB      0
RXNO            DB      0                       ; 送信元No.
                DW      7 DUP(0)
                DW      OFFSET JBUF             ; 受信ﾊﾞｯﾌｧOFFSET
                DW      SEG JBUF                ; 受信ﾊﾞｯﾌｧSEGMENT
                DW      2048                    ; 受信ﾊﾞｯﾌｧｻｲｽﾞ
RXSIZE          DW      0                       ; 受信ﾃﾞｰﾀｻｲｽﾞ
                DW      0
                                                ;
                                                ;
                                                ; = ｼﾘｱﾙﾃﾞｰﾀWRITE =
BSWT            DW      1821H                   ; ｺﾏﾝﾄﾞｺｰﾄﾞ
                DB      0                       ; 登録ﾎﾞｰﾄﾞNO.
                DB      0                       ; ﾕｰｻﾞｰID NO.
                DW      OFFSET SBUF             ; 送信ﾊﾞｯﾌｧOFFSET
                DW      SEG SBUF
                DW      0
TXSIZE          DW      0                       ; 送信ﾃﾞｰﾀ･ｻｲｽﾞ
                DW      18 DUP(0)               ; ｼｽﾃﾑ使用領域用確保
                                                ;
                                                ;
; ============================================
; =          メッセージ ／ バッファ領域          =
; ============================================
                                                ;
MSG1            DB      '初期設定===>$'
OK_MSG          DB      'ＯＫ！',CR,LF,'$'

MSG3            DB      'ｴﾗｰ.....ｺｰﾄﾞ='
MSG3_1          DB      0,0,0,0
                DB      CR,LF,'$'

MSG4            DB      '使用ボード登録 & 起動===>$'

BSSG_MSG        DB      'シリアル伝送多重接続（マスター）登録===>$'

SRVST_MSG       DB      'スレーブ局（$'
SRVST_MSG2      DB      '番機）接続===>$'

TX_DATA_MSG     DB      '発行データ＝$'
TX_MSG          DB      'シリアルデータ発行===>$'

RX_MSG          DB      'データ読出===>$'

DSP_RXSIZE      DB      '受信ﾃﾞｰﾀｻｲｽﾞ(Hex)=$'
DSP_RXNO        DB      '送信元No.          =$'
DSP_DATA        DB      '受信データ         =$'

ERR_RSP_MSG     DB      'ｴﾗｰﾚｽﾎﾟﾝｽｺｰﾄﾞ=$'
CRLF            DB      CR,LF,'$'

SBUF            DB      2048 DUP(0)             ; 送信ﾊﾞｯﾌｧ
JBUF            DB      2048 DUP(0)             ; 受信ﾊﾞｯﾌｧ

TXDTTB          DB      31H                     ; 1番機用局番(文字)
                DB      0                       ; ﾀﾞﾐｰ
                DW      OFFSET SRVDT1           ;       接続用
                DB      32H                     ; 2番機用局番(文字)
                DB      0                       ; ﾀﾞﾐｰ
                DW      OFFSET SRVDT2           ;       接続用
                DB      33H                     ; 3番機用局番(文字)
                DB      0                       ; ﾀﾞﾐｰ
                DW      OFFSET SRVDT3           ;       接続用

SRVDT1          DB      '#S101'                 ; 1番機用接続ｺﾏﾝﾄﾞ
SRVDT2          DB      '#S102'                 ; 2番機用接続ｺﾏﾝﾄﾞ
SRVDT3          DB      '#S103'                 ; 3番機用接続ｺﾏﾝﾄﾞ

RDDT            DB      '#S2'                   ; ﾃﾞｰﾀ読出用ｺﾏﾝﾄﾞ

TXNO            DB      0                       ; 接続先No.用
```

```
STACK       DW      256 DUP(0)              ; ｽﾀｯｸ領域
STPT        DB      0


            CODE    ENDS
            END     START
```

パソコン機種別変更箇所

ＰＣ／ＡＴ互換機

```
BOARD       DW      1100H           ; ﾎﾞｰﾄﾞ登録･起動ｺﾏﾝﾄﾞｺｰﾄﾞ
            DB      0               ; ﾎﾞｰﾄﾞ登録No.
            DB      8               ; 使用ｾｸﾞﾒﾝﾄｱﾄﾞﾚｽﾄNo.
            DB      10              ; 割り込みNo.
            DB      0
```

ＦＭＲシリーズ

```
BOARD       DW      1100H           ; ﾎﾞｰﾄﾞ登録･起動ｺﾏﾝﾄﾞｺｰﾄﾞ
            DB      0               ; ﾎﾞｰﾄﾞ登録No.
            DB      7               ; 基板I/OﾎﾟｰﾄｱﾄﾞﾚｽNo.
            DW      7800H           ; ｺﾝﾄﾛｰﾙ/ｽﾃｰﾀｽI/Oﾎﾟｰﾄｱﾄﾞﾚｽ
            DB      5               ; 割り込みNo.
            DB      0
```

Cプログラム（ソフトウェア割り込みint60）

```
/*****************************************************************/
/*                  シリアル伝送機能 単一接続                    */
/*                          (BSSG.C)                             */
/*****************************************************************/
/*                    ＰＣ９８シリーズ                           */
/*      マスター処理:                                            */
/*      ６４番機(ﾊﾟｿｺﾝI/Fﾎﾞｰﾄﾞ) へ文字データを送信します。      */
/*      ６４番機から返信があれば発行データを順次変更して        */
/*      いきます。                                               */
/*      ６４番機をBSSG2タイプで起動させて下さい。               */
/*                                                               */
/*       使用コマンド：登録 =1802H(単一接続ﾓｰﾄﾞ)                 */
/*                     WRITE=1821H(ﾀｲﾑｱｳﾄ 付き)                 */
/*                     READ =1832H(完了ｾﾝｽﾀｲﾌﾟ)                 */
/*****************************************************************/

#include  <dos.h>
#include  <stdio.h>
#include  <stdlib.h>

#define PC_N  64                        /* 通信局番 64 */

static union REGS inregs;
static union REGS outregs;
struct SREGS segregs;
static char far *p1;
static unsigned char dat[48];
unsigned char sbuf[2048];               /* 送信ﾊﾞｯﾌｧ   */
unsigned char jbuf[2048];               /* 受信ﾊﾞｯﾌｧ   */

void main( void )
{
        int     i;
        int     err;
        long    lwork;
        char    cwork[5];
        char    c_dmy1[4];
        char    *dmy;
        int     datano,rxsize,txsize;

/* -------------------------*/
/* ------ 初期設定処理 ------*/
/* -------------------------*/

        dat[0]  = 0x00;         /* 初期設定ｺﾏﾝﾄﾞ･ｺｰﾄﾞ=1000H  */
        dat[1]  = 0x10;
        dat[2]  = 0;            /* ｱﾌﾟﾘｹｰｼｮﾝ･ｿﾌﾄﾓｰﾄﾞ=0       */
        dat[3]  = 1;            /* 使用ボード枚数=1          */

        printf("初期設定===>");

        if((err = mew_send())  != 0)
        {
                printf("ｴﾗｰ.....ｺｰﾄﾞ=  %x¥n",  err);
                exit(-1);
        }
        printf("ＯＫ！¥n");

/* ---------------------------------*/
/* ------ 使用ボード登録 及び起動 ------*/
/* ---------------------------------*/

        dat[0]  = 0x00;         /* 使用ボード登録及び起動    */
        dat[1]  = 0x11;         /* ｺﾏﾝﾄﾞｺｰﾄﾞ=1100H          */
        dat[2]  = 0x00;         /* 登録ボードNo.=0           */
        dat[3]  = 27;           /* 使用ｾｸﾞﾒﾝﾄｱﾄﾞﾚｽNo.=27     */
        dat[4]  = 0;            /* 割り込みNo.=0             */

        printf("使用ボード登録 & 起動===>");
```

パソコン機種別変更箇所
（左記はＰＣ９８シリーズの場合）

```
        if((err = mew_send()) != 0)
        {
                printf("ｴﾗｰ.....ｺｰﾄﾞ= %x¥n", err);
                exit(-1);
        }
        printf("ＯＫ！¥n");

/* ---------------------------------*/
/* ------ 単一接続モード登録 ------*/
/* ---------------------------------*/

        dat[0] = 0x02;          /* ｼﾘｱﾙ伝送単一接続ﾓｰﾄﾞ  */
        dat[1] = 0x18;          /* 登録ｺﾏﾝﾄﾞｺｰﾄﾞ=1802H  */
        dat[2] = 0x00;          /* 登録ﾎﾞｰﾄﾞNo.=0        */
        dat[3] = 0x00;          /* ﾕｰｻﾞｰID No.=0(固定)  */
        dat[4] = 0x02;          /* 動作ﾓｰﾄﾞ=単一接続     */
        dat[5] = 0x0a;          /* ﾀｲﾑｱｳﾄ値=10*0.5(5s)  */
        dat[6] = 0x01;          /* 再送指定=有り         */
        *((unsigned *)&dat[7])= 0x00;       /* 再送回数=無限          */
        dat[9] = PC_N;          /* 送信先ﾕﾆｯﾄNo.=64      */
        dat[10]= 0x0F;          /* 送信先CH No.=F        */
        dat[11]= 0x00;          /* 送信先ID No.=0(固定) */
        dat[12]= 0x00;          /* 伝送経路階層=0(無し) */

        printf("シリアル伝送単一接続登録===>");

        if((err = mew_send()) != 0)
        {
                printf("ｴﾗｰ.....ｺｰﾄﾞ= %x ¥n", err);
                exit(-1);
        }

        printf("ＯＫ！¥n");

/* ------------------------------------*/
/* ------ シリアルデータＷＲＩＴ ------*/
/* ------------------------------------*/

        datano = 0;
TXLP:
        memset( sbuf, 0, sizeof( sbuf ) );
        switch(datano)
        {
                case 0:
                        sprintf(sbuf,"AAAAAAAAAA");
                        txsize = 10;
                        break;
                case 1:
                        sprintf(sbuf,"BBBBBBBBBBBBBB");
                        txsize = 15;
                        break;
                case 2:
                        sprintf(sbuf,"CCCCCCCCCCCCCCCCCCC");
                        txsize = 20;
                        break;
                case 3:
                        sprintf(sbuf,"DDDDDDDDDDDDDDDDDDDDDDDDDDDDDD");
                        txsize = 30;
                        break;
        }
TXLP2:
        printf("発行データ=%s¥n",sbuf);
        p1 = (char far *)&sbuf[0];

        dat[0] = 0x21;          /* ｼﾘｱﾙﾃﾞｰﾀWRITE       */
        dat[1] = 0x18;          /* ｺﾏﾝﾄﾞｺｰﾄﾞ=1821H     */
        dat[2] = 0x00;          /* 登録ボードNo.=0     */
        dat[3] = 0x00;          /* 0固定               */
        *((unsigned *)&dat[4]) = FP_OFF(p1);        /* 送信ﾊﾞｯﾌｧｵﾌｾｯﾄ    */
        *((unsigned *)&dat[6]) = FP_SEG(p1);        /* 送信ﾊﾞｯﾌｧｾｸﾞﾒﾝﾄ   */
        *((unsigned *)&dat[10]) = txsize;           /* 送信ﾃﾞｰﾀｻｲｽﾞ       */

        printf("シリアルデータ発行===>");

        if((err = mew_send()) != 0)
        {
```

```
                printf("ｴﾗｰ.....ｺｰﾄﾞ= %x ¥n", err);
                goto TXLP2;
        }

        printf("ＯＫ！¥n");


/* ------------------------------------*/
/* ------ シリアルデータＲＥＡＤ ------*/
/* ------------------------------------*/

        memset( jbuf, 0, sizeof( jbuf ) );
        p1 = (char far *)&jbuf[0];

        dat[0] = 0x32;          /* ｼﾘｱﾙﾃﾞｰﾀREAD     */
        dat[1] = 0x18;          /* ｺﾏﾝﾄﾞｺｰﾄﾞ=1832H  */
        dat[2] = 0x00;          /* 登録ボードno.=0  */
        dat[3] = 0x00;          /* 0固定            */
        *((unsigned *)&dat[4]) = FP_OFF(p1);        /* 受信ﾊﾞｯﾌｧｵﾌｾｯﾄ   */
        *((unsigned *)&dat[6]) = FP_SEG(p1);        /* 受信ﾊﾞｯﾌｧｾｸﾞﾒﾝﾄ  */
        *((unsigned *)&dat[8]) = 2048;              /* 受信ﾊﾞｯﾌｧｻｲｽﾞ     */

        printf("シリアルデータ受信センス===>");

        if((err = mew_send()) != 0)
        {
                if(err == 0x133)
                        printf("受信センス出来ず！！¥n");
                else
                        printf("ｴﾗｰ.....ｺｰﾄﾞ= %x¥n", err);
        }
        else
        {
                printf("ＯＫ！¥n");

                rxsize = dat[10] + dat[11] * 0x100;
                printf("受信ﾃﾞｰﾀｻｲｽﾞ=%d¥n",rxsize);
                printf("送信元No.    =%d¥n",dat[23]);
                printf("受信データ   =%s¥n",jbuf);

                if(datano == 3)
                        datano = 0;
                else
                        datano = datano + 1;
        }
        printf("¥n");
        goto TXLP;
}


/* ---------------------------*/
/* ------ ﾘﾝｸﾘｿﾌﾄ呼び出し------*/
/* ---------------------------*/
int mew_send(void)
{

        p1 = (char far *)&dat[0];

        inregs.x.dx = FP_OFF(p1);
        segregs.ds = FP_SEG(p1);

        int86x( 0x60, &inregs, &outregs, &segregs);
        return(outregs.x.ax);
}
```

```
/*************************************************************/
/*                シリアル伝送機能 単一接続                  */
/*                        (BSSG2.C)                          */
/*************************************************************/
/*                  ＰＣ９８シリーズ                         */
/*      スレーブ処理:                                        */
/*      ６３番機からのデータを受信したら応答を返します。     */
/*      （受信各文字に＋２０ｈした文字を返却します。）       */
/*      本プログラムを起動するパソコンＩ／Ｆボードは         */
/*      ６４番機として下さい。                               */
/*      ６３番機側をBSSGタイプで起動させて下さい。           */
/*                                                           */
/*       使用コマンド：登録 =1802H(単一接続ﾓｰﾄﾞ)             */
/*                     WRITE=1821H(ﾀｲﾑｱｳﾄ付き)              */
/*                     READ =1832H(完了ｾﾝｽﾀｲﾌﾟ)             */
/*************************************************************/

#include  <dos.h>
#include  <stdio.h>
#include  <stdlib.h>

#define PC_N  63            /* 通信局番 63 */

static union REGS inregs;
static union REGS outregs;
struct SREGS segregs;
static char far *p1;
static unsigned char dat[48];
unsigned char sbuf[2048];           /* 送信ﾊﾞｯﾌｧ   */
unsigned char jbuf[2048];           /* 受信ﾊﾞｯﾌｧ   */

void main( void )
{
       int    i;
       int    err;
       long   lwork;
       char   cwork[5];
       char   c_dmy1[4];
       char   *dmy, rxno;
       int    datano, rxsize, lped;

/* ---------------------------*/
/* ------ 初期設定処理 ------*/
/* ---------------------------*/

       dat[0] = 0x00;       /* 初期設定ｺﾏﾝﾄﾞ･ｺｰﾄﾞ=1000H  */
       dat[1] = 0x10;
       dat[2] = 0;          /* ｱﾌﾟﾘｹｰｼｮﾝ･ｿﾌﾄﾓｰﾄﾞ=0       */
       dat[3] = 1;          /* 使用ボード枚数=1           */

       printf("初期設定===>");

       if((err = mew_send()) != 0)
       {
              printf("ｴﾗｰ.....ｺｰﾄﾞ= %x¥n", err);
              exit(-1);
       }
       printf("ＯＫ！¥n");

/* ------------------------------------*/
/* ------ 使用ボード登録 及び起動 ------*/
/* ------------------------------------*/

       dat[0] = 0x00;       /* 使用ボード登録及び起動    */
       dat[1] = 0x11;       /* ｺﾏﾝﾄﾞｺｰﾄﾞ=1100H           */
       dat[2] = 0x00;       /* 登録ボードNo.=0           */
       dat[3] = 27;         /* 使用ｾｸﾞﾒﾝﾄｱﾄﾞﾚｽNo.=27     */
       dat[4] = 0;          /* 割り込みNo.=0             */

       printf("使用ボード登録 & 起動===>");

       if((err = mew_send()) != 0)
       {
              printf("ｴﾗｰ.....ｺｰﾄﾞ= %x¥n", err);
```

パソコン機種別変更箇所
（左記はＰＣ９８シリーズの場合）

```
            exit(-1);
        }
        printf("ＯＫ！¥n");

/* ---------------------------------*/
/* ------ 単一接続モード登録 ------*/
/* ---------------------------------*/

        dat[0]  = 0x02;         /* ｼﾘｱﾙ伝送単一接続ﾓｰﾄﾞ  */
        dat[1]  = 0x18;         /* 登録ｺﾏﾝﾄﾞｺｰﾄﾞ=1802H  */
        dat[2]  = 0x00;         /* 登録ﾎﾞｰﾄﾞNo.=0        */
        dat[3]  = 0x00;         /* ﾕｰｻﾞｰID No.=0(固定)   */
        dat[4]  = 0x02;         /* 動作ﾓｰﾄﾞ=単一接続     */
        dat[5]  = 0x0a;         /* ﾀｲﾑｱｳﾄ値=10*0.5(5s)   */
        dat[6]  = 0x01;         /* 再送指定=有り         */
        *((unsigned *)&dat[7])= 0x00;       /* 再送回数=無限          */
        dat[9]  = PC_N;         /* 送信先ﾕﾆｯﾄNo.=64      */
        dat[10]= 0x0F;          /* 送信先CH No.=F        */
        dat[11]= 0x00;          /* 送信先ID No.=0(固定)  */
        dat[12]= 0x00;          /* 伝送経路階層=0(無し)  */

        printf("シリアル伝送単一接続登録===>");

        if((err = mew_send()) != 0)
        {
                printf("ｴﾗｰ.....ｺｰﾄﾞ= %x ¥n", err);
                exit(-1);
        }

        printf("ＯＫ！¥n");

/* ----------------------------------*/
/* ------ シリアルデータＲＥＡＤ ------*/
/* ----------------------------------*/

RXLP:
        lped = 0;
        memset( jbuf, 0, sizeof( jbuf ) );

        do
        {
                p1 = (char far *)&jbuf[0];

                dat[0]  = 0x32;         /* ｼﾘｱﾙﾃﾞｰﾀREAD     */
                dat[1]  = 0x18;         /* ｺﾏﾝﾄﾞｺｰﾄﾞ=1832H  */
                dat[2]  = 0x00;         /* 登録ボードNo.=0  */
                dat[3]  = 0x00;         /* 0固定            */
                *((unsigned *)&dat[4])  = FP_OFF(p1);       /* 受信ﾊﾞｯﾌｧｵﾌｾｯﾄ   */
                *((unsigned *)&dat[6])  = FP_SEG(p1);       /* 受信ﾊﾞｯﾌｧｾｸﾞﾒﾝﾄ */
                *((unsigned *)&dat[8])  = 2048;             /* 受信ﾊﾞｯﾌｧｻｲｽﾞ    */

                printf("シリアルデータ受信センス===>");

                if((err = mew_send()) != 0)
                {
                        if(err == 0x133)
                                printf("受信センス出来ず！！¥n");
                        else
                                printf("ｴﾗｰ.....ｺｰﾄﾞ= %x¥n", err);
                }
                else
                {
                        printf("ＯＫ！¥n");

                        rxsize = dat[10] + dat[11] * 0x100;
                        printf("受信ﾃﾞｰﾀｻｲｽﾞ=%d¥n",rxsize);
                        printf("送信元No.    =%d¥n",dat[23]);
                        printf("受信データ   =%s¥n",jbuf);
                        memset( sbuf, 0, sizeof( sbuf ) );
                        for(i = 0 ; i < rxsize ; i++)
                        {
                                sbuf[i] = jbuf[i] + 0x20;
                        }
                        lped = 1;
                }
        }
        while(lped == 0);
```

```
/* ------------------------------------*/
/* ------ シリアルデータＷＲＩＴ ------*/
/* ------------------------------------*/

        printf("発行データ=%s¥n",sbuf);
        p1 = (char far *)&sbuf[0];

        dat[0] = 0x21;          /* ｼﾘｱﾙﾃﾞｰﾀwRITE     */
        dat[1] = 0x18;          /* ｺﾏﾝﾄﾞｺｰﾄﾞ=1821H   */
        dat[2] = 0x00;          /* 登録ボードno.=0   */
        dat[3] = 0x00;          /* 0固定             */
        *((unsigned *)&dat[4]) = FP_OFF(p1);       /* 送信ﾊﾞｯﾌｧｵﾌｾｯﾄ    */
        *((unsigned *)&dat[6]) = FP_SEG(p1);       /* 送信ﾊﾞｯﾌｧｾｸﾞﾒﾝﾄ   */
        *((unsigned *)&dat[10]) = rxsize;          /* 送信ﾃﾞｰﾀｻｲｽﾞ       */

        printf("シリアルデータ発行===>");

        if((err = mew_send()) != 0)
        {
                printf("ｴﾗｰ.....ｺｰﾄﾞ= %x ¥n", err);
                goto RXLP;
        }

        printf("ＯＫ！¥n");
        printf("¥n");
        goto RXLP;
}


/* ---------------------------*/
/* ------ ﾘﾝｸﾘｿﾌﾄ呼び出し------*/
/* ---------------------------*/
int mew_send(void)
{

        p1 = (char far *)&dat[0];

        inregs.x.dx = FP_OFF(p1);
        segregs.ds = FP_SEG(p1);

        int86x( 0x60, &inregs, &outregs, &segregs);
        return(outregs.x.ax);
}
```

```
/*************************************************************/
/*                  シリアル伝送機能 多重接続                */
/*                          (BSML.C)                         */
/*************************************************************/
/*                  ＰＣ９８シリーズ   */
/*      マスター処理:                                        */
/*      各スレーブ局（１～３番機）に対して、下記処理を実施   */
/*      します。                                             */
/*        ・各スレーブ局受信データの読出しを行います。       */
/*        ・受信データが有れば、同一データを返却します。     */
/*                                                           */
/*       使用コマンド：登録 =1802H(多重ﾏｽﾀｰ接続ﾓｰﾄﾞ)         */
/*                     制御 =1811H(ﾀｲﾑｱｳﾄ付き)              */
/*                     WRITE=1822H(ﾀｲﾑｱｳﾄ付き)              */
/*************************************************************/

#include  <dos.h>
#include  <stdio.h>
```

```
#include  <stdlib.h>

#define PC_N  64                        /* 通信局番 64 */

static union REGS inregs;
static union REGS outregs;
struct SREGS segregs;
static char far *p1;
static unsigned char dat[48];
unsigned char sbuf[2048];               /* 送信ﾊﾞｯﾌｧ   */
unsigned char jbuf[2048];               /* 受信ﾊﾞｯﾌｧ   */

void main( void )
{
        int     i;
        int     err;
        long    lwork;
        char    cwork[5];
        char    c_dmy1[4];
        char    *dmy,txno,txch;
        int     rxsize,lped;

/* ---------------------------*/
/* ------ 初期設定処理 ------*/
/* ---------------------------*/

        dat[0]  = 0x00;         /* 初期設定ｺﾏﾝﾄﾞ･ｺｰﾄﾞ=1000H  */
        dat[1]  = 0x10;
         dat[2]  = 0;           /* ｱﾌﾟﾘｹｰｼｮﾝ･ｿﾌﾄﾓｰﾄﾞ=0       */
         dat[3]  = 1;           /* 使用ボード枚数=1          */

        printf("初期設定===>");

        if((err = mew_send())  != 0)
        {
                printf("ｴﾗｰ.....ｺｰﾄﾞ=  %x¥n",  err);
                exit(-1);
        }
        printf("ＯＫ！¥n");

/* ---------------------------------------*/
/* ------ 使用ボード登録 及び起動 ------*/
/* ---------------------------------------*/

        dat[0]  = 0x00;         /* 使用ボード登録及び起動    */
        dat[1]  = 0x11;         /* ｺﾏﾝﾄﾞｺｰﾄﾞ=1100H          */
        dat[2]  = 0x00;         /* 登録ボードNo.=0           */
        dat[3]  = 27;           /* 使用ｾｸﾞﾒﾝﾄｱﾄﾞﾚｽNo.=27     */
        dat[4]  = 0;            /* 割り込みNo.=0             */

        printf("使用ボード登録 & 起動===>");

        if((err = mew_send())  != 0)
        {
                printf("ｴﾗｰ.....ｺｰﾄﾞ=  %x¥n",  err);
                exit(-1);
        }
        printf("ＯＫ！¥n");

/* ---------------------------------------------*/
/* ------ 多重接続（マスター）モード登録 ------*/
/* ---------------------------------------------*/

        dat[0]  = 0x02;         /* ｼﾘｱﾙ伝送多重接続ﾓｰﾄ       */
        dat[1]  = 0x18;         /* 登録ｺﾏﾝﾄﾞｺｰﾄﾞ=1802H       */
        dat[2]  = 0x00;         /* 登録ﾎﾞｰﾄﾞNo.=0             */
        dat[3]  = 0x00;         /* ﾕｰｻﾞｰID  No.=0(固定)       */
        dat[4]  = 0x00;         /* 動作ﾓｰﾄﾞ=多重ﾏｽﾀｰ          */
        dat[5]  = 0x0a;         /* ﾀｲﾑｱｳﾄ値=10*0.5(5s)        */
        dat[6]  = 0x01;         /* 再送指定=有り             */
        *((unsigned *)&dat[7])= 0x00;       /* 再送回数=無限         */

        printf("シリアル伝送多重接続（マスター）登録===>");

        if((err = mew_send())  != 0)
        {
```

パソコン機種別変更箇所
（左記はＰＣ９８シリーズの場合）

```
                printf("ｴﾗｰ.....ｺｰﾄﾞ= %x ¥n", err);
                exit(-1);
        }

        printf("ＯＫ！¥n");

        txno = 3;                               /* 接続局No.用     */


/* ------------------------------------------------*/
/* ------ 制御コマンド発行（スレーブ局指定） ------*/
/* ------------------------------------------------*/

LP:
        lped = 0;

        do
        {
                printf("¥n");
                if(txno == 3)
                        txno = 1;
                else
                        txno++;

                sprintf(sbuf,"#S1%02d",txno);
                p1 = (char far *)&sbuf[0];
                dat[0] = 0x11;          /* 制御ｺﾏﾝﾄﾞ発行    */
                dat[1] = 0x18;          /* ｺﾏﾝﾄﾞｺｰﾄﾞ=1811H  */
                dat[2] = 0x00;          /* 登録ボードNo.=0  */
                dat[3] = 0x00;          /* 0固定            */
                *((unsigned *)&dat[4]) = FP_OFF(p1);        /* 送信ﾊﾞｯﾌｧｵﾌｾｯﾄ   */
                *((unsigned *)&dat[6]) = FP_SEG(p1);        /* 送信ﾊﾞｯﾌｧｾｸﾞﾒﾝﾄ */
                *((unsigned *)&dat[10]) = 5;                /* 送信ﾃﾞｰﾀｻｲｽﾞ     */

                p1 = (char far *)&jbuf[0];
                *((unsigned *)&dat[38]) = FP_OFF(p1);       /* 受信ﾊﾞｯﾌｧｵﾌｾｯﾄ   */
                *((unsigned *)&dat[40]) = FP_SEG(p1);       /* 受信ﾊﾞｯﾌｧｾｸﾞﾒﾝﾄ */
                *((unsigned *)&dat[42]) = 2048;             /* 受信ﾊﾞｯﾌｧｻｲｽﾞ    */

                printf("スレーブ局（%d番機）接続===>",txno);

                if((err = mew_send()) != 0)
                {
                        printf("ｴﾗｰ.....ｺｰﾄﾞ= %x¥n", err);
                }
                else
                {
                        if(jbuf[0] =='!')
                        {
                                printf("ｴﾗｰﾚｽﾎﾟﾝｽｺｰﾄﾞ=%c%c¥n",jbuf[1],jbuf[2]);
                        }
                        else
                        {
                                printf("ＯＫ！¥n");


/* ------------------------------------------------*/
/* ------ 制御コマンド発行（データ読出指定） ------*/
/* ------------------------------------------------*/

                                memset( jbuf, 0, sizeof( jbuf ) );
                                sprintf(sbuf,"#S2");
                                p1 = (char far *)&sbuf[0];
                                dat[0] = 0x11;          /* 制御ｺﾏﾝﾄﾞ発行    */
                                dat[1] = 0x18;          /* ｺﾏﾝﾄﾞｺｰﾄﾞ=1811H  */
                                dat[2] = 0x00;          /* 登録ボードNo.=0  */
                                dat[3] = 0x00;          /* 0固定            */
                                *((unsigned *)&dat[4]) = FP_OFF(p1);        /* 送信ﾊﾞｯﾌｧｵﾌｾｯﾄ   */
                                *((unsigned *)&dat[6]) = FP_SEG(p1);        /* 送信ﾊﾞｯﾌｧｾｸﾞﾒﾝﾄ */
                                *((unsigned *)&dat[10]) = 3;                /* 送信ﾃﾞｰﾀｻｲｽﾞ     */

                                p1 = (char far *)&jbuf[0];
                                *((unsigned *)&dat[38]) = FP_OFF(p1);       /* 受信ﾊﾞｯﾌｧｵﾌｾｯﾄ   */
                                *((unsigned *)&dat[40]) = FP_SEG(p1);       /* 受信ﾊﾞｯﾌｧｾｸﾞﾒﾝﾄ */
                                *((unsigned *)&dat[42]) = 2048;             /* 受信ﾊﾞｯﾌｧｻｲｽﾞ    */

                                printf("データ読出===>",txno);

                                if((err = mew_send()) != 0)
                                {
                                        printf("ｴﾗｰ.....ｺｰﾄﾞ= %x¥n", err);
                                }
                                else
                                {
                                        if(jbuf[0] =='!')
                                        {
                                                printf("ｴﾗｰﾚｽﾎﾟﾝｽｺｰﾄﾞ=%c%c¥n",jbuf[1],jbuf[2]);
                                        }
```

```
                                        else
                                        {
                                                printf("ＯＫ！¥n");
                                                memset( sbuf, 0, sizeof( sbuf ) );
                                                rxsize = dat[44] + dat[45] * 0x100;
                                                printf("受信ﾃﾞｰﾀｻｲｽﾞ=%d¥n",rxsize);
                                                printf("送信元No.    =%d¥n",dat[23]);
                                                printf("受信データ   =%s¥n",jbuf);
                                                for(i = 0 ; i < rxsize ; i++)
                                                {
                                                        sbuf[i] = jbuf[i];
                                                }
                                                lped = 1;
                                        }
                                }
                        }
                }
        }
        while(lped == 0);


/* ------------------------------------*/
/* ------ シリアルデータＷＲＩＴ ------*/
/* ------------------------------------*/

        printf("発行データ＝%s¥n",sbuf);
        p1 = (char far *)&sbuf[0];

        dat[0] = 0x21;          /* ｼﾘｱﾙﾃﾞｰﾀWRITE       */
        dat[1] = 0x18;          /* ｺﾏﾝﾄﾞｺｰﾄﾞ=1821H     */
        dat[2] = 0x00;          /* 登録ボードNo.=0     */
        dat[3] = 0x00;           /* 0固定              */
        *((unsigned *)&dat[4]) = FP_OFF(p1);        /* 送信ﾊﾞｯﾌｧｵﾌｾｯﾄ    */
        *((unsigned *)&dat[6]) = FP_SEG(p1);        /* 送信ﾊﾞｯﾌｧｾｸﾞﾒﾝﾄ   */
        *((unsigned *)&dat[10]) = rxsize;           /* 送信ﾃﾞｰﾀｻｲｽﾞ       */

        printf("シリアルデータ発行===>");

        if((err = mew_send()) != 0)
                printf("ｴﾗｰ.....ｺｰﾄﾞ= %x ¥n", err);
        else
                printf("ＯＫ！¥n");

        goto LP;
}


/* ----------------------------*/
/* ------ ﾘﾝｸｿﾌﾄ呼び出し------*/
/* ----------------------------*/
int mew_send(void)
{

        p1 = (char far *)&dat[0];

        inregs.x.dx = FP_OFF(p1);
        segregs.ds = FP_SEG(p1);

        int86x( 0x60, &inregs, &outregs, &segregs);
        return(outregs.x.ax);
}
```

ＰＣ／ＡＴ互換機

```
        dat[0] = 0x00;          /* 使用ボード登録及び起動   */
        dat[1] = 0x11;          /* ｺﾏﾝﾄﾞｺｰﾄﾞ=1100H          */
        dat[2] = 0x00;          /* 登録ボードNo.=0          */
        dat[3] = 8;             /* 使用ｾｸﾞﾒﾝﾄｱﾄﾞﾚｽNo.=8     */
        dat[4] = 10;            /* 割り込みNo.=10           */
```

ＦＭＲシリーズ

```
        dat[0] = 0x00;          /* 使用ボード登録及び起動   */
        dat[1] = 0x11;          /* ｺﾏﾝﾄﾞｺｰﾄﾞ=1100H          */
        dat[2] = 0x00;          /* 登録ボードNo.=0          */
        dat[3] = 7;             /* 基板I/OﾎﾟｰﾄｱﾄﾞﾚｽNo.=7    */
        dat[4] = 0x00;          /* ｽﾃｰﾀｽ･ｺﾝﾄﾛｰﾙI/Oﾎﾟｰﾄｱﾄﾞﾚｽ */
        dat[5] = 0x78;          /* =7800H                   */
        dat[6] = 5;             /* 割り込みNo.=5            */
```

3－5 データ転送の機能コマンド

3-5-1 MEWTOCOL-DAT WRITE

データ転送／データ送信（コマンドコード&H153x）

機　能

指定バッファに格納されたMEWTOCOL-DATコマンドをリンクボードに書き込み、送信要求を行います。

解　説

・機能コマンドを実行するには、各種プログラミング言語でI/Oパラメータのセグメントアドレスとオフセットアドレスをds:dxレジスタにセットし、割り込みを実行するユーザープログラムを作成してください。
・コマンド実行後の結果（リターン値）は、axレジスタに格納されます。
　0：正常終了時
　その他：エラーコード（「3－9」参照）

登録送受信要求タイプ、登録送信要求タイプ、登録受信要求タイプの機能コマンドには、同時に使用できる（実行状態にある）コマンド数に制限があります。同時に使用できるコマンド数は、MEWNET-Hリンクソフト（デバイスドライバ）のCONFIG.SYSへの登録時のパラメータスイッチで、0～16まで指定できます。
CONFIG.SYSでの指定がない場合は、同時に実行できるコマンド数は4つまでです。CONFIG.SYSでの指定については、「1-1-2」をお読みください。

■コマンドタイプ

コード	機能
&H1530	送信完了待ち（PC98のみKEY Break付き）
&H1531	送信完了待ち（タイムアウト付き）
&H1532	送信要求のみ
&H1533	送信完了センス
&H1535	登録送信要求（タイムアウト付き）

注　意

・&H1520のKEY Breakは、ESCキーのみです。
・&H1521・&H1525は、タイムアウト付きです。
　タイムアウト値は、コントロールレジスタWRITEコマンドにて変更できます。
・コマンドタイプの動作の違いについては、「3-3-2」をお読みください。

文　例

●MS-DOSファンクションコール（int21）

```
;----- データ転送（送信） ----
SND:
        mov     bx,[FILHDL]
        mov     ah,44h
        mov     al,03h
        mov     dx,offset SND
        int     21h
;------- I/Oﾊﾟﾗﾒｰﾀ ------
SND     dw      1531h
        db      0,0
        dw      offset WBUF,seg WBUF,2
        db      ?,80h,50h,09h,
                1,2,0,2,0,?
                3,9,1,3,7,2,3,1
        dw      9 dup(?)
WBUF    dw      0,10
```

●ソフトウェア割り込み（int60h）

```
;----- データ転送（送信） ----
SND:
        mov     dx,offset SND
        int     60h
;------- I/Oﾊﾟﾗﾒｰﾀ ------
SND     dw      1531h
        db      0,0
        dw      offset WBUF,seg WBUF,2
        db      ?,80h,50h,09h,
                1,2,0,2,0,?
                3,9,1,3,7,2,3,1
        dw      9 dup(?)
WBUF    dw      0,10
```

I/Oパラメータ

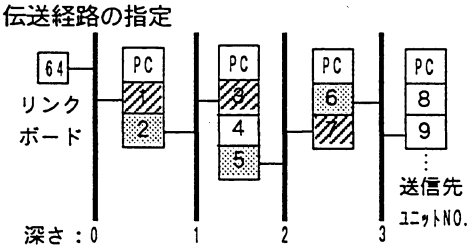
△：設定する［渡す値］
▼：実行後に格納される［得る値］

dsレジスタ ←I/Oパラメータの先頭セグメントアドレス
dxレジスタ ←I/Oパラメータの先頭オフセットアドレス

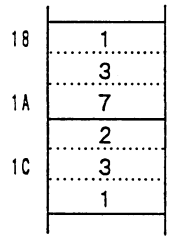
0	△［コマンドコード］&H153x
2	△［登録ボードNo.］0～3（使用するリンクボードの登録ボードNo.）
	△ 設定値：00固定で使用してください。
4	△［送信バッファアドレス（オフセット）］
6	△［送信バッファアドレス（セグメント）］
8	
A	△ ［送信データサイズ］送信バッファにセットしたデータのワード数を指定します。
C	
	△［ヘッダ］
E	△［コマンド／レスポンスコード］
	△［エリアコード／終了コード］
10	△［制御コード］00：階層リンク不使用／01：階層リンク使用
	△［送信先CH］02固定で使用してください。
12	
14	
16	△［深さ］0～3（制御コードで階層リンク使用時にリンクの深さを指定します。）
	△［送信先ユニットNo.］1～64（送信先UNIT No.）
18	△［伝送経路］制御コードで階層リンク使用時に中継局No.→中継リンクの順で指定します。
1A	
1C	
1E	
20	
22	
24	
26	
28	
2A	
2C	
2E	▼［登録送信要求完了フラグ］0：未完了 1：正常終了／その他：エラーコード &H1535のみセットされます。

MEWTOCOL-DAT送信データのエリアコード／終了コード以降をセットします

伝送経路の指定



左図のネットワーク例では
・深さ⇒3
・送信先ユニットNo.⇒9
・中継局No.⇒1、3、7
・中継リンク⇒2、3、1



＊ネットワークの各層のリンクユニットが2台までの場合、深さ0を指定することもできます。

▨中継局No.（UNIT No.を指定）
▩中継リンク（ＰＣから何ユニット目かを指定）

3-5-2 MEWTOCOL-DAT READ

データ転送／データ受信（コマンドコード&H143x）

機　能

リンクボードにMEWTOCOL-DATデータの受信要求を発行し、受信データを指定バッファへ格納します。

解　説

・機能コマンドを実行するには、各種プログラミング言語でI/Oパラメータのセグメントアドレスとオフセットアドレスをds:dxレジスタにセットし、割り込みを実行するユーザープログラムを作成してください。
・コマンド実行後の結果（リターン値）は、axレジスタに格納されます。
　0：正常終了時
　その他：エラーコード（「3－9」参照）

登録送受信要求タイプ、登録送信要求タイプ、登録受信要求タイプの機能コマンドには、同時に使用できる（実行状態にある）コマンド数に制限があります。同時に使用できるコマンド数は、MEWNET-Hリンクソフト（デバイスドライバ）のCONFIG.SYSへの登録時のパラメータスイッチで、0～16まで指定できます。
CONFIG.SYSでの指定がない場合は、同時に実行できるコマンド数は4つまでです。CONFIG.SYSでの指定については、「1-1-2」をお読みください。

■コマンドタイプ

コード	機能
&H1430	受信完了待ち（KEY Break付き）。
&H1431	受信完了待ち（タイムアウト付き）
&H1432	受信完了センス
&H1433	受信バッファクリア（空読み）
&H1434	登録受信要求（タイムアウト無し）
&H1435	登録送信要求（タイムアウト付き）

注　意

・&H1430のKEY Breakは、ESCキーのみです。
・&H1431・&H1435は、タイムアウト付きです。
　タイムアウト値は、コントロールレジスタWRITEコマンドにて変更できます。
・&H1434（登録受信要求タイムアウト無し）の取消には、登録受信タイムアウト無しタイプ解除（&H1460）を実行します。（「3－7」参照）
・コマンドタイプの動作の違いについては、「3-3-3」をお読みください。

文　例

●MS-DOSファンクションコール（int21h）

```
;----- データ転送（受信） ----
RCV:
        mov     bx,[FILHDL]
        mov     ah,44h
        mov     al,03h
        mov     dx,offset RCV
        int     21h
;------- I/Oﾊﾟﾗﾒｰﾀ ------
RCV     dw      1431h
        db      0,0
        dw      offset RBUF,seg RBUF,256,?
        db      ?,?,?,?,
                ?,?,?,?,?,
                ?,?,?,?,?,?,?,?
        dw      9 dup(?)
RBUF    db      255 dub(?)
```

●ソフトウェア割り込み（int60h）

```
;----- データ転送（受信） ----
RCV:
        mov     dx,offset RCV
        int     60h
;------- I/Oﾊﾟﾗﾒｰﾀ ------
RCV     dw      1431h
        db      0,0
        dw      offset RBUF,seg RBUF,256,?
        db      ?,?,?,?,
                ?,?,?,?,?,
                ?,?,?,?,?,?,?,?,?
        dw      9 dup(?)
RBUF    db      255 dub(?)
```

I/Oパラメータ

△：設定する［渡す値］
▼：実行後に格納される［得る値］

dsレジスタ ←I/Oパラメータの先頭セグメントアドレス
dxレジスタ ←I/Oパラメータの先頭オフセットアドレス

0	△［コマンドコード］&H143x
2	△［登録ボードNo.］0～3（使用するリンクボードの登録ボードNo.）
	△ 00固定で使用してください。
4	△［受信バッファアドレス（オフセット］）
6	△［受信バッファアドレス（セグメント）］
8	△［受信バッファサイズ］バイト数を指定します。
A	▼［受信データサイズ］受信バッファに格納したデータのサイズ（ワード数）。
C	
	▼［ヘッダ］
E	▼［コマンド／レスポンスコード］
	▼［エリアコード／終了コード］
10	△［制御コード］送信元指定用
	▼［送信元CH］受信完了時にセットされます。
12	
	△［受信CH］02固定で使用してください。
14	
	▼△［深さ］0～3（制御コードで階層リンク使用時のリンクの深さが指定／格納されます。）
16	▼△［送信元ﾕﾆｯﾄNo.］1～64（送信元指定時にはUNIT No.をセットします。）
	▼△［伝送経路］制御コードで階層リンク使用時に中継局No.→中継リンクの順で指定／格納されます。
18	
1A	
1C	
1E	
20	
22	
24	
26	
28	
2A	
2C	▼［登録No.］登録受信要求（タイムアウト無し）タイプを解除するときに使用する登録No.
2E	▼［登録受信要求完了フラグ］0：未完了／1：正常終了／その他：エラーコード &H1435実行時のみセットされます。

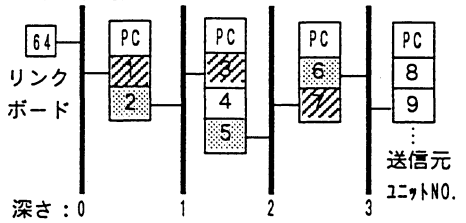
［6］［4］［2］ MEWTOCOL-DAT送信データのエリアコード／終了コード以降がセットされます

■制御コード

	階層リンク	
	使用する	使用しない
送信元指定する	9	8
送信元指定しない	1	0

＊送信元を指定した場合、指定局からのデータのみREADします。

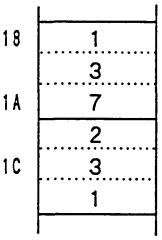
伝送経路の指定



▨中継局No.（UNIT No.を指定）
▩中継リンク（ＰＣから何ユニット目かを指定）

左図のネットワーク例では
・深さ⇒3
・送信元ﾕﾆｯﾄNo.⇒9
・中継局No.⇒1、3、7
・中継リンク⇒2、3、1

＊ネットワークの各層のリンクユニットが2台までの場合、深さ0を指定することもできます。



上記送信元からのデータのみREADする場合
制御コード　=9
深さ　　=3
送信元ﾕﾆｯﾄ　=9
伝送廣　中継局No.　=1、3、7
　　中継リンク　=2、3、1

3-5-3 データ転送のプログラム例

アセンブラプログラム（MS-DOSシステムコールint21）

```
; ********************************************************
; *              MEWTOCOL-DAT                            *
; *                 <AMWDT.ASM>                          *
; ********************************************************
; *                 PC98シリーズ                         *
; *                                                      *
; *     1番機のPCのDT0から10ワード分読み出す             *
; *     階層無し                                         *
; *                                                      *
; *     使用コマンド：WRITE=1531H(ﾀｲﾑｱｳﾄ付き)             *
; *                   READ =1431H(ﾀｲﾑｱｳﾄ付き)           *
; ********************************************************
                                        ;
CODE    SEGMENT
        ASSUME CS:CODE,DS:CODE,ES:CODE,SS:CODE


CR              EQU     0DH             ; CRｺｰﾄﾞ
LF              EQU     0AH             ; LFｺｰﾄﾞ
                                        ;
                                        ;
START:
                MOV     AX,CS
                MOV     DS,AX
                MOV     ES,AX
                MOV     SS,AX
                MOV     AX,OFFSET STPT
                MOV     SP,AX
                                        ;
; ----------------------------------------------
;                 初 期 設 定
; ----------------------------------------------
INI:
                MOV     DX,OFFSET MSG1  ; 初期設定メッセージ表示
                CALL    MESSAGE
                MOV     DX,OFFSET INIT  ; 初期設定用I/Oﾊﾟﾗﾒｰﾀ ｱﾄﾞﾚｽ設定
                CALL    COMMAND         ; ﾘﾝｸﾘｿﾌﾄ呼出
                JNB     INI_OK
                JMP     DOS             ; ｴﾗｰ時 MS-DOSへ戻る
                                        ;
INI_OK:
                MOV     DX,OFFSET MSG1_OK ; 初期設定ＯＫﾒｯｾｰｼﾞ 表示
                CALL    MESSAGE
                                        ;
; ----------------------------------------------
;        ボ ー ド 登 録 ・ 起 動
; ----------------------------------------------
                                        ;
BDST:
                MOV     DX,OFFSET MSG4  ; ﾎﾞｰﾄﾞ登録･起動メッセージ表示
                CALL    MESSAGE
                MOV     DX,OFFSET BOARD ; ﾎﾞｰﾄﾞ登録用I/Oﾊﾟﾗﾒｰﾀ ｱﾄﾞﾚｽ
                CALL    COMMAND         ; ﾘﾝｸﾘｿﾌﾄ呼出
                JNB     BDST_OK
                JMP     DOS             ; ｴﾗｰ時 MS-DOSへ戻る
                                        ;
BDST_OK:
                MOV     DX,OFFSET MSG4_OK ; ﾎﾞｰﾄﾞ登録&起動OKﾒｯｾｰｼﾞ 表示
                CALL    MESSAGE
                                        ;
; ----------------------------------------------
;              CH-2  OPEN
; ----------------------------------------------
                                        ;
CH2OP:
                MOV     DX,OFFSET MSG5  ; ＣＨ－２ ＯＰＥＮメッセージ
                CALL    MESSAGE
```

```
                MOV     DX,OFFSET CHOP          ; CH OPEN I/Oﾊﾟﾗﾒｰﾀｱﾄﾞﾚｽ
                CALL    COMMAND
                JNB     CH2OP_OK
                JMP     DOS                     ; ｴﾗｰ時 MS-DOSへ戻る
                                                ;
CH2OP_OK:
                MOV     DX,OFFSET MSG5_OK       ; CH-2 OPEN OKﾒｯｾｰｼﾞ 表示
                CALL    MESSAGE
                                                ;
;  ---------------------------------------------
;       MEWTOCOL-DAT WRITE
;  ---------------------------------------------
LP:
                MOV     DX,OFFSET TX_MSG        ; 送信コマンド内容表示
                CALL    MESSAGE
                MOV     DX,OFFSET MEWWT         ; 送信用I/Oﾊﾟﾗﾒｰﾀ ｱﾄﾞﾚｽ
                CALL    COMMAND
                JB      LP                      ; ｴﾗｰ時 再送
                                                ;
                MOV     DX,OFFSET TX_OK_MSG     ; 送信ＯＫﾒｯｾｰｼﾞ 表示
                CALL    MESSAGE
                                                ;
;  ---------------------------------------------
;       MEWTOCOL-DAT READ
;  ---------------------------------------------
                                                ;
                MOV     DX,OFFSET RX_MSG        ; ﾚｽﾎﾟﾝｽ読出ﾒｯｾｰｼﾞ 表示
                CALL    MESSAGE
                MOV     DX,OFFSET MEWRD         ; 受信用I/Oﾊﾟﾗﾒｰﾀ ｱﾄﾞﾚｽ
                CALL    COMMAND
                JB      LP                      ; ｴﾗｰ時 送信処理へ
                                                ;
                MOV     DX,OFFSET RX_OK_MSG     ; ﾚｽﾎﾟﾝｽ読出ＯＫﾒｯｾｰｼﾞ 表示
                CALL    MESSAGE
                CALL    RXDATA_DSP              ; 受信内容表示
                JMP     LP
                                                ;
                                                ;
;  ==============================================
;  =          ＭＳ－ＤＯＳへ戻る処理           =
;  ==============================================
                                                ;
DOS:
                MOV     AH,4CH
                INT     21H
                                                ;
                                                ;
;  ==============================================
;  =          メッセージの画面表示処理          =
;  ==============================================
                                                ;
MESSAGE:
                MOV     AH,9                    ; DXｱﾄﾞﾚｽから'$'迄を画面表示
                INT     21H
                RET
                                                ;
                                                ;
;  ==============================================
;  =          リンクソフト呼出処理              =
;  ==============================================
                                                ;
COMMAND:
                INT     60H                     ; ｼｽﾃﾑｺｰﾙ
                CMP     AX,0                    ; AX>0時ｴﾗｰ
                JE      CMD2                    ;
                                                ;
                CALL    BI2ASC                  ; ｴﾗｰｺｰﾄﾞ ASCII変換
                MOV     DX,OFFSET MSG3          ; ｴﾗｰﾒｯｾｰｼﾞ 画面表示
                CALL    MESSAGE
                STC
CMD2:
                RET
                                                ;
                                                ;
;  ==============================================
;  =        エラーコードのＡＳＣＩＩ変換       =
```

```
; ============================================
                                        ;
BI2ASC:
            MOV     BL,AH
            MOV     AH,AL
            MOV     BH,BL
            AND     AL,0FH
            AND     BL,0FH
            MOV     CL,4
            SHR     AH,CL
            SHR     BH,CL
            ADD     BX,3030H
            ADD     AX,3030H
            CMP     AL,3AH
            JB      B0
            ADD     AL,7
B0:
            CMP     AH,3AH
            JB      B1
            ADD     AH,7
B1:
            CMP     BL,3AH
            JB      B2
            ADD     BL,7
B2:
            CMP     BH,3AH
            JB      B3
            ADD     BH,7
B3:
            MOV     [MSG3_1+0],BH
            MOV     [MSG3_1+1],BL
            MOV     [MSG3_1+2],AH
            MOV     [MSG3_1+3],AL
            RET
                                        ;
                                        ;
; ============================================
; =        ＢＩＮ==>ＡＳＣＩＩ変換          =
; ============================================
                                        ;
BIN_ASC:                                ; IN...AL=BINｺｰﾄﾞ
            PUSH    SI                  ; OUT..AX=ASCIIｺｰﾄﾞ
            PUSH    BX
            PUSH    DX
                                        ;
            MOV     BL,AL
            AND     AL,0F0H             ; 上位４ビット
            SHR     AL,1
            SHR     AL,1
            SHR     AL,1
            SHR     AL,1
            LEA     SI,BINTOASC_TB      ; 変換テーブル
            MOV     AH,0
            ADD     SI,AX
            MOV     DH,BYTE PTR CS:[SI]
                                        ;
            MOV     AL,BL
            AND     AL,0FH              ; 下位４ビット
            LEA     SI,BINTOASC_TB      ; 変換テーブル
            MOV     AH,0
            ADD     SI,AX
            MOV     AL,BYTE PTR CS:[SI]
            MOV     AH,DH
            XCHG    AL,AH
                                        ;
            POP     DX
            POP     BX
            POP     SI
            RET
                                        ;
                                        ;
; --------------------------------------------
;       ＨＥＸ→ＡＳＣＩＩ　変換テーブル
; --------------------------------------------
                                        ;
BINTOASC_TB   DB      "0123456789ABCDEF"
```

```
                                        ;
                                        ;
; ========================================
; =              受信データ表示          =
; ========================================
                                        ;
RXDATA_DSP:
            CMP     BYTE PTR CS:[EDCD],0FFH     ; ﾚｽﾎﾟﾝｽOK?
            JE      RX_D_OK             ; YES===>RX_D_OK へ
                                        ; ...NO
            MOV     DX,OFFSET RX_BD_MSG ; ｴﾗｰﾚｽﾎﾟﾝｽ表示
            CALL    MESSAGE
            MOV     AL,CS:[EDCD]
            CALL    BIN_ASC
            CALL    BIN_DSP
            MOV     DX,OFFSET CRLF      ; 改行復帰
            CALL    MESSAGE
            RET
RX_D_OK:
            MOV     DX,OFFSET DSP_RSPCD ; ﾚｽﾎﾟﾝｽｺｰﾄﾞ 表示
            CALL    MESSAGE
            MOV     AL,CS:[RSPCD]
            CALL    BIN_ASC
            CALL    BIN_DSP
            MOV     DX,OFFSET CRLF
            CALL    MESSAGE
                                        ;
            MOV     DX,OFFSET DSP_ENDCD ; 終了ｺｰﾄﾞ 表示
            CALL    MESSAGE
            MOV     AL,CS:[EDCD]
            CALL    BIN_ASC
            CALL    BIN_DSP
            MOV     DX,OFFSET CRLF
            CALL    MESSAGE
                                        ;
            MOV     DX,OFFSET DSP_RXSIZE ; 受信ｻｲｽﾞ 表示
            CALL    MESSAGE
            MOV     BX,CS:[RXSIZE]
            MOV     AL,BH
            CALL    BIN_ASC
            CALL    BIN_DSP
            MOV     AL,BL
            CALL    BIN_ASC
            CALL    BIN_DSP
            MOV     DX,OFFSET CRLF
            CALL    MESSAGE
                                        ;
            XOR     CX,CX
            LEA     DI,JBUF
RX_D_LP:
            MOV     DX,OFFSET DSP_DT    ; "DT" 表示
            CALL    MESSAGE
            MOV     AL,CL
            CALL    BIN_ASC
            CALL    BIN_DSP             ; DTxxのxx表示
            MOV     DX,OFFSET DSP_E
            CALL    MESSAGE             ; "=" 表示
            MOV     BX,DS:[DI]
            MOV     AL,BH
            CALL    BIN_ASC
            CALL    BIN_DSP             ; 上位ﾃﾞｰﾀ表示
            MOV     AL,BL
            CALL    BIN_ASC
            CALL    BIN_DSP             ; 下位ﾃﾞｰﾀ表示
                                        ;
            MOV     DX,OFFSET CRLF      ; 改行復帰
            CALL    MESSAGE
                                        ;
            ADD     DI,2                ; ﾎﾟｲﾝﾀｰ更新
            INC     CL
            CMP     CL,10
            JB      RX_D_LP
                                        ;
            MOV     DX,OFFSET CRLF
            CALL    MESSAGE
            RET
```

```
                                        ;
                                        ;
BIN_DSP:                                ; IN...AX=表示文字ｺｰﾄﾞ
                PUSH    AX
                MOV     DL,AL
                MOV     AH,2
                INT     21H
                POP     AX
                MOV     DL,AH
                MOV     AH,2
                INT     21H
                RET
                                        ;
                                        ;
; ============================================
; =            Ｉ／Ｏパラメータ              =
; ============================================
                                        ; = 初期設定 =
INIT            DW      1000H           ; 初期設定ｺﾏﾝﾄﾞｺｰﾄﾞ
                DB      0               ; ｱﾌﾟﾘｹｰｼｮﾝ･ｿﾌﾄﾓｰﾄﾞ
                DB      1               ; 使用ﾎﾞｰﾄﾞ枚数
                                        ;
                                        ; = ﾎﾞｰﾄﾞ登録･起動 =
BOARD           DW      1100H           ; ﾎﾞｰﾄﾞ登録･起動ｺﾏﾝﾄﾞｺｰﾄﾞ
                DB      0               ; ﾎﾞｰﾄﾞ登録No.
                DB      27              ; 使用ｾｸﾞﾒﾝﾄｱﾄﾞﾚｽNo.27(D0000H)
                DB      0               ; 割り込みNo.
                DB      0
                                        ; = MEWTOCOL-DAT送信(WT) =
MEWWT           DW      1531H           ; 送信ｺﾏﾝﾄﾞｺｰﾄﾞ
                DB      0               ; 登録ﾎﾞｰﾄﾞNO.
                DB      0
                DW      OFFSET SBUF     ; 送信ﾊﾞｯﾌｧOFFSET
                DW      SEG SBUF
                DW      0
                DW      2               ; 送信ﾃﾞｰﾀ･ｻｲｽﾞ
                DB      0               ;
                DB      80H             ; ﾍｯﾀﾞ
                DB      51H             ; ｺﾏﾝﾄﾞｺｰﾄﾞ
                DB      9               ; ｴﾘｱｺｰﾄﾞ
                DB      0               ; 制御ｺｰﾄﾞ
                DB      2               ; 送信先CH
                DB      0               ; 送信先ID
                DB      2               ; 送信元CH
                DB      0               ; 送信元ID
                DB      0
                DB      0               ; Depth
                DB      1                 ; 送信先NO.
                DB      14 DUP(0)
                DW      4 DUP(0)
                DW      0               ;
                                        ;
                                        ;
                                        ; = MEWTOCOL-DAT受信 =
MEWRD           DW      1431H           ; 受信ｺﾏﾝﾄﾞｺｰﾄﾞ
                DB      0               ; 登録ﾎﾞｰﾄﾞNO.
                DB      0
                DW      OFFSET JBUF     ; 受信ﾊﾞｯﾌｧOFFSET
                DW      SEG JBUF
                DW      2048            ; 受信ﾊﾞｯﾌｧｻｲｽﾞ
RXSIZE          DW      0               ; 受信ﾃﾞｰﾀｻｲｽﾞ用
                DB      0               ;
HEAD            DB      0               ; ﾍｯﾀﾞ用
RSPCD           DB      0               ; ﾚｽﾎﾟﾝｽｺｰﾄﾞ用
EDCD            DB      0               ; 終了ｺｰﾄﾞ用
                DB      0               ; 制御ｺｰﾄﾞ
                DB      0               ; 送信元CH
                DB      0               ; 送信元ID
                DB      2               ; 受信CH
                DB      0               ; 受信ID
                DB      0
                DB      0               ; Depth
                DB      0               ; 送信元NO.
                DB      14 DUP(0)       ; 受信経路用
                DW      4 DUP(0)
                DW      0               ;
                                        ;
                                        ; = CH OPEN =
CHOP            DW      1601H           ; ｺﾏﾝﾄﾞｺｰﾄﾞ
```

パソコン機種別変更箇所
（左記はＰＣ９８シリーズの場合）

```
                DB      0                       ; 登録ﾎﾞｰﾄﾞNO.
                DB      2                       ; OPEN CH NO.
                DB      8 DUP(0)                ; CH状態格納用
                                                ;
                                                ;
                                                ;
; ===========================================
; =          メッセージ ／ バッファ領域          =
; ===========================================
                                                ;
MSG1            DB      '初期設定===>$'
MSG1_OK         DB      'ＯＫ！'
                DB      CR,LF,'$'

MSG3            DB      'ｴﾗｰ.....ｺｰﾄﾞ='
MSG3_1          DB      0,0,0,0
                DB      CR,LF,'$'

MSG4            DB      '使用ボード登録 & 起動===>$'
MSG4_OK         DB      'ＯＫ！'
                DB      CR,LF,'$'

MSG5            DB       'ＣＨ２ ＯＰＥＮ===>$'
MSG5_OK         DB      'ＯＫ！',CR,LF,'$'

TX_MSG          DB      'ｺﾏﾝﾄﾞ=80510900000A00',CR,LF
                DB      'ﾊﾟｹｯﾄ送信===>$'
TX_OK_MSG       DB      'ＯＫ！',CR,LF,'$'

RX_MSG          DB      'ﾚｽﾎﾟﾝｽ読出===>$'
RX_OK_MSG       DB      'ＯＫ！',CR,LF,'$'
RX_BD_MSG       DB      'ｴﾗｰﾚｽﾎﾟﾝｽ ｺｰﾄﾞ=$'

DSP_RSPCD       DB      'ﾚｽﾎﾟﾝｽｺｰﾄﾞ          =$'
DSP_ENDCD       DB      '終了ｺｰﾄﾞ             =$'
DSP_RXSIZE      DB      '受信ﾃﾞｰﾀｻｲｽﾞ(Hex)=$'

DSP_DT          DB      'DT$'
DSP_E           DB      '=$'
CRLF            DB      CR,LF,'$'

SBUF            DW      0,0AH                           ; 送信ﾃﾞｰﾀ
JBUF            DW      1024 DUP(0)                     ; 受信ﾊﾞｯﾌｧ

STACK           DW      256 DUP(0)                      ; ｽﾀｯｸ領域
STPT            DB      0


                CODE    ENDS
                END     START
```

パソコン機種別変更箇所

ＰＣ／ＡＴ互換機

```
BOARD           DW      1100H                   ; ﾎﾞｰﾄﾞ登録･起動ｺﾏﾝﾄﾞｺｰﾄﾞ
                DB      0                       ; ﾎﾞｰﾄﾞ登録No.
                DB      8                       ; 使用ｾｸﾞﾒﾝﾄｱﾄﾞﾚｽNo.
                DB      10                      ; 割り込みNo.
                DB      0
```

ＦＭＲシリーズ

```
BOARD           DW      1100H                   ; ﾎﾞｰﾄﾞ登録･起動ｺﾏﾝﾄﾞｺｰﾄﾞ
                DB      0                       ; ﾎﾞｰﾄﾞ登録NO.
                DB      7                       ; 基板I/OﾎﾟｰﾄｱﾄﾞﾚｽNo.
                DW      7800H                   ; ｺﾝﾄﾛｰﾙ･ｽﾃｰﾀｽI/Oﾎﾟｰﾄｱﾄﾞﾚｽ
                DB      5                       ; 割り込みNo.
                DB      0
```

Cプログラム（ソフトウェア割り込みint60）

```
/*************************************************************/
/*                  MEWTOCOL－DAT                            */
/*                      (MWDT.C)                             */
/*************************************************************/
/*                    PC９８リーズ                           */
/*                                                           */
/*     １番機のＰＣのＤＴ０から１０ワード分読み出す          */
/*     階層無し                                              */
/*                                                           */
/*     使用コマンド：WRITE=1531H(ﾀｲﾑｱｳﾄ付き)                  */
/*                   READ =1431H(ﾀｲﾑｱｳﾄ付き)                 */
/*************************************************************/

#include  <dos.h>
#include  <stdio.h>
#include  <stdlib.h>

#define ECODE "D"                   /* DT          */
#define START_N 0                   /* DT 0 から   */
#define END_N  9                    /* DT 9 までﾘｰﾄﾞ*/
#define PC_N  1                     /* 局番 1      */

static union REGS inregs;
static union REGS outregs;
struct SREGS segregs;
static char far *p1;
static unsigned char dat[48];
unsigned int sbuf[1024];            /* 送信ﾊﾞｯﾌｧ   */
unsigned int jbuf[1024];            /* 受信ﾊﾞｯﾌｧ   */

void main( void )
{
       int    i;
       int    err;
       long   lwork;
       char   cwork[5];
       char   c_dmy1[4];
       char   *dmy;
       int    data,rxsize;

/* ---------------------------*/
/* ------ 初期設定処理 ------*/
/* ---------------------------*/

       dat[0] = 0x00;        /* 初期設定ｺﾏﾝﾄﾞ･ｺｰﾄﾞ=1000H  */
       dat[1] = 0x10;
       dat[2] = 0;           /* ｱﾌﾟﾘｹｰｼｮﾝ･ｿﾌﾄﾓｰﾄﾞ=0       */
       dat[3] = 1;           /* 使用ボード枚数=1          */

       printf("初期設定===>");

       if((err = mew_send()) != 0)
       {
              printf("ｴﾗｰ.....ｺｰﾄﾞ=  %x¥n",  err);
              exit(-1);
       }
       printf("ＯＫ！¥n");

/* ------------------------------------*/
/* ------ 使用ボード登録 及び起動 ------*/
/* ------------------------------------*/

       dat[0] = 0x00;        /* 使用ボード登録及び起動    */
       dat[1] = 0x11;        /* ｺﾏﾝﾄﾞｺｰﾄﾞ=1100H           */
       dat[2] = 0x00;        /* 登録ボードNo.=0           */
       dat[3] = 27;          /* 使用ｾｸﾞﾒﾝﾄｱﾄﾞﾚｽNo.=27     */
       dat[4] = 0;           /* 割り込みNo.=0             */

       printf("使用ボード登録 & 起動===>");
```

パソコン機種別変更箇所
（左記はＰＣ９８シリーズの場合）

```
        if((err = mew_send()) != 0)
        {
                printf("ｴﾗｰ.....ｺｰﾄﾞ= %x¥n", err);
                exit(-1);
        }
        printf("ＯＫ！¥n");

/* ------------------------------*/
/* ------ ＣＨ２ ＯＰＥＮ ------*/
/* ------------------------------*/

        dat[0] = 0x01;          /* CH OPEN ｺﾏﾝﾄﾞｺｰﾄﾞ=1601H   */
        dat[1] = 0x16;
        dat[2] = 0x00;          /* 登録ﾎﾞｰﾄﾞNo.=0     */
        dat[3] = 0x02;          /* OPEN CH No. =2     */

        printf("ＣＨ２ ＯＰＥＮ===>");

        if((err = mew_send()) != 0)
        {
                printf("ｴﾗｰ...ｺｰﾄﾞ= %x¥n",err);
                exit(-1);
        }
        printf("ＯＫ！¥n");

/* --------------------------------*/
/* ------ MEWTOCOL-DAT WRITE ------*/
/* --------------------------------*/

TXLP:
        printf("ｺﾏﾝﾄﾞ=805109000000A00¥n");
        p1 = (char far *)&sbuf[0];

        dat[0] = 0x31;          /* MEWTOCOL-DAT WRITE */
        dat[1] = 0x15;          /* ｺﾏﾝﾄﾞｺｰﾄﾞ=1531H     */
         dat[2] = 0x00;         /* 登録ボードno.=0   */
         dat[3] = 0x00;
        *((unsigned *)&dat[4]) = FP_OFF(p1);        /* 送信ﾊﾞｯﾌｧｵﾌｾｯﾄ    */
        *((unsigned *)&dat[6]) = FP_SEG(p1);        /* 送信ﾊﾞｯﾌｧｾｸﾞﾒﾝﾄ   */
        dat[10] = 2;                    /* 送信ﾃﾞｰﾀｻｲ   */
        dat[11] = 0;
        dat[13] = 0x80;                 /* ﾍｯﾀﾞｺｰﾄ             */
        dat[14] = 0x51;                 /* ｺﾏﾝﾄﾞｺｰﾄ            */
        dat[15] = 0x9;                  /* ｴﾘｱｺｰﾄ              */
        dat[16] = 0;                    /* 書き込み制御ｺｰﾄ    */
        dat[17] = 2;                    /* 送信先CH=2          */
        dat[18] = 0;                     /* 送信先ID=0         */
        dat[19] = 2;                    /* 送信元CH=2          */
        dat[20] = 0;                    /* 送信元ID=0          */
        dat[22] = 0;                    /* Depth=0(階層なし)  */
        dat[23] = PC_N;                 /* 送信先No.(PC局番)  */

        sbuf[0] = 0;
        sbuf[1] = 0x0A;

        printf("ﾊﾟｹｯﾄ送信===>");

        if((err = mew_send()) != 0)
        {
                printf("ｴﾗｰ.....ｺｰﾄﾞ= %x ¥n", err);
                goto TXLP;
        }

        printf("ＯＫ！¥n");


/* --------------------------------*/
/* ------ MEWTOCOL-DAT READ ------*/
/* --------------------------------*/

        p1 = (char far *)&jbuf[0];

        dat[0] = 0x31;          /* MEWTOCOL-DAT READ */
        dat[1] = 0x14;          /* ｺﾏﾝﾄﾞｺｰﾄﾞ=1431H    */
        dat[2] = 0x00;          /* 登録ボードno.=0    */
```

```
        dat[3] = 0x00;                  /* 受信ch no. */
        *((unsigned *)&dat[4]) = FP_OFF(p1);        /* 受信ﾊﾞｯﾌｧｵﾌｾｯﾄ  */
        *((unsigned *)&dat[6]) = FP_SEG(p1);        /* 受信ﾊﾞｯﾌｧｾｸﾞﾒﾝﾄ */
        *((unsigned *)&dat[8]) = 2048;              /* 受信ﾊﾞｯﾌｧｻｲｽﾞ    */
        dat[16] = 0;                    /* 受信制御ｺｰﾄﾞ        */
        dat[19] = 2;                    /* 受信CH No.=2        */

        printf("ﾚｽﾎﾟﾝｽ読出===>");

        if((err = mew_send()) != 0)
        {
                printf("ｴﾗｰ.....ｺｰﾄﾞ= %x¥n", err);
        }
        else
        {
                printf("ＯＫ！¥n");

                if( dat[15] != 0xFF )
                {
                        printf("ｴﾗｰﾚｽﾎﾟﾝｽ ｺｰﾄﾞ= %x¥n",dat[15]);
                }
                else
                {
                        printf("ﾚｽﾎﾟﾝｽｺｰﾄﾞ    =%x¥n",dat[14]);
                        printf("終了ｺｰﾄﾞ      =%x¥n",dat[15]);
                        rxsize = dat[10] + dat[11]*0x100;
                        printf("受信ﾃﾞｰﾀｻｲｽﾞ=%d¥n",rxsize);

                        for( i = 0 ; i < END_N - START_N + 1 ; i++)
                        {
                                printf("%s%d=%x¥n",ECODE,i,jbuf[i]);
                        }
                        printf("¥n");
                }
        }
        goto TXLP;
}


/* ---------------------------*/
/* ------ ﾘﾝｸｿﾌﾄ呼び出し------*/
/* ---------------------------*/
int mew_send(void)
{

        p1 = (char far *)&dat[0];

        inregs.x.dx = FP_OFF(p1);
        segregs.ds = FP_SEG(p1);

        int86x( 0x60, &inregs, &outregs, &segregs);
        return(outregs.x.ax);
}
```

パソコン機種別変更箇所

ＰＣ／ＡＴ互換機

```
dat[0] = 0x00;      /* 使用ボード登録及び起動    */
dat[1] = 0x11;      /* ｺﾏﾝﾄﾞｺｰﾄﾞ=1100H           */
dat[2] = 0x00;      /* 登録ボードNo.=0           */
dat[3] = 8;         /* 使用ｾｸﾞﾒﾝﾄｱﾄﾞﾚｽNo.=8      */
dat[4] = 10;        /* 割り込みNo.=10     */
```

ＦＭＲシリーズ

```
dat[0] = 0x00;      /* 使用ボード登録及び起動    */
dat[1] = 0x11;      /* ｺﾏﾝﾄﾞｺｰﾄﾞ=1100H           */
dat[2] = 0x00;      /* 登録ボードNo.=0           */
dat[3] = 7;         /* 基板I/OﾎﾟｰﾄｱﾄﾞﾚｽNo.=7     */
dat[4] = 0x00;      /* ｽﾃｰﾀｽ･ｺﾝﾄﾛｰﾙI/Oﾎﾟｰﾄｱﾄﾞﾚｽ */
dat[5] = 0x78;      /* =7800H                    */
dat[6] = 5;         /* 割り込みNo.=5             */
```

3－6 コンピュータ間通信の機能コマンド

3-6-1 コンピュータ間通信WRITE（文字型）

コンピュータ間通信／書き込み・文字型（コマンドコード&H150x）

機　能

指定された文字型バッファに格納されたデータをリンクボードに書き込み、送信要求を行います。

解　説

・機能コマンドを実行するには、各種プログラミング言語でI/Oパラメータのセグメントアドレスとオフセットアドレスをds:dxレジスタにセットし、割り込みを実行するユーザープログラムを作成してください。
・コマンド実行後の結果（リターン値）は、axレジスタに格納されます。
　0：正常終了時
　その他：エラーコード（「3－9」参照）

登録送受信要求タイプ、登録送信要求タイプ、登録受信要求タイプの機能コマンドには、同時に使用できる（実行状態にある）コマンド数に制限があります。同時に使用できるコマンド数は、MEWNET-Hリンクソフト（デバイスドライバ）のCONFIG.SYSへの登録時のパラメータスイッチで、0～16まで指定できます。CONFIG.SYSでの指定がない場合は、同時に実行できるコマンド数は4つまでです。CONFIG.SYSでの指定については、「1-1-2」をお読みください。

■コマンドタイプ

コード	機能
&H1500	送信完了待ち（PC98のみKey Break付き）
&H1501	送信完了待ち（タイムアウト付き）
&H1502	送信要求のみ
&H1503	送信完了センス
&H1505	登録送信要求（タイムアウト付き）

注　意

・&H1520のKEY Breakは、ESCキーのみです。
・&H1521・&H1525は、タイムアウト付きです。
　タイムアウト値は、コントロールレジスタWRITEコマンドにて変更できます。
・コマンドタイプの動作の違いについては、「3-3-2」をお読みください。

文　例

●MS-DOSファンクションコール（int21h）

```
;----- ｺﾝﾋﾟｭｰﾀ間送信（文字型） ----
CC_WA:
        mov     bx,[FILHDL]
        mov     ah,44h
        mov     al,03h
        mov     dx,offset CC_WA
        int     21h
;------- I/Oﾊﾟﾗﾒｰﾀ ------
CC_WA   dw      1501h
        db      0,0
        dw      offset WBUF,seg WBUF,?,4,?,?
        db      1,1,0,1,0,?,
                3,9,1,3,7,2,3,1
        dw      9 dup(?)
WBUF    db      'ABCD'
```

●ソフトウェア割り込み（int60h）

```
;----- ｺﾝﾋﾟｭｰﾀ間送信（文字型） ----
CC_WA:
        mov     dx,offset CC_WA
        int     40h
;------- I/Oﾊﾟﾗﾒｰﾀ ------
CC_WA   dw      1501h
        db      0,0
        dw      offset WBUF,seg WBUF,?,4,?,?
        db      1,1,0,1,0,?,
                3,9,1,3,7,2,3,1
        dw      9 dup(?)
WBUF    db      'ABCD'
```

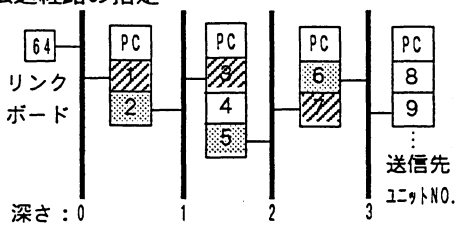
I/Oパラメータ

△：設定する［渡す値］
▼：実行後に格納される［得る値］

dsレジスタ ←I/Oパラメータの先頭セグメントアドレス
dxレジスタ ←I/Oパラメータの先頭オフセットアドレス

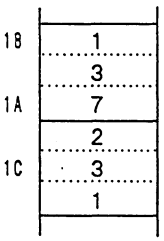
0	△［コマンドコード］&H150x
2	△［登録ボードNo.］0～3（使用するリンクボードの登録ボードNo.）
	△ 設定値：00固定で使用してください。
4	△［送信バッファアドレス（オフセット）］
6	△［送信バッファアドレス（セグメント）］
8	
A	△［送信データサイズ］バイト数を指定
C	
E	
10	△［制御コード］00：階層リンク不使用／01：階層リンク使用
	△［送信先CH］01固定で使用してください。
12	
14	
16	△［深さ］0～3（制御コードで階層リンク使用時にリンクの深さを指定します。）
	△［送信先ユニットNo.］1～64（送信先UNIT No.）
18	△［伝送経路］制御コードで階層リンク使用時に中継局No.→中継リンクの順で指定します。
1A	
1C	
1E	
20	
22	
24	
26	
28	
2A	
2C	
2E	▼［登録送信要求完了フラグ］0：未完了／1：正常終了／その他：エラーコード &H1505実行時のみ設定されます。

伝送経路の指定



左図のネットワーク例では
・深さ⇒3
・送信先ユニットNo.⇒9
・中継局No.⇒1、3、7
・中継リンク⇒2、3、1

＊ネットワークの各層のリンクユニットが2台までの場合、深さ0を指定することもできます。



▨中継局No.（UNIT No.を指定）
▩中継リンク（ＰＣから何ユニット目かを指定）

3-6-2 コンピュータ間通信WRITE（整数型）

コンピュータ間通信／書き込み・整数型（コマンドコード&H151x）

機　能

指定された整数型バッファに格納されたデータをリンクボードに書き込み、送信要求を行います。

解　説

・機能コマンドを実行するには、各種プログラミング言語でI/Oパラメータのセグメントアドレスとオフセットアドレスをds:dxレジスタにセットし、割り込みを実行するユーザープログラムを作成してください。
・コマンド実行後の結果（リターン値）は、axレジスタに格納されます。
　0：正常終了時
　その他：エラーコード（「3－9」参照）

登録送受信要求タイプ、登録送信要求タイプ、登録受信要求タイプの機能コマンドには、同時に使用できる（実行状態にある）コマンド数に制限があります。同時に使用できるコマンド数は、MEWNET-Hリンクソフト（デバイスドライバ）のCONFIG.SYSへの登録時のパラメータスイッチで、0～16まで指定できます。
CONFIG.SYSでの指定がない場合は、同時に実行できるコマンド数は4つまでです。CONFIG.SYSでの指定については、「1-1-2」をお読みください。

■コマンドタイプ

コード	機能
&H1510	送信完了待ち（PC98のみKey Break付き）
&H1511	送信完了待ち（タイムアウト付き）
&H1512	送信要求のみ
&H1513	送信完了センス
&H1515	登録送信要求（タイムアウト付き）

注　意

・&H1520のKEY Breakは、ESCキーのみです。
・&H1521・&H1525は、タイムアウト付きです。
　タイムアウト値は、コントロールレジスタWRITEコマンドにて変更できます。
・コマンドタイプの動作の違いについては、「3-3-2」をお読みください。

文　例

●MS-DOSファンクションコール（int21h）

```
;----- コンピュータ間送信（整数型） ----
CC_WB:
        mov     bx,[FILHDL]
        mov     ah,44h
        mov     al,03h
        mov     dx,offset CC_WB
        int     21h
;------- I/Oﾊﾟﾗﾒｰﾀ ------
CC_WB   dw      1511h
        db      0,0
        dw      offset WBUF,seg WBUF,?,2,?,?,
        db      1,1,0,1,0,?,
                3,9,1,3,7,2,3,1
        dw      9 dup(?)
WBUF    dw      1234h
```

●ソフトウェア割り込み（int60h）

```
;----- コンピュータ間送信（整数型）----
PCRRD:
        mov     dx,offset CC_WB
        int     40h
;------- I/Oﾊﾟﾗﾒｰﾀ ------
CC_WB   dw      1511h
        db      0,0
        dw      offset WBUF,seg WBUF,?,2,?,?,
        db      1,1,0,1,0,?,
                3,9,1,3,7,2,3,1
        dw      9 dup(?)
WBUF    dw      1234h
```

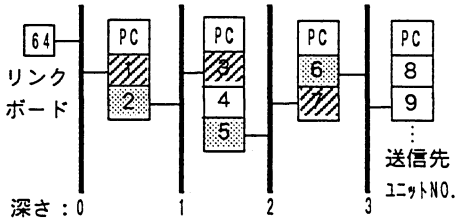
I/Oパラメータ

△：設定する［渡す値］
▼：実行後に格納される［得る値］

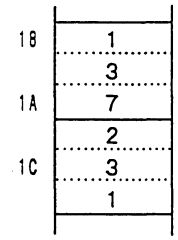
dsレジスタ ←I/Oパラメータの先頭セグメントアドレス
dxレジスタ ←I/Oパラメータの先頭オフセットアドレス

0	△［コマンドコード］&H151x
2	△［登録ボードNo.］0～3（使用するリンクボードの登録ボードNo.）
	△ 設定値：00固定で使用してください。
4	△［送信バッファアドレス（オフセット）］
6	△［送信バッファアドレス（セグメント）］
8	
A	△［送信データサイズ］バイト数を指定
C	
E	
10	△［制御コード］00：階層リンク不使用／01：階層リンク使用
	△［送信先CH］01固定で使用してください。
12	
14	
16	△［深さ］0～3（制御コードで階層リンク使用時にリンクの深さを指定します。）
	△［送信先ﾕﾆｯﾄNo.］1～64（送信先UNIT No.）
18	△［伝送経路］制御コードで階層リンク使用時に中継局No.→中継リンクの順で指定します。
1A	
1C	
1E	
20	
22	
24	
26	
28	
2A	
2C	
2E	▼［登録送信要求完了フラグ］　格納値0：未完了／1：正常終了／その他：エラーコード &H1515実行時のみセットされます。

伝送経路の指定



左図のネットワーク例では
・深さ⇒3
・送信先ﾕﾆｯﾄNo.⇒9
・中継局No.⇒1、3、7
・中継リンク⇒2、3、1
＊ネットワークの各層のリンクユニットが2台までの場合、深さ0を指定することもできます。



▨中継局No.（UNIT No.を指定）
▩中継リンク（ＰＣから何ユニット目かを指定）

3-6-3 コンピュータ間通信READ（文字型）

コンピュータ間通信／読み出し・文字型（コマンドコード&H140x）

機　能

リンクボードにコンピュータ間通信データ（文字型）の受信要求を発行し、受信データを指定バッファへ格納します。

解　説

・機能コマンドを実行するには、各種プログラミング言語でI/Oパラメータのセグメントアドレスとオフセットアドレスをds:dxレジスタにセットし、割り込みを実行するユーザープログラムを作成してください。
・コマンド実行後の結果（リターン値）は、axレジスタに格納されます。
　0：正常終了時
　その他：エラーコード（「3－9」参照）

登録送受信要求タイプ、登録送信要求タイプ、登録受信要求タイプの機能コマンドには、同時に使用できる（実行状態にある）コマンド数に制限があります。同時に使用できるコマンド数は、MEWNET-Hリンクソフト（デバイスドライバ）のCONFIG.SYSへの登録時のパラメータスイッチで、0～16まで指定できます。CONFIG.SYSでの指定がない場合は、同時に実行できるコマンド数は4つまでです。CONFIG.SYSでの指定については、「1-1-2」をお読みください。

■コマンドタイプ

コード	機能
&H1400	受信完了待ち（PC98のみKEY Break付き）
&H1401	受信完了待ち（タイムアウト付き）
&H1402	受信完了センス （センス時受信データ読み出し付き）
&H1403	受信バッファクリア（空読み）
&H1404	登録受信要求（タイムアウト無し）
&H1405	登録送信要求（タイムアウト付き）

注　意

・&H1400のKEY Breakは、 ESC キーのみです。
・&H1401・&H1405は、タイムアウト付きです。
　タイムアウト値は、コントロールレジスタＷＲＩＴＥコマンドにて変更できます。
・&H1404（登録受信要求タイムアウト無し）の取消には、登録受信タイムアウト無しタイプ解除（&H1460）を実行します。（「3－7」参照）
・コマンドタイプの動作の違いについては、「3-3-3」をお読みください。

文　例

●MS-DOSファンクションコール（int21h）

```
;----- ｺﾝﾋﾟｭｰﾀ間受信（文字型） ----
CC_RA:
        mov     bx,[FILHDL]
        mov     ah,44h
        mov     al,03h
        mov     dx,offset CC_RA
        int     21h
;------- I/Oﾊﾟﾗﾒｰﾀ ------
CC_RA   dw      1401h
        db      0,0
        dw      offset RBUF,seg WBUF,256,?,?,?
        db      0,?,?,1,0,?,
                ?,?,?,?,?,?,?,?
        dw      9 dup(?)
RBUF    db      255 dub(?)
```

●ソフトウェア割り込み（int60h）

```
;----- ｺﾝﾋﾟｭｰﾀ間受信（文字型） ----
CC_RA:
        mov     dx,offset CC_RA
        int     60h
;------- I/Oﾊﾟﾗﾒｰﾀ ------
CC_RA   dw      1401h
        db      0,0
        dw      offset RBUF,seg WBUF,256,?,?,?
        db      0,?,?,1,0,?,
                ?,?,?,?,?,?,?,?
        dw      9 dup(?)
RBUF    db      255 dub(?)
```

I/Oパラメータ

△：設定する［渡す値］
▼：実行後に格納される［得る値］

dsレジスタ ←I/Oパラメータの先頭セグメントアドレス
dxレジスタ ←I/Oパラメータの先頭オフセットアドレス

0 △［コマンドコード］&H140x

2 △［登録ボードNo.］0～3（使用するリンクボードの登録ボードNo.）
△ 設定値：00固定で使用してください。

4 △［受信バッファアドレス（オフセット］）

6 △［受信バッファアドレス（セグメント）］

8 △［受信バッファサイズ］バイト数を指定

A ▼［受信データサイズ］受信バッファに格納したデータのサイズ（バイト数）

C

E

■制御コード

	階層リンク	
	使用する	使用しない
送信元指定する	9	8
送信元指定しない	1	0

＊送信元を指定した場合、指定局からのデータのみREADします。

10 △［制御コード］送信元指定用
▼［送信元CH］受信完了時に格納されます。

12 △［受信CH］01固定で指定してください。

14

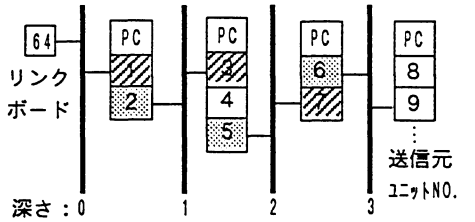
16 ▼△［深さ］0～3（制御コードで階層リンク使用時のリンクの深さが指定／格納されます。）

▼△［送信元ユニットNo.］1～64（送信元指定時にはUNIT No.をセットします。）

18 ▼△［伝送経路］制御コードで階層リンク使用時に中継局No.→中継リンクの順で指定／格納されます。

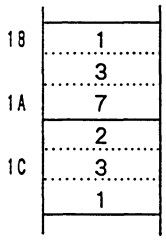
1A

伝送経路の指定



左図のネットワーク例では
・深さ⇒3
・送信元ユニットNo.⇒9
・中継局No.⇒1、3、7
・中継リンク⇒2、3、1

＊ネットワークの各層のリンクユニットが2台までの場合、深さ0を指定することもできます。



▨中継局No.（UNIT No.を指定）
▩中継リンク（ＰＣから何ユニット目かを指定）

上記送信元からのデータのみREADする場合
制御コード =9
深さ =3
送信元ユニット =9
伝送経路 中継局No. =1、3、7
中継リンク =2、3、1

1C

1E

20

22

24

26 ▼［登録No.］登録受信要求（タイムアウト無し）タイプを解除するときに使用する登録No.

28

2A

2C

2E ▼［登録受信要求完了フラグ］0：未完了／1：正常終了／その他：エラーコード
&H1404・&H1405実行時のみセットされます。

3-6-4 コンピュータ間通信READ（整数型）

コンピュータ間通信／読み出し・整数型（コマンドコード&H141x）

機　能

リンクボードにコンピュータ間通信データ（整数型）の受信要求を発行し、受信データを指定バッファへ格納します。

解　説

・機能コマンドを実行するには、各種プログラミング言語でI/Oパラメータのセグメントアドレスとオフセットアドレスをds:dxレジスタにセットし、割り込みを実行するユーザープログラムを作成してください。
・コマンド実行後の結果（リターン値）は、axレジスタに格納されます。
　0：正常終了時
　その他：エラーコード（「3－9」参照）

登録送受信要求タイプ、登録送信要求タイプ、登録受信要求タイプの機能コマンドには、同時に使用できる（実行状態にある）コマンド数に制限があります。同時に使用できるコマンド数は、MEWNET-Hリンクソフト（デバイスドライバ）のCONFIG.SYSへの登録時のパラメータスイッチで、0～16まで指定できます。
CONFIG.SYSでの指定がない場合は、同時に実行できるコマンド数は4つまでです。CONFIG.SYSでの指定については、「1-1-2」をお読みください。

■コマンドタイプ

コード	機能
&H1410	受信完了待ち（PC98のみKEY Break付き）
&H1411	受信完了待ち（タイムアウト付き）
&H1412	受信完了センス （センス時受信データ読み出し付き）
&H1413	受信バッファクリア（空読み）
&H1414	登録受信要求（タイムアウト無し）
&H1415	登録送信要求（タイムアウト付き）

注　意

・&H1410のKEY Breakは、ESCキーのみです。
・&H1411・&H1415は、タイムアウト付きです。
タイムアウト値は、コントロールレジスタWRITEコマンドにて変更できます。
・&H1414（登録受信要求タイムアウト無し）の取消には、登録受信タイムアウト無しタイプ解除（&H1460）を実行します。（「3－7」参照）
・コマンドタイプの動作の違いについては、「3-3-3」をお読みください。

文　例

●MS-DOSファンクションコール（int21h）

```
;----- ｺﾝﾋﾟｭｰﾀ間受信（整数型） ----
CC_RB:
        mov     bx,[FILHDL]
        mov     ah,44h
        mov     al,03h
        mov     dx,offset CC_RB
        int     21h
;------- I/Oﾊﾟﾗﾒｰﾀ ------
CC_RB   dw      1411h
        db      0,0
        dw      offset RBUF,seg RBUF,256,?,?,?
        db      0,?,?,1,?,?,
                ?,?,?,?,?,?,?,?
        dw      9 dup(?)
RBUF    db      255 dub(?)
```

●ソフトウェア割り込み（int60h）

```
;----- ｺﾝﾋﾟｭｰﾀ間受信（整数型） ----
CC_RB:
        mov     dx,offset CC_RB
        int     60h
;------- I/Oﾊﾟﾗﾒｰﾀ ------
CC_RB   dw      1411h
        db      0,0
        dw      offset RBUF,seg RBUF,256,?,?,?
        db      0,?,?,1,?,?,
                ?,?,?,?,?,?,?,?
        dw      9 dup(?)
RBUF    db      255 dub(?)
```

I/Oパラメータ

△：設定する［渡す値］
▼：実行後に格納される［得る値］

dsﾚｼﾞｽﾀ ←I/Oパラメータの先頭セグメントアドレス
dxﾚｼﾞｽﾀ ←I/Oパラメータの先頭オフセットアドレス

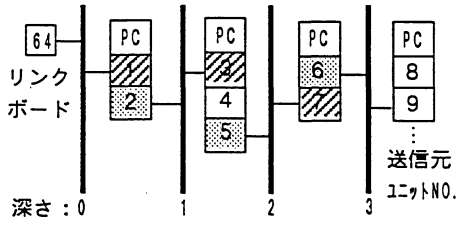
0	△［コマンドコード］&H141x
2	△［登録ボードNo.］0～3（使用するリンクボードの登録ボードNo.）
	△ 設定値：00固定で使用してください。
4	△［受信バッファアドレス（オフセット］）
6	△［受信バッファアドレス（セグメント）］
8	△［受信バッファサイズ］バイト数を指定
A	▼［受信データサイズ］受信バッファに格納したデータのサイズ（バイト数）
C	
E	
10	△［制御コード］送信元指定用
	▼［送信元CH］受信完了時にセットされます。
12	
	△［受信CH］01固定で使用してください。
14	
16	▼△［深さ］0～3（制御コードで階層リンク使用時のリンクの深さが指定／格納されます。）
	▼△［送信元ﾕﾆｯﾄNo.］1～64（送信元指定時にはUNIT No.をセットします。）
18	▼△［伝送経路］制御コードで階層リンク使用時に中継局No.→中継リンクの順で指定／格納されます。
1A	
1C	
1E	
20	
22	
24	
26	
28	
2A	
2C	▼［登録No.］登録受信要求（タイムアウト無し）タイプを解除するときに使用する登録No.
2E	▼［登録受信要求完了フラグ］0：未完了／1：正常終了／その他：エラーコード &H1414・&H1415実行時のみセットされます。

■制御コード

	階層リンク	
	使用する	使用しない
送信元指定する	9	8
送信元指定しない	1	0

＊送信元を指定した場合、指定局からのデータのみREADします。

伝送経路の指定



▨中継局No.（UNIT No.を指定）
▒中継リンク（ＰＣから何ユニット目かを指定）

左図のネットワーク例では
・深さ⇒3
・送信元ﾕﾆｯﾄNo.⇒9
・中継局No.⇒1、3、7
・中継リンク⇒2、3、1

＊ネットワークの各層のリンクユニットが2台までの場合、深さ0を指定することもできます。

18	1
	3
1A	7
	2
1C	3
	1

上記送信元からのデータのみREADする場合
制御コード　=9
深さ　　=3
送信元ﾕﾆｯﾄ　=9
伝送経路　中継局No.　=1、3、7
　　中継リンク　=2、3、1

3-6-5 コンピュータ間通信のプログラム例

アセンブラプログラム（MS-DOSシステムコールint21）

```
; ********************************************************
; *                  ＣＨ文字型 Ｗ／Ｒ                   *
; *                    <ACHCR.ASM>                       *
; ********************************************************
; *                    ＰＣ９８シリーズ                  *
; *     マスター処理:                                    *
; *     ６４番機へ文字データを送信します。               *
; *     ６４番機から返信があれば発行データを変更して     *
; *     いきます。                                       *
; *     ６４番機をCHCR2ﾀｲﾌﾟで起動させて下さい。          *
; *                                                      *
; *     使用コマンド：WRITE=1501H(ﾀｲﾑｱｳﾄ付き)             *
; *                   READ =1402H(完了ｾﾝｽﾀｲﾌﾟ)           *
; ********************************************************
                                        ;
CODE    SEGMENT
        ASSUME CS:CODE,DS:CODE,ES:CODE,SS:CODE


CR              EQU     0DH             ; CRｺｰﾄﾞ
LF              EQU     0AH             ; LFｺｰﾄﾞ
                                        ;
                                        ;
START:
                MOV     AX,CS
                MOV     DS,AX
                MOV     ES,AX
                MOV     SS,AX
                MOV     AX,OFFSET STPT
                MOV     SP,AX
                                        ;
; ----------------------------------------
;               初 期 設 定
; ----------------------------------------
INI:
                MOV     DX,OFFSET MSG1  ; 初期設定メッセージ表示
                CALL    MESSAGE
                MOV     DX,OFFSET INIT  ; 初期設定用I/Oﾊﾟﾗﾒｰﾀ ｱﾄﾞﾚｽ設定
                CALL    COMMAND         ; ﾘﾝｸﾘｿﾌﾄ呼出
                JNB     INI_OK
                JMP     DOS             ; ｴﾗｰ時 MS-DOSへ戻る
                                        ;
INI_OK:
                MOV     DX,OFFSET MSG1_OK ; 初期設定ＯＫﾒｯｾｰｼﾞ表示
                CALL    MESSAGE
                                        ;
; ----------------------------------------
;       ボ ー ド 登 録 ・ 起 動
; ----------------------------------------
                                        ;
                MOV     DX,OFFSET MSG4  ; ﾎﾞｰﾄﾞ登録･起動メッセージ表示
                CALL    MESSAGE
                MOV     DX,OFFSET BOARD ; ﾎﾞｰﾄﾞ登録用I/Oﾊﾟﾗﾒｰﾀ ｱﾄﾞﾚｽ
                CALL    COMMAND         ; ﾘﾝｸﾘｿﾌﾄ呼出
                JNB     BDST_OK
                JMP     DOS             ; ｴﾗｰ時 MS-DOSへ戻る
                                        ;
BDST_OK:
                MOV     DX,OFFSET MSG4_OK ; ﾎﾞｰﾄﾞ登録&起動OKﾒｯｾｰｼﾞ表示
                CALL    MESSAGE
                                        ;
; ----------------------------------------
;             ＣＨ－１  ＯＰＥＮ
; ----------------------------------------
                                        ;
CH1OP:
                MOV     DX,OFFSET MSG5  ; ＣＨ－１ ＯＰＥＮメッセージ
```

```
                CALL    MESSAGE
                MOV     DX,OFFSET CHOP          ; CH OPEN I/Oﾊﾟﾗﾒｰﾀｱﾄﾞﾚｽ
                CALL    COMMAND
                JNB     CH1OP_OK
                JMP     DOS                     ; ｴﾗｰ時 MS-DOSへ戻る
                                                ;
CH1OP_OK:
                MOV     DX,OFFSET MSG5_OK       ; CH-1 OPEN OKﾒｯｾｰｼﾞ 表示
                CALL    MESSAGE
                                                ;
; ---------------------------------------------
;               CH文字型 WRITE
; ---------------------------------------------
                                                ;
                MOV     CS:[DATANO],0
TXLP:
                MOV     BX,CS:[DATANO]          ; BX=送信ﾃﾞｰﾀNo.
                SHL     BX,1
                SHL     BX,1
                LEA     DI,TX_DATA_TB
                MOV     AX,CS:[DI+BX]           ; AX=送信ﾃﾞｰﾀ･ｻｲｽﾞ
                MOV     CS:[TXSIZE],AX
                MOV     AX,CS:[DI+BX+2]         ; AX=送信ﾊﾞｯﾌｧOFFSET
                MOV     CS:[TXBUFF_OFF],AX
TXLP2:
                MOV     DX,OFFSET TX_DATA_MSG ; 発行ﾃﾞｰﾀ表示
                CALL    MESSAGE
                MOV     DX,CS:[TXBUFF_OFF]
                CALL    MESSAGE
                                                ;
                MOV     DX,OFFSET TX_MSG        ; 文字ﾃﾞｰﾀ発行ﾒｯｾｰｼﾞ 表示
                CALL    MESSAGE
                MOV     DX,OFFSET MEWWT         ; WRITEI/Oﾊﾟﾗﾒｰﾀ ｱﾄﾞﾚｽ
                CALL    COMMAND
                JB      TXLP2                   ; ｴﾗｰ時 再送
                                                ;
                MOV     DX,OFFSET TX_OK_MSG     ; ＯＫﾒｯｾｰｼﾞ 表示
                CALL    MESSAGE
                                                ;
; ---------------------------------------------
;               CH文字型 READ
; ---------------------------------------------
                                                ;
                MOV     DX,OFFSET RX_MSG        ; 文字型ﾃﾞｰﾀ受信ｾﾝｽﾒｯｾｰｼﾞ 表示
                CALL    MESSAGE
                MOV     DX,OFFSET MEWRD         ; READ用I/Oﾊﾟﾗﾒｰﾀ ｱﾄﾞﾚｽ
                CALL    COMMAND
                JNB     RD_OK                   ; 受信センス===>RD_OK へ
                                                ; ...NO
                CMP     AX,133H                 ; 受信ｾﾝｽ出来ず?
                JNE     RD_ED                   ; NO ===>RD_ED へ
                                                ; ...YES
                MOV     DX,OFFSET RX_BD_MSG     ; 受信ｾﾝｽ出来ずﾒｯｾｰｼﾞ 表示
                CALL    MESSAGE
                JMP     RD_ED
                                                ;
RD_OK:                                          ; == 受信センス時 ==
                MOV     DX,OFFSET RX_OK_MSG             ; ＯＫﾒｯｾｰｼﾞ 表示
                CALL    MESSAGE
                                                ;
                CALL    RXDATA_DSP              ; 受信内容表示
                MOV     AX,CS:[DATANO]
                INC     AX
                CMP     AX,4
                JB      RD_10
                XOR     AX,AX
RD_10:
                MOV     CS:[DATANO],AX          ; 送信ﾃﾞｰﾀNO.更新
RD_ED:
                MOV     DX,OFFSET CRLF          ; 改行復帰
                CALL    MESSAGE
                JMP     TXLP                    ; 文字型WRITEへ
                                                ;
                                                ;
; ===============================================
; =           ＭＳ－ＤＯＳへ戻る処理             =
```

```
; ============================================
                                        ;
DOS:
            MOV     AH,4CH
            INT     21H
                                        ;
                                        ;
; ============================================
; =          メッセージの画面表示処理          =
; ============================================
                                        ;
MESSAGE:
            MOV     AH,9                ; DXｱﾄﾞﾚｽから'$'迄を画面表示
            INT     21H
            RET
                                        ;
                                        ;
; ============================================
; =          リンクソフト呼出処理          =
; ============================================
                                        ;
COMMAND:
            INT     60H                 ; ｼｽﾃﾑｺｰﾙ
            CMP     AX,0                ; 正常終了?
            JE      CMD_ED              ; YES===>CMD_ED へ
                                        ;
            CMP     AX,133H             ; 受信ｾﾝｽ出来ず?
            JE      CMD_ER              ; YES===>CMD_ER へ
                                        ; ...NO
            PUSH    AX                  ; ｴﾗｰｺｰﾄﾞ待避
            CALL    BI2ASC              ; ｴﾗｰｺｰﾄﾞ ASCII変換
            MOV     DX,OFFSET MSG3      ; ｴﾗｰﾒｯｾｰｼﾞ画面表示
            CALL    MESSAGE
            POP     AX                  ; ｴﾗｰｺｰﾄﾞ復帰
CMD_ER:
            STC
CMD_ED:
            RET
                                        ;
                                        ;
; ============================================
; =        エラーコードのＡＳＣＩＩ変換        =
; ============================================
                                        ;
BI2ASC:
            MOV     BL,AH
            MOV     AH,AL
            MOV     BH,BL
            AND     AL,0FH
            AND     BL,0FH
            MOV     CL,4
            SHR     AH,CL
            SHR     BH,CL
            ADD     BX,3030H
            ADD     AX,3030H
            CMP     AL,3AH
            JB      B0
            ADD     AL,7
B0:
            CMP     AH,3AH
            JB      B1
            ADD     AH,7
B1:
            CMP     BL,3AH
            JB      B2
            ADD     BL,7
B2:
            CMP     BH,3AH
            JB      B3
            ADD     BH,7
B3:
            MOV     [MSG3_1+0],BH
            MOV     [MSG3_1+1],BL
            MOV     [MSG3_1+2],AH
            MOV     [MSG3_1+3],AL
            RET
```

```
                                        ;
                                        ;
; ========================================
; =        ＢＩＮ==>ＡＳＣＩＩ変換        =
; ========================================
                                        ;
BIN_ASC:                                ; IN...AL=BINｺｰﾄﾞ
            PUSH    SI                  ; OUT..AX=ASCIIｺｰﾄﾞ
            PUSH    BX
            PUSH    DX
                                        ;
            MOV     BL,AL
            AND     AL,0F0H             ; 上位４ビット
            SHR     AL,1
            SHR     AL,1
            SHR     AL,1
            SHR     AL,1
            LEA     SI,BINTOASC_TB      ; 変換テーブル
            MOV     AH,0
            ADD     SI,AX
            MOV     DH,BYTE PTR CS:[SI]
                                        ;
            MOV     AL,BL
            AND     AL,0FH              ; 下位４ビット
            LEA     SI,BINTOASC_TB      ; 変換テーブル
            MOV     AH,0
            ADD     SI,AX
            MOV     AL,BYTE PTR CS:[SI]
            MOV     AH,DH
            XCHG    AL,AH
                                        ;
            POP     DX
            POP     BX
            POP     SI
            RET
                                        ;
                                        ;
; ----------------------------------------
;       ＨＥＸ→ＡＳＣＩＩ　変換テーブル
; ----------------------------------------
                                        ;
BINTOASC_TB DB      "0123456789ABCDEF"
                                        ;
                                        ;
; ========================================
; =              受信データ表示            =
; ========================================
                                        ;
RXDATA_DSP:
            MOV     DX,OFFSET DSP_RXSIZE ; 受信ｻｲｽﾞ表示
            CALL    MESSAGE
            MOV     BX,CS:[RXSIZE]
            LEA     DI,JBUF
            MOV     BYTE PTR CS:[DI+BX],CR
            MOV     BYTE PTR CS:[DI+BX+1],LF
            MOV     BYTE PTR CS:[DI+BX+2],'$'
            MOV     AL,BH
            CALL    BIN_ASC
            CALL    BIN_DSP
            MOV     AL,BL
            CALL    BIN_ASC
            CALL    BIN_DSP
            MOV     DX,OFFSET CRLF
            CALL    MESSAGE
                                        ;
            MOV     DX,OFFSET DSP_RXNO  ; 送信元No.表示
            CALL    MESSAGE
            MOV     AL,CS:[RXNO]
            CALL    BIN_ASC
            CALL    BIN_DSP
            MOV     DX,OFFSET CRLF
            CALL    MESSAGE
                                        ;
            MOV     DX,OFFSET DSP_DATA  ; 受信データ表示
            CALL    MESSAGE
            MOV     DX,OFFSET JBUF
```

```
                CALL    MESSAGE
                                                ;
                RET
                                                ;
                                                ;
BIN_DSP:                                        ; IN...AX=表示文字ｺｰﾄﾞ
                PUSH    AX
                MOV     DL,AL
                MOV     AH,2
                INT     21H
                POP     AX
                MOV     DL,AH
                MOV     AH,2
                INT     21H
                RET
                                                ;
                                                ;
; ============================================
; =            Ｉ／Ｏパラメータ                =
; ============================================
                                                ; = 初期設定 =
INIT            DW      1000H                   ; 初期設定ｺﾏﾝﾄﾞｺｰﾄﾞ
                DB      0                       ; ｱﾌﾟﾘｹｰｼｮﾝ･ｿﾌﾄﾓｰﾄﾞ
                DB      1                       ; 使用ﾎﾞｰﾄﾞ枚数
                                                ;
                                                ; = ﾎﾞｰﾄﾞ登録･起動 =
BOARD           DW      1100H                   ; ﾎﾞｰﾄﾞ登録･起動ｺﾏﾝﾄﾞｺｰﾄﾞ
                DB      0                       ; ﾎﾞｰﾄﾞ登録No.
                DB      27                      ; 使用ｾｸﾞﾒﾝﾄｱﾄﾞﾚｽNo.27(D0000H)
                DB      0                       ; 割り込みNo.
                DB      0
                                                ;
                                                ; = ＣＨ文字型ＷＲＩＴＥ =
MEWWT           DW      1501H                   ; ｺﾏﾝﾄﾞｺｰﾄﾞ
                DB      0                       ; 登録ﾎﾞｰﾄﾞNO.
                DB      0
TXBUFF_OFF      DW      0                       ; 送信ﾊﾞｯﾌｧOFFSET
                DW      SEG SBUF0
                DW      0
TXSIZE          DW      0                       ; 送信ﾃﾞｰﾀ･ｻｲｽﾞ
                DB      0                       ;
                DB      0,0,0
                DB      0                       ; 制御ｺｰﾄﾞ
                DB      1                       ; 送信先CH
                DB      0                       ; 送信先ID
                DB      1                       ; 送信元CH
                DB      0                       ; 送信元ID
                DB      0
                DB      0                       ; Depth
                DB      64                      ; 送信先NO.
                DB      14 DUP(0)
                DW      4 DUP(0)
                DW      0                       ;
                                                ;
                                                ;
                                                ; = ＣＨ文字型ＲＥＡＤ =
MEWRD           DW      1402H                   ; ｺﾏﾝﾄﾞｺｰﾄﾞ(完了ｾﾝｽﾀｲﾌﾟ)
                DB      0                       ; 登録ﾎﾞｰﾄﾞNO.
                DB      0
                DW      OFFSET JBUF             ; 受信ﾊﾞｯﾌｧOFFSET
MEWRD2          DW      SEG JBUF
                DW      2048                    ; 受信ﾊﾞｯﾌｧｻｲｽﾞ
RXSIZE          DW      0                       ; 受信ﾃﾞｰﾀｻｲｽﾞ用
                DB      0                       ;
                DB      0,0,0
                DB      0                       ; 制御ｺｰﾄﾞ
                DB      0                       ; 送信元CH
                DB      0                       ; 送信元ID
                DB      1                       ; 受信CH
                DB      0                       ; 受信ID
                DB      0
                DB      0                       ; Depth
RXNO            DB      0                       ; 送信元NO.
                DB      14 DUP(0)               ; 受信経路用
                DW      4 DUP(0)
                DW      0                       ;
```

ﾎﾞｰﾄﾞ登録･起動ｺﾏﾝﾄﾞｺｰﾄﾞ
ﾎﾞｰﾄﾞ登録No.
使用ｾｸﾞﾒﾝﾄｱﾄﾞﾚｽNo.27(D0000H)
割り込みNo.

パソコン機種別変更箇所
（左記はＰＣ９８シリーズの場合）

```
                                        ;
                                        ; = CH OPEN =
CHOP            DW      1601H           ; ｺﾏﾝﾄﾞｺｰﾄﾞ
                DB      0               ; 登録ﾎﾞｰﾄﾞNO.
                DB      1               ; OPEN CH NO.
                DB      8 DUP(0)        ; CH状態格納用
                                        ;
                                        ;
; ============================================
; =          メッセージ ／ バッファ領域          =
; ============================================
                                        ;
MSG1            DB      '初期設定===>$'
MSG1_OK         DB      'ＯＫ！'
                DB      CR,LF,'$'

MSG3            DB      'ｴﾗｰ.....ｺｰﾄﾞ='
MSG3_1          DB      0,0,0,0
                DB      CR,LF,'$'

MSG4            DB      '使用ボード登録 & 起動===>$'
MSG4_OK         DB      'ＯＫ！'
                DB      CR,LF,'$'

MSG5            DB      'ＣＨ１ ＯＰＥＮ===>$'
MSG5_OK         DB      'ＯＫ！',CR,LF,'$'

TX_DATA_MSG     DB      '発行データ＝$'
TX_MSG          DB      '文字データ発行===>$'
TX_OK_MSG       DB      'ＯＫ！',CR,LF,'$'

RX_MSG          DB      '文字データ受信センス===>$'
RX_OK_MSG       DB      'ＯＫ！',CR,LF,'$'
RX_BD_MSG       DB      '受信センス出来ず！！',CR,LF,'$'

DSP_RXSIZE      DB      '受信ﾃﾞｰﾀｻｲｽﾞ(Hex)=$'
DSP_RXNO        DB      '送信元No.      =$'
DSP_DATA        DB      '受信データ     =$'

CRLF            DB      CR,LF,'$'

SBUF0           DB      'AAAAAAAAAA',CR,LF,'$'                  ; 送信ﾃﾞｰﾀ0
SBUF1           DB      'BBBBBBBBBBBBBBB',CR,LF,'$'             ; 送信ﾃﾞｰﾀ1
SBUF2           DB      'CCCCCCCCCCCCCCCCCCCC',CR,LF,'$'        ; 送信ﾃﾞｰﾀ2
SBUF3           DB      'DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD',CR,LF,'$' ; 送信ﾃﾞｰﾀ3

                                        ; 送信ﾃﾞｰﾀ ﾃｰﾌﾞﾙ
TX_DATA_TB      DW      10              ; ﾃﾞｰﾀ0... 送信ｻｲｽﾞ
                DW      OFFSET SBUF0    ;          送信ﾊﾞｯﾌｧ
                DW      15              ; ﾃﾞｰﾀ1... 送信ｻｲｽﾞ
                DW      OFFSET SBUF1    ;          送信ﾊﾞｯﾌｧ
                DW      20              ; ﾃﾞｰﾀ2... 送信ｻｲｽﾞ
                DW      OFFSET SBUF2    ;          送信ﾊﾞｯﾌｧ
                DW      30              ; ﾃﾞｰﾀ3... 送信ｻｲｽﾞ
                DW      OFFSET SBUF3    ;          送信ﾊﾞｯﾌｧ

JBUF            DB      2048 DUP(0)     ; 受信ﾊﾞｯﾌｧ

DATANO          DW      0               ; 送信ﾃﾞｰﾀNo.

STACK           DW      256 DUP(0)      ; ｽﾀｯｸ領域
STPT            DB      0


CODE    ENDS
        END     START
```

```
; ********************************************************
; *                  ＣＨ文字型 Ｗ／Ｒ                   *
; *                    <ACHCR2.ASM>                      *
; ********************************************************
; *                    ＰＣ９８シリーズ                  *
; *      スレーブ処理:                                   *
; *      マスターからのデータを受信したら応答を返します。*
; *      （受信各文字に＋２０ｈした文字を返却します。）  *
; *      本プログラムを起動するパソコンＩ／Ｆボードは    *
; *      ６４番機として下さい。                          *
; *      マスター局側をCHCRﾀｲﾌﾟで起動させて下さい。      *
; *                                                      *
; *      使用コマンド：WRITE=1501H(ﾀｲﾑｱｳﾄ付き)           *
; *                    READ =1402H(完了ｾﾝｽﾀｲﾌﾟ)          *
; ********************************************************
                                        ;
CODE    SEGMENT
        ASSUME CS:CODE,DS:CODE,ES:CODE,SS:CODE


CR              EQU     0DH                     ; CRｺｰﾄﾞ
LF              EQU     0AH                     ; LFｺｰﾄﾞ
                                                ;
                                                ;
START:
                MOV     AX,CS
                MOV     DS,AX
                MOV     ES,AX
                MOV     SS,AX
                MOV     AX,OFFSET STPT
                MOV     SP,AX
                                                ;
;  ---------------------------------------------
;                  初 期 設 定
;  ---------------------------------------------
INI:
                MOV     DX,OFFSET MSG1          ; 初期設定メッセージ表示
                CALL    MESSAGE
                MOV     DX,OFFSET INIT          ; 初期設定用I/Oﾊﾟﾗﾒｰﾀ ｱﾄﾞﾚｽ設定
                CALL    COMMAND                 ; ﾘﾝｸｿﾌﾄ呼出
                JNB     INI_OK
                JMP     DOS                     ; ｴﾗｰ時 MS-DOSへ戻る
                                                ;
INI_OK:
                MOV     DX,OFFSET MSG1_OK       ; 初期設定ＯＫﾒｯｾｰｼﾞ 表示
                CALL    MESSAGE
                                                ;
;  ---------------------------------------------
;         ボ ー ド 登 録 ・ 起 動
;  ---------------------------------------------
                                                ;
                MOV     DX,OFFSET MSG4          ; ﾎﾞｰﾄﾞ登録･起動メッセージ表示
                CALL    MESSAGE
                MOV     DX,OFFSET BOARD         ; ﾎﾞｰﾄﾞ登録用I/Oﾊﾟﾗﾒｰﾀ ｱﾄﾞﾚｽ
                CALL    COMMAND                 ; ﾘﾝｸｿﾌﾄ呼出
                JNB     BDST_OK
                JMP     DOS                     ; ｴﾗｰ時 MS-DOSへ戻る
                                                ;
BDST_OK:
                MOV     DX,OFFSET MSG4_OK               ; ﾎﾞｰﾄﾞ登録&起動OKﾒｯｾｰｼﾞ 表示
                CALL    MESSAGE
                                                ;
;  ---------------------------------------------
;              ＣＨ－１　ＯＰＥＮ
;  ---------------------------------------------
                                                ;
CH1OP:
                MOV     DX,OFFSET MSG5          ; ＣＨ－１ ＯＰＥＮメッセージ
                CALL    MESSAGE
                MOV     DX,OFFSET CHOP          ; CH OPEN I/Oﾊﾟﾗﾒｰﾀｱﾄﾞﾚｽ
                CALL    COMMAND
                JNB     CH1OP_OK
                JMP     DOS                     ; ｴﾗｰ時 MS-DOSへ戻る
                                                ;
```

```
CH1OP_OK:
            MOV     DX,OFFSET MSG5_OK     ; CH-1 OPEN OKﾒｯｾｰｼﾞ 表示
            CALL    MESSAGE
                                          ;
; ------------------------------------------
;           ＣＨ文字型 ＲＥＡＤ
; ------------------------------------------
                                          ;
RXLP:
            MOV     DX,OFFSET RX_MSG      ; 文字型ﾃﾞｰﾀ受信ｾﾝｽﾒｯｾｰｼﾞ 表示
            CALL    MESSAGE
            MOV     DX,OFFSET MEWRD       ; READ用I/Oﾊﾟﾗﾒｰﾀ ｱﾄﾞﾚｽ
            CALL    COMMAND
            JNB     RD_OK                 ; 受信センス===>RD_OK へ
                                          ; ...NO
            CMP     AX,133H               ; 受信ｾﾝｽ出来ず?
            JNE     RXLP                  ; NO ===>RXLP へ
                                          ; ...YES
            MOV     DX,OFFSET RX_BD_MSG   ; 受信ｾﾝｽ出来ずﾒｯｾｰｼﾞ 表示
            CALL    MESSAGE
            JMP     RXLP
                                          ;
RD_OK:                                    ; == 受信センス時 ==
            MOV     DX,OFFSET RX_OK_MSG   ; ＯＫﾒｯｾｰｼﾞ 表示
            CALL    MESSAGE
                                          ;
            CALL    RXDATA_DSP            ; 受信内容表示
                                          ;
            MOV     CX,CS:[RXSIZE]
            LEA     SI,JBUF
            LEA     DI,SBUF
RD_LP:
            MOV     AL,CS:[SI]            ; AL=受信ﾃﾞｰﾀ
            ADD     AL,20H                ; 受信ﾃﾞｰﾀ+20H
            MOV     CS:[DI],AL            ; <==送信ﾊﾞｯﾌｧへ転送
            INC     SI
            INC     DI
            LOOP    RD_LP
                                          ;
            MOV     BYTE PTR CS:[DI],CR
            MOV     BYTE PTR CS:[DI+1],LF
            MOV     BYTE PTR CS:[DI+2],'$'
                                          ;
; ------------------------------------------
;           ＣＨ文字型 ＷＲＩＴＥ
; ------------------------------------------
                                          ;
            MOV     AX,CS:[RXSIZE]
            MOV     CS:[TXSIZE],AX        ; 送信ﾃﾞｰﾀ･ｻｲｽﾞ ｾｯﾄ
            MOV     AL,CS:[RXNO]
            MOV     CS:[TXNO],AL          ; 送信先No.ｾｯﾄ
            MOV     DX,OFFSET TX_DATA_MSG ; 発行ﾃﾞｰﾀ表示
            CALL    MESSAGE
            MOV     DX,OFFSET SBUF
            CALL    MESSAGE
                                          ;
            MOV     DX,OFFSET TX_MSG      ; 文字ﾃﾞｰﾀ発行ﾒｯｾｰｼﾞ 表示
            CALL    MESSAGE
            MOV     DX,OFFSET MEWWT       ; WRITEI/Oﾊﾟﾗﾒｰﾀ ｱﾄﾞﾚｽ
            CALL    COMMAND
            JB      TX_ED                 ; ｴﾗｰ時 再送
                                          ;
            MOV     DX,OFFSET TX_OK_MSG   ; ＯＫﾒｯｾｰｼﾞ 表示
            CALL    MESSAGE
TX_ED:
            MOV     DX,OFFSET CRLF        ; 改行復帰
            CALL    MESSAGE
            JMP     RXLP
                                          ;
; ==========================================
; =          ＭＳ－ＤＯＳへ戻る処理          =
; ==========================================
                                          ;
DOS:
            MOV     AH,4CH
            INT     21H
```

```
                                        ;
                                        ;
; ========================================
; =          メッセージの画面表示処理          =
; ========================================
                                        ;
MESSAGE:
            MOV     AH, 9               ; DXｱﾄﾞﾚｽから'$'迄を画面表示
            INT     21H
            RET
                                        ;
                                        ;
; ========================================
; =          リンクソフト呼出処理            =
; ========================================
                                        ;
COMMAND:
            INT     60H                 ; ｼｽﾃﾑｺｰﾙ
            CMP     AX, 0               ; 正常終了?
            JE      CMD_ED              ; YES===>CMD_ED へ
                                        ;
            CMP     AX, 133H            ; 受信ｾﾝｽ出来ず?
            JE      CMD_ER              ; YES===>CMD_ER へ
                                        ; ...NO
            PUSH    AX                  ; ｴﾗｰｺｰﾄﾞ待避
            CALL    BI2ASC              ; ｴﾗｰｺｰﾄﾞ ASCII変換
            MOV     DX, OFFSET MSG3     ; ｴﾗｰﾒｯｾｰｼﾞ画面表示
            CALL    MESSAGE
            POP     AX                  ; ｴﾗｰｺｰﾄﾞ復帰
CMD_ER:
            STC
CMD_ED:
            RET
                                        ;
                                        ;
; ========================================
; =       エラーコードのＡＳＣＩＩ変換       =
; ========================================
                                        ;
BI2ASC:
            MOV     BL, AH
            MOV     AH, AL
            MOV     BH, BL
            AND     AL, 0FH
            AND     BL, 0FH
            MOV     CL, 4
            SHR     AH, CL
            SHR     BH, CL
            ADD     BX, 3030H
            ADD     AX, 3030H
            CMP     AL, 3AH
            JB      B0
            ADD     AL, 7
B0:
            CMP     AH, 3AH
            JB      B1
            ADD     AH, 7
B1:
            CMP     BL, 3AH
            JB      B2
            ADD     BL, 7
B2:
            CMP     BH, 3AH
            JB      B3
            ADD     BH, 7
B3:
            MOV     [MSG3_1+0], BH
            MOV     [MSG3_1+1], BL
            MOV     [MSG3_1+2], AH
            MOV     [MSG3_1+3], AL
            RET
                                        ;
                                        ;
; ========================================
; =        ＢＩＮ==>ＡＳＣＩＩ変換          =
```

```
; ============================================
                                        ;
BIN_ASC:                                ; IN...AL=BINｺｰﾄﾞ
                PUSH    SI              ; OUT..AX=ASCIIｺｰﾄﾞ
                PUSH    BX
                PUSH    DX
                                        ;
                MOV     BL,AL
                AND     AL,0F0H         ; 上位４ビット
                SHR     AL,1
                SHR     AL,1
                SHR     AL,1
                SHR     AL,1
                LEA     SI,BINTOASC_TB  ; 変換テーブル
                MOV     AH,0
                ADD     SI,AX
                MOV     DH,BYTE PTR CS:[SI]
                                        ;
                MOV     AL,BL
                AND     AL,0FH          ; 下位４ビット
                LEA     SI,BINTOASC_TB  ; 変換テーブル
                MOV     AH,0
                ADD     SI,AX
                MOV     AL,BYTE PTR CS:[SI]
                MOV     AH,DH
                XCHG    AL,AH
                                        ;
                POP     DX
                POP     BX
                POP     SI
                RET
                                        ;
                                        ;
; ------------------------------------------
;       ＨＥＸ→ＡＳＣＩＩ　変換テーブル
; ------------------------------------------
                                        ;
BINTOASC_TB     DB      "0123456789ABCDEF"
                                        ;
                                        ;
; ============================================
; =               受信データ表示               =
; ============================================
                                        ;
RXDATA_DSP:
                MOV     DX,OFFSET DSP_RXSIZE ; 受信ｻｲｽﾞ表示
                CALL    MESSAGE
                MOV     BX,CS:[RXSIZE]
                LEA     DI,JBUF
                MOV     BYTE PTR CS:[DI+BX],CR
                MOV     BYTE PTR CS:[DI+BX+1],LF
                MOV     BYTE PTR CS:[DI+BX+2],'$'
                MOV     AL,BH
                CALL    BIN_ASC
                CALL    BIN_DSP
                MOV     AL,BL
                CALL    BIN_ASC
                CALL    BIN_DSP
                MOV     DX,OFFSET CRLF
                CALL    MESSAGE
                                        ;
                MOV     DX,OFFSET DSP_RXNO ; 送信元No.表示
                CALL    MESSAGE
                MOV     AL,CS:[RXNO]
                CALL    BIN_ASC
                CALL    BIN_DSP
                MOV     DX,OFFSET CRLF
                CALL    MESSAGE
                                        ;
                MOV     DX,OFFSET DSP_DATA ; 受信データ表示
                CALL    MESSAGE
                MOV     DX,OFFSET JBUF
                CALL    MESSAGE
                                        ;
                RET
```

```
                                        ;
                                        ;
BIN_DSP:                                ; IN...AX=表示文字ｺｰﾄﾞ
                PUSH    AX
                MOV     DL,AL
                MOV     AH,2
                INT     21H
                POP     AX
                MOV     DL,AH
                MOV     AH,2
                INT     21H
                RET
                                        ;
                                        ;
; ==========================================
; =            Ｉ／Ｏパラメータ              =
; ==========================================
                                        ; = 初期設定 =
INIT            DW      1000H           ; 初期設定ｺﾏﾝﾄﾞｺｰﾄﾞ
                DB      0               ; ｱﾌﾟﾘｹｰｼｮﾝ･ｿﾌﾄﾓｰﾄﾞ
                DB      1               ; 使用ﾎﾞｰﾄﾞ枚数
                                        ;
                                        ; = ﾎﾞｰﾄﾞ登録･起動 =
BOARD           DW      1100H           ; ﾎﾞｰﾄﾞ登録･起動ｺﾏﾝﾄﾞｺｰﾄﾞ
                DB      0               ; ﾎﾞｰﾄﾞ登録No.
                DB      27              ; 使用ｾｸﾞﾒﾝﾄｱﾄﾞﾚｽNo.27(D0000H)
                DB      0               ; 割り込みNo.
                DB      0
                                        ;
                                        ; = ＣＨ文字型ＷＲＩＴＥ =
MEWWT           DW      1501H           ; ｺﾏﾝﾄﾞｺｰﾄﾞ
                DB      0               ; 登録ﾎﾞｰﾄﾞNO.
                DB      0
                DW      OFFSET SBUF     ; 送信ﾊﾞｯﾌｧOFFSET
                DW      SEG SBUF        ; 送信ﾊﾞｯﾌｧSEGMENT
                DW      0
TXSIZE          DW      0               ; 送信ﾃﾞｰﾀ･ｻｲｽﾞ
                DB      0               ;
                DB      0,0,0
                DB      0               ; 制御ｺｰﾄﾞ
                DB      1               ; 送信先CH
                DB      0               ; 送信先ID
                DB      1               ; 送信元CH
                DB      0               ; 送信元ID
                DB      0
                DB      0               ; Depth
TXNO            DB      0                 ; 送信先NO.
                DB      14 DUP(0)
                DW      4 DUP(0)
                DW      0               ;
                                        ;
                                        ;
                                        ; = ＣＨ文字型ＲＥＡＤ =
MEWRD           DW      1402H           ; ｺﾏﾝﾄﾞｺｰﾄﾞ(完了ｾﾝｽﾀｲﾌﾟ)
                DB      0               ; 登録ﾎﾞｰﾄﾞNO.
                DB      0
                DW      OFFSET JBUF     ; 受信ﾊﾞｯﾌｧOFFSET
MEWRD2          DW      SEG JBUF
                DW      2048            ; 受信ﾊﾞｯﾌｧｻｲｽﾞ
RXSIZE          DW      0               ; 受信ﾃﾞｰﾀｻｲｽﾞ用
                DB      0               ;
                DB      0,0,0
                DB      0               ; 制御ｺｰﾄﾞ
                DB      0               ; 送信元CH
                DB      0               ; 送信元ID
                DB      1               ; 受信CH
                DB      0               ; 受信ID
                DB      0
                DB      0               ; Depth
RXNO            DB      0               ; 送信元NO.
                DB      14 DUP(0)       ; 受信経路用
                DW      4 DUP(0)
                DW      0               ;
                                        ;
                                        ; = ＣＨ ＯＰＥＮ =
CHOP            DW      1601H           ; ｺﾏﾝﾄﾞｺｰﾄﾞ
```

パソコン機種別変更箇所
（左記はＰＣ９８シリーズの場合）

```
                DB      0                       ; 登録ﾎﾞｰﾄﾞNO.
                DB      1                       ; OPEN CH NO.
                DB      8 DUP(0)                ; CH状態格納用
                                                ;
                                                ;
; ============================================
; =          メッセージ ／ バッファ領域          =
; ============================================
                                                ;
MSG1            DB      '初期設定===>$'
MSG1_OK         DB      'ＯＫ！'
                DB      CR,LF,'$'

MSG3            DB      'ｴﾗｰ.....ｺｰﾄﾞ='
MSG3_1          DB      0,0,0,0
                DB      CR,LF,'$'

MSG4            DB      '使用ボード登録 & 起動===>$'
MSG4_OK         DB      'ＯＫ！'
                DB      CR,LF,'$'

MSG5            DB       'ＣＨ１ ＯＰＥＮ===>$'
MSG5_OK         DB      'ＯＫ！',CR,LF,'$'

TX_DATA_MSG     DB      '発行データ＝$'
TX_MSG          DB      '文字データ発行===>$'
TX_OK_MSG       DB      'ＯＫ！',CR,LF,'$'

RX_MSG          DB      '文字データ受信センス===>$'
RX_OK_MSG       DB      'ＯＫ！',CR,LF,'$'
RX_BD_MSG       DB      '受信センス出来ず！！',CR,LF,'$'

DSP_RXSIZE      DB      '受信ﾃﾞｰﾀｻｲｽﾞ(Hex)=$'
DSP_RXNO        DB       '送信元No.          =$'
DSP_DATA        DB      '受信データ         =$'

CRLF            DB      CR,LF,'$'

SBUF            DB      2048 DUP(0)                     ; 送信ﾊﾞｯﾌｧ
JBUF            DB      2048 DUP(0)                     ; 受信ﾊﾞｯﾌｧ

STACK           DW      256 DUP(0)                      ; ｽﾀｯｸ領域
STPT            DB      0


                CODE    ENDS
                END     START
```

```
PAGE    60,132
; ************************************************************
; *                    ＣＨ整数型 Ｗ／Ｒ                     *
; *                       <ACHIT.ASM>                        *
; ************************************************************
; *                      ＰＣ９８シリー                      *
; *      マスター処理:                                       *
; *      ６４番機へ数値データを送信します。                  *
; *      ６４番機から返信があれば発行データを変更して        *
; *      いきます。                                          *
; *      ６４番機をCHIT2ﾀｲﾌﾟで起動させて下さい。            *
; *                                                          *
; *      使用コマンド：WRITE=1511H(ﾀｲﾑｱｳﾄ付き)                *
; *                    READ =1412H(完了ｾﾝｽﾀｲﾌﾟ)               *
; ************************************************************
                                                ;
CODE    SEGMENT
        ASSUME CS:CODE,DS:CODE,ES:CODE,SS:CODE
```

```
CR              EQU     0DH                     ; CRｺｰﾄﾞ
LF              EQU     0AH                     ; LFｺｰﾄﾞ
                                                ;
                                                ;
START:
                MOV     AX,CS
                MOV     DS,AX
                MOV     ES,AX
                MOV     SS,AX
                MOV     AX,OFFSET STPT
                MOV     SP,AX
                                                ;
;  ---------------------------------------------
;                  初 期 設 定
;  ---------------------------------------------
INI:
                MOV     DX,OFFSET MSG1          ; 初期設定メッセージ表示
                CALL    MESSAGE
                MOV     DX,OFFSET INIT          ; 初期設定用I/Oﾊﾟﾗﾒｰﾀ ｱﾄﾞﾚｽ設定
                CALL    COMMAND                 ; ﾘﾝｸｿﾌﾄ呼出
                JNB     INI_OK
                JMP     DOS                     ; ｴﾗｰ時 MS-DOSへ戻る
                                                ;
INI_OK:
                MOV     DX,OFFSET MSG1_OK       ; 初期設定ＯＫﾒｯｾｰｼﾞ 表示
                CALL    MESSAGE
                                                ;
;  ---------------------------------------------
;        ボ ー ド 登 録 ・ 起 動
;  ---------------------------------------------
                                                ;
                MOV     DX,OFFSET MSG4          ; ﾎﾞｰﾄﾞ登録･起動メッセージ表示
                CALL    MESSAGE
                MOV     DX,OFFSET BOARD         ; ﾎﾞｰﾄﾞ登録用I/Oﾊﾟﾗﾒｰﾀ ｱﾄﾞﾚｽ
                CALL    COMMAND                 ; ﾘﾝｸｿﾌﾄ呼出
                JNB     BDST_OK
                JMP     DOS                     ; ｴﾗｰ時 MS-DOSへ戻る
                                                ;
BDST_OK:
                MOV     DX,OFFSET MSG4_OK       ; ﾎﾞｰﾄﾞ登録&起動OKﾒｯｾｰｼﾞ 表示
                CALL    MESSAGE
                                                ;
;  ---------------------------------------------
;              CH－1  OPEN
;  ---------------------------------------------
                                                ;
CH1OP:
                MOV     DX,OFFSET MSG5          ; ＣＨ－１ ＯＰＥＮメッセージ
                CALL    MESSAGE
                MOV     DX,OFFSET CHOP          ; CH OPEN I/Oﾊﾟﾗﾒｰﾀｱﾄﾞﾚｽ
                CALL    COMMAND
                JNB     CH1OP_OK
                JMP     DOS                     ; ｴﾗｰ時 MS-DOSへ戻る
                                                ;
CH1OP_OK:
                MOV     DX,OFFSET MSG5_OK       ; CH-1 OPEN OKﾒｯｾｰｼﾞ 表示
                CALL    MESSAGE
                                                ;
;  ---------------------------------------------
;              CH整数型 WRITE
;  ---------------------------------------------
                                                ;
                MOV     CS:[DATANO],0
TXLP:
                MOV     BX,CS:[DATANO]          ; BX=送信ﾃﾞｰﾀNo.
                SHL     BX,1
                SHL     BX,1
                LEA     DI,TX_DATA_TB
                MOV     AX,CS:[DI+BX]           ; AX=送信ﾃﾞｰﾀ先頭ｺｰﾄﾞ
                MOV     CX,CS:[DI+BX+2]         ; CX=送信ﾃﾞｰﾀｻｲｽﾞ
                MOV     CS:[TXSIZE],CX
                LEA     DI,SBUF
TX_DTST_LP:
                MOV     CS:[DI],AL              ; 送信ﾃﾞｰﾀｾｯﾄ
```

```
                INC     DI
                INC     AL
                LOOP    TX_DTST_LP
TXLP2:
                CALL    TXDATA_DSP              ; 発行ﾃﾞｰﾀ表示
                                                ;
                MOV     DX,OFFSET TX_MSG        ; 数値ﾃﾞｰﾀ発行ﾒｯｾｰｼﾞ 表示
                CALL    MESSAGE
                MOV     DX,OFFSET MEWWT         ; WRITEI/Oﾊﾟﾗﾒｰﾀ ｱﾄﾞﾚｽ
                CALL    COMMAND
                JB      TXLP2                   ; ｴﾗｰ時 再送
                                                ;
                MOV     DX,OFFSET TX_OK_MSG     ; ＯＫﾒｯｾｰｼﾞ 表示
                CALL    MESSAGE
                                                ;
; ---------------------------------------------
;               ＣＨ整数型 ＲＥＡＤ
; ---------------------------------------------
                                                ;
                MOV     DX,OFFSET RX_MSG        ; 数値型ﾃﾞｰﾀ受信ｾﾝｽﾒｯｾｰｼﾞ 表示
                CALL    MESSAGE
                MOV     DX,OFFSET MEWRD         ; READ用I/Oﾊﾟﾗﾒｰﾀ ｱﾄﾞﾚｽ
                CALL    COMMAND
                JNB     RD_OK                   ; 受信センス===>RD_OK へ
                                                ; ...NO
                CMP     AX,133H                 ; 受信ｾﾝｽ出来ず?
                JNE     RD_ED                   ; NO ===>RD_ED へ
                                                ; ...YES
                MOV     DX,OFFSET RX_BD_MSG     ; 受信ｾﾝｽ出来ずﾒｯｾｰｼﾞ 表示
                CALL    MESSAGE
                JMP     RD_ED
                                                ;
RD_OK:                                          ; == 受信センス時 ==
                MOV     DX,OFFSET RX_OK_MSG     ; ＯＫﾒｯｾｰｼﾞ 表示
                CALL    MESSAGE
                                                ;
                CALL    RXDATA_DSP              ; 受信内容表示
                MOV     AX,CS:[DATANO]
                INC     AX
                CMP     AX,4
                JB      RD_10
                XOR     AX,AX
RD_10:
                MOV     CS:[DATANO],AX          ; 送信ﾃﾞｰﾀNO.更新
RD_ED:
                MOV     DX,OFFSET CRLF          ; 改行復帰
                CALL    MESSAGE
                JMP     TXLP                    ; 整数型WRITEへ
                                                ;
                                                ;
; ============================================
; =          ＭＳ－ＤＯＳへ戻る処理          =
; ============================================
                                                ;
DOS:
                MOV     AH,4CH
                INT     21H
                                                ;
                                                ;
; ============================================
; =          メッセージの画面表示処理          =
; ============================================
                                                ;
MESSAGE:
                MOV     AH,9                    ; DXｱﾄﾞﾚｽから'$'迄を画面表示
                INT     21H
                RET
                                                ;
                                                ;
; ============================================
; =          リンクソフト呼出処理              =
; ============================================
                                                ;
COMMAND:
                INT     60H                     ; ｼｽﾃﾑｺｰﾙ
                CMP     AX,0                    ; 正常終了?
```

```
                JE      CMD_ED                  ; YES===>CMD_ED へ
                                                ;
                CMP     AX,133H                 ; 受信ｾﾝｽ出来ず?
                JE      CMD_ER                  ; YES===>CMD_ER へ
                                                ; ...NO
                PUSH    AX                      ; ｴﾗｰｺｰﾄﾞ待避
                CALL    BI2ASC                  ; ｴﾗｰｺｰﾄﾞ ASCII変換
                MOV     DX,OFFSET MSG3          ; ｴﾗｰﾒｯｾｰｼﾞ画面表示
                CALL    MESSAGE
                POP     AX                      ; ｴﾗｰｺｰﾄﾞ復帰
CMD_ER:
                STC
CMD_ED:
                RET
                                                ;
                                                ;
; ==========================================
; =       エラーコードのＡＳＣＩＩ変換       =
; ==========================================
                                                ;
BI2ASC:
                MOV     BL,AH
                MOV     AH,AL
                MOV     BH,BL
                AND     AL,0FH
                AND     BL,0FH
                MOV     CL,4
                SHR     AH,CL
                SHR     BH,CL
                ADD     BX,3030H
                ADD     AX,3030H
                CMP     AL,3AH
                JB      B0
                ADD     AL,7
B0:
                CMP     AH,3AH
                JB      B1
                ADD     AH,7
B1:
                CMP     BL,3AH
                JB      B2
                ADD     BL,7
B2:
                CMP     BH,3AH
                JB      B3
                ADD     BH,7
B3:
                MOV     [MSG3_1+0],BH
                MOV     [MSG3_1+1],BL
                MOV     [MSG3_1+2],AH
                MOV     [MSG3_1+3],AL
                RET
                                                ;
                                                ;
; ==========================================
; =        ＢＩＮ==>ＡＳＣＩＩ変換         =
; ==========================================
                                                ;
BIN_ASC:                                        ; IN...AL=BINｺｰﾄﾞ
                PUSH    SI                      ; OUT..AX=ASCIIｺｰﾄﾞ
                PUSH    BX
                PUSH    DX
                                                ;
                MOV     BL,AL
                AND     AL,0F0H                 ; 上位４ビット
                SHR     AL,1
                SHR     AL,1
                SHR     AL,1
                SHR     AL,1
                LEA     SI,BINTOASC_TB          ; 変換テーブル
                MOV     AH,0
                ADD     SI,AX
                MOV     DH,BYTE PTR CS:[SI]
                                                ;
                MOV     AL,BL
                AND     AL,0FH                  ; 下位４ビット
```

```
                LEA     SI,BINTOASC_TB          ; 変換テーブル
                MOV     AH,0
                ADD     SI,AX
                MOV     AL,BYTE PTR CS:[SI]
                MOV     AH,DH
                XCHG    AL,AH
                                                ;
                POP     DX
                POP     BX
                POP     SI
                RET
                                                ;
                                                ;
;  ---------------------------------------------
;         ＨＥＸ→ＡＳＣＩＩ　変換テーブル
;  ---------------------------------------------
                                                ;
BINTOASC_TB     DB      "0123456789ABCDEF"
                                                ;
                                                ;
;  =============================================
;  =                送信データ表示                =
;  =============================================
                                                ;
TXDATA_DSP:
                MOV     DX,OFFSET TX_DATA_MSG   ; 発行ﾃﾞｰﾀ表示
                CALL    MESSAGE
                LEA     DI,SBUF                 ; DI=送信ﾊﾞｯﾌｧ
                MOV     CX,CS:[TXSIZE]          ; CX=送信ｻｲｽﾞ
                CALL    DATA_DSP
                RET
                                                ;
                                                ;
;  =============================================
;  =                受信データ表示                =
;  =============================================
                                                ;
RXDATA_DSP:
                MOV     DX,OFFSET DSP_RXSIZE    ; 受信ｻｲｽﾞ表示
                CALL    MESSAGE
                MOV     BX,CS:[RXSIZE]
                MOV     AL,BH
                CALL    BIN_ASC
                CALL    BIN_DSP
                MOV     AL,BL
                CALL    BIN_ASC
                CALL    BIN_DSP
                MOV     DX,OFFSET CRLF
                CALL    MESSAGE
                                                ;
                MOV     DX,OFFSET DSP_RXNO      ; 送信元No.表示
                CALL    MESSAGE
                MOV     AL,CS:[RXNO]
                CALL    BIN_ASC
                CALL    BIN_DSP
                MOV     DX,OFFSET CRLF
                CALL    MESSAGE
                                                ;
                MOV     DX,OFFSET DSP_DATA      ; 受信データ表示
                CALL    MESSAGE
                LEA     DI,JBUF                 ; DI=受信ﾊﾞｯﾌｧ
                MOV     CX,CS:[RXSIZE]          ; CX=受信ｻｲｽﾞ
                CALL    DATA_DSP
                RET
                                                ;
;  ---------------------------------------------
;               数値データ表示ＳＵＢ
;  ---------------------------------------------
                                                ; IN...CX=表示数
DATA_DSP:                                       ;       DI=表示ﾃﾞｰﾀ位置
                MOV     AL,DS:[DI]
                CALL    BIN_ASC
                CALL    BIN_DSP                 ; ﾃﾞｰﾀ表示
                INC     DI                      ; ﾎﾟｲﾝﾀｰ更新
                LOOP    DATA_DSP
                                                ;
```

```
                MOV     DX,OFFSET CRLF
                CALL    MESSAGE
                RET
                                        ;
                                        ;
BIN_DSP:                                ; IN...AX=表示文字ｺｰﾄﾞ
                PUSH    AX
                MOV     DL,AL
                MOV     AH,2
                INT     21H
                POP     AX
                MOV     DL,AH
                MOV     AH,2
                INT     21H
                RET
                                        ;
                                        ;
; ==========================================
; =          I／Oパラメータ                =
; ==========================================
                                        ; = 初期設定 =
INIT            DW      1000H           ; 初期設定ｺﾏﾝﾄﾞｺｰﾄﾞ
                DB      0               ; ｱﾌﾟﾘｹｰｼｮﾝ･ｿﾌﾄﾓｰﾄﾞ
                DB      1               ; 使用ﾎﾞｰﾄﾞ枚数
                                        ;
                                        ; = ﾎﾞｰﾄﾞ登録･起動 =
BOARD           DW      1100H           ; ﾎﾞｰﾄﾞ登録･起動ｺﾏﾝﾄﾞｺｰﾄﾞ
                DB      0               ; ﾎﾞｰﾄﾞ登録No.
                DB      27              ; 使用ｾｸﾞﾒﾝﾄｱﾄﾞﾚｽNo.27(D0000H)
                DB      0               ; 割り込みNo.
                DB      0
                                        ;
                                        ; = ＣＨ整数型ＷＲＩＴＥ =
MEWWT           DW      1511H           ; ｺﾏﾝﾄﾞｺｰﾄﾞ
                DB      0               ; 登録ﾎﾞｰﾄﾞNO.
                DB      0
                DW      OFFSET SBUF     ; 送信ﾊﾞｯﾌｧOFFSET
                DW      SEG SBUF
                DW      0
TXSIZE          DW      0               ; 送信ﾃﾞｰﾀ･ｻｲｽﾞ
                DB      0               ;
                DB      0,0,0
                DB      0               ; 制御ｺｰﾄﾞ
                DB      1               ; 送信先CH
                DB      0               ; 送信先ID
                DB      1               ; 送信元CH
                DB      0               ; 送信元ID
                DB      0
                DB      0               ; Depth
                DB      64              ; 送信先NO.
                DB      14 DUP(0)
                DW      4 DUP(0)
                DW      0               ;
                                        ;
                                        ;
                                        ; = ＣＨ整数型ＲＥＡＤ =
MEWRD           DW      1412H           ; ｺﾏﾝﾄﾞｺｰﾄﾞ(完了ｾﾝｽﾀｲﾌﾟ)
                DB      0               ; 登録ﾎﾞｰﾄﾞNO.
                DB      0
                DW      OFFSET JBUF     ; 受信ﾊﾞｯﾌｧOFFSET
MEWRD2          DW      SEG JBUF
                DW      2048            ; 受信ﾊﾞｯﾌｧｻｲｽﾞ
RXSIZE          DW      0               ; 受信ﾃﾞｰﾀｻｲｽﾞ用
                DB      0               ;
                DB      0,0,0
                DB      0               ; 制御ｺｰﾄﾞ
                DB      0               ; 送信元CH
                DB      0               ; 送信元ID
                DB      1               ; 受信CH
                DB      0               ; 受信ID
                DB      0
                DB      0               ; Depth
RXNO            DB      0               ; 送信元NO.
                DB      14 DUP(0)       ; 受信経路用
                DW      4 DUP(0)
                DW      0               ;
```

パソコン機種別変更箇所
（左記はＰＣ９８シリーズの場合）

```
                                        ;
                                        ; = CH OPEN =
CHOP        DW      1601H               ; ｺﾏﾝﾄﾞｺｰﾄﾞ
            DB      0                   ; 登録ﾎﾞｰﾄﾞNO.
            DB      1                   ; OPEN CH NO.
            DB      8 DUP(0)            ; CH状態格納用
                                        ;
                                        ;
; ==========================================
; =        メッセージ ／ バッファ領域        =
; ==========================================
                                        ;
MSG1        DB      '初期設定===>$'
MSG1_OK     DB      'ＯＫ！'
            DB      CR,LF,'$'

MSG3        DB      'ｴﾗｰ.....ｺｰﾄﾞ='
MSG3_1      DB      0,0,0,0
            DB      CR,LF,'$'

MSG4        DB      '使用ボード登録 & 起動===>$'
MSG4_OK     DB      'ＯＫ！'
            DB      CR,LF,'$'

MSG5        DB      'ＣＨ１ ＯＰＥＮ===>$'
MSG5_OK     DB      'ＯＫ！',CR,LF,'$'

TX_DATA_MSG DB      '発行データ＝$'
TX_MSG      DB      '数値データ発行===>$'
TX_OK_MSG   DB      'ＯＫ！',CR,LF,'$'

RX_MSG      DB      '数値データ受信センス===>$'
RX_OK_MSG   DB      'ＯＫ！',CR,LF,'$'
RX_BD_MSG   DB      '受信センス出来ず！！',CR,LF,'$'

DSP_RXSIZE  DB      '受信ﾃﾞｰﾀｻｲｽﾞ(Hex)=$'
DSP_RXNO    DB      '送信元No.          =$'
DSP_DATA    DB      '受信データ         =$'

CRLF        DB      CR,LF,'$'

                                        ; 送信ﾃﾞｰﾀ ﾃｰﾌﾞﾙ
TX_DATA_TB  DW      0                   ; ﾃﾞｰﾀ0... 送信ﾃﾞｰﾀ先頭ｺｰﾄﾞ
            DW      10                  ;          送信ｻｲｽﾞ
            DW      10                  ; ﾃﾞｰﾀ1... 送信ﾃﾞｰﾀ先頭ｺｰﾄﾞ
            DW      20                  ;          送信ｻｲｽﾞ
            DW      30                  ; ﾃﾞｰﾀ2... 送信ﾃﾞｰﾀ先頭ｺｰﾄﾞ
            DW      30                  ;          送信ｻｲｽﾞ
            DW      70                  ; ﾃﾞｰﾀ3... 送信ﾃﾞｰﾀ先頭ｺｰﾄﾞ
            DW      30                  ;          送信ｻｲｽﾞ

SBUF        DB      2048 DUP(0)         ; 送信ﾊﾞｯﾌｧ
JBUF        DB      2048 DUP(0)         ; 受信ﾊﾞｯﾌｧ

DATANO      DW      0                   ; 送信ﾃﾞｰﾀNo.

STACK       DW      256 DUP(0)          ; ｽﾀｯｸ領域
STPT        DB      0


            CODE    ENDS
            END     START
```

```
; ********************************************************
; *                  ＣＨ整数型 Ｗ／Ｒ                   *
; *                    <ACHIT2.ASM>                      *
; ********************************************************
; *                      ＰＣ９８シリーズ                *
; *     スレーブ処理:                                    *
; *     マスターからのデータを受信したら応答を返します。 *
; *     （受信データをそのまま返却します。）             *
; *     本プログラムを起動するパソコンＩ／Ｆボードは     *
; *     ６４番機として下さい。                           *
; *     マスター局側をCHITﾀｲﾌﾟで起動させて下さい。       *
; *                                                      *
; *     使用コマンド：WRITE=1511H(ﾀｲﾑｱｳﾄ付き)             *
; *                   READ =1412H(完了ｾﾝｽﾀｲﾌﾟ)           *
; ********************************************************
                                        ;
CODE    SEGMENT
        ASSUME CS:CODE,DS:CODE,ES:CODE,SS:CODE


CR              EQU     0DH             ; CRｺｰﾄﾞ
LF              EQU     0AH             ; LFｺｰﾄﾞ
                                        ;
                                        ;
START:
                MOV     AX,CS
                MOV     DS,AX
                MOV     ES,AX
                MOV     SS,AX
                MOV     AX,OFFSET STPT
                MOV     SP,AX
                                        ;
; ---------------------------------------------
;                 初 期 設 定
; ---------------------------------------------
INI:
                MOV     DX,OFFSET MSG1  ; 初期設定メッセージ表示
                CALL    MESSAGE
                MOV     DX,OFFSET INIT  ; 初期設定用I/Oﾊﾟﾗﾒｰﾀ ｱﾄﾞﾚｽ設定
                CALL    COMMAND         ; ﾘﾝｸｿﾌﾄ呼出
                JNB     INI_OK
                JMP     DOS             ; ｴﾗｰ時 MS-DOSへ戻る
                                        ;
INI_OK:
                MOV     DX,OFFSET MSG1_OK ; 初期設定ＯＫﾒｯｾｰｼﾞ 表示
                CALL    MESSAGE
                                        ;
; ---------------------------------------------
;         ボ ー ド 登 録 ・ 起 動
; ---------------------------------------------
                                        ;
                MOV     DX,OFFSET MSG4  ; ﾎﾞｰﾄﾞ登録･起動メッセージ表示
                CALL    MESSAGE
                MOV     DX,OFFSET BOARD ; ﾎﾞｰﾄﾞ登録用I/Oﾊﾟﾗﾒｰﾀ ｱﾄﾞﾚｽ
                CALL    COMMAND         ; ﾘﾝｸｿﾌﾄ呼出
                JNB     BDST_OK
                JMP     DOS             ; ｴﾗｰ時 MS-DOSへ戻る
                                        ;
BDST_OK:
                MOV     DX,OFFSET MSG4_OK ; ﾎﾞｰﾄﾞ登録&起動OKﾒｯｾｰｼﾞ 表示
                CALL    MESSAGE
                                        ;
; ---------------------------------------------
;               ＣＨ－１  ＯＰＥＮ
; ---------------------------------------------
                                        ;
CH1OP:
                MOV     DX,OFFSET MSG5  ; ＣＨ－１ ＯＰＥＮメッセージ
                CALL    MESSAGE
                MOV     DX,OFFSET CHOP  ; CH OPEN I/Oﾊﾟﾗﾒｰﾀｱﾄﾞﾚｽ
                CALL    COMMAND
                JNB     CH1OP_OK
                JMP     DOS             ; ｴﾗｰ時 MS-DOSへ戻る
                                        ;
```

```
CH1OP_OK:
            MOV     DX,OFFSET MSG5_OK     ; CH-1 OPEN OKﾒｯｾｰｼﾞ 表示
            CALL    MESSAGE
                                          ;
; ------------------------------------------
;             ＣＨ整数型 ＲＥＡＤ
; ------------------------------------------
                                          ;
RXLP:
            MOV     DX,OFFSET RX_MSG      ; 数値型ﾃﾞｰﾀ受信ｾﾝｽﾒｯｾｰｼﾞ 表示
            CALL    MESSAGE
            MOV     DX,OFFSET MEWRD       ; READ用I/Oﾊﾟﾗﾒｰﾀ ｱﾄﾞﾚｽ
            CALL    COMMAND
            JNB     RD_OK                 ; 受信センス===>RD_OK へ
                                          ; ...NO
            CMP     AX,133H               ; 受信ｾﾝｽ出来ず?
            JNE     RXLP                  ; NO ===>RXLP へ
                                          ; ...YES
            MOV     DX,OFFSET RX_BD_MSG   ; 受信ｾﾝｽ出来ずﾒｯｾｰｼﾞ 表示
            CALL    MESSAGE
            JMP     RXLP
                                          ;
RD_OK:                                    ; == 受信センス時 ==
            MOV     DX,OFFSET RX_OK_MSG   ; ＯＫﾒｯｾｰｼﾞ 表示
            CALL    MESSAGE
                                          ;
            CALL    RXDATA_DSP            ; 受信内容表示
                                          ;
            MOV     CX,CS:[RXSIZE]
            LEA     SI,JBUF
            LEA     DI,SBUF
            CLD
            REP     MOVSB                 ; 受信ﾃﾞｰﾀ===>送信ﾊﾞｯﾌｧ へ転送
                                          ;
                                          ;
; ------------------------------------------
;             ＣＨ数値型 ＷＲＩＴＥ
; ------------------------------------------
                                          ;
            MOV     AX,CS:[RXSIZE]
            MOV     CS:[TXSIZE],AX        ; 送信ﾃﾞｰﾀ･ｻｲｽﾞ ｾｯﾄ
            MOV     AL,CS:[RXNO]
            MOV     CS:[TXNO],AL          ; 送信先No.ｾｯﾄ
            CALL    TXDATA_DSP            ; 発行ﾃﾞｰﾀ表示
                                          ;
            MOV     DX,OFFSET TX_MSG      ; 数値ﾃﾞｰﾀ発行ﾒｯｾｰｼﾞ 表示
            CALL    MESSAGE
            MOV     DX,OFFSET MEWWT       ; WRITEI/Oﾊﾟﾗﾒｰﾀ ｱﾄﾞﾚｽ
            CALL    COMMAND
            JB      TX_ED                 ; ｴﾗｰ時 再送
                                          ;
            MOV     DX,OFFSET TX_OK_MSG   ; ＯＫﾒｯｾｰｼﾞ 表示
            CALL    MESSAGE
TX_ED:
            MOV     DX,OFFSET CRLF        ; 改行復帰
            CALL    MESSAGE
            JMP     RXLP
                                          ;
; ==========================================
; =          ＭＳ－ＤＯＳへ戻る処理          =
; ==========================================
                                          ;
DOS:
            MOV     AH,4CH
            INT     21H
                                          ;
                                          ;
; ==========================================
; =        メッセージの画面表示処理          =
; ==========================================
                                          ;
MESSAGE:
            MOV     AH,9                  ; DXｱﾄﾞﾚｽから'$'迄を画面表示
            INT     21H
            RET
                                          ;
```

```
                                        ;
; ========================================
; =          リンクソフト呼出処理          =
; ========================================
                                        ;
COMMAND:
            INT     60H                 ; ｼｽﾃﾑｺｰﾙ
            CMP     AX,0                ; 正常終了?
            JE      CMD_ED              ; YES===>CMD_ED へ
                                        ;
            CMP     AX,133H             ; 受信ｾﾝｽ出来ず?
            JE      CMD_ER              ; YES===>CMD_ER へ
                                        ; ...NO
            PUSH    AX                  ; ｴﾗｰｺｰﾄﾞ待避
            CALL    BI2ASC              ; ｴﾗｰｺｰﾄﾞ ASCII変換
            MOV     DX,OFFSET MSG3      ; ｴﾗｰﾒｯｾｰｼﾞ画面表示
            CALL    MESSAGE
            POP     AX                  ; ｴﾗｰｺｰﾄﾞ復帰
CMD_ER:
            STC
CMD_ED:
            RET
                                        ;
                                        ;
; ========================================
; =      エラーコードのＡＳＣＩＩ変換      =
; ========================================
                                        ;
BI2ASC:
            MOV     BL,AH
            MOV     AH,AL
            MOV     BH,BL
            AND     AL,0FH
            AND     BL,0FH
            MOV     CL,4
            SHR     AH,CL
            SHR     BH,CL
            ADD     BX,3030H
            ADD     AX,3030H
            CMP     AL,3AH
            JB      B0
            ADD     AL,7
B0:
            CMP     AH,3AH
            JB      B1
            ADD     AH,7
B1:
            CMP     BL,3AH
            JB      B2
            ADD     BL,7
B2:
            CMP     BH,3AH
            JB      B3
            ADD     BH,7
B3:
            MOV     [MSG3_1+0],BH
            MOV     [MSG3_1+1],BL
            MOV     [MSG3_1+2],AH
            MOV     [MSG3_1+3],AL
            RET
                                        ;
                                        ;
; ========================================
; =       ＢＩＮ==>ＡＳＣＩＩ変換       =
; ========================================
                                        ;
BIN_ASC:                                ; IN...AL=BINｺｰﾄﾞ
            PUSH    SI                  ; OUT..AX=ASCIIｺｰﾄﾞ
            PUSH    BX
            PUSH    DX
                                        ;
            MOV     BL,AL
            AND     AL,0F0H             ; 上位４ビット
            SHR     AL,1
            SHR     AL,1
            SHR     AL,1
```

```
                SHR     AL,1
                LEA     SI,BINTOASC_TB          ; 変換テーブル
                MOV     AH,0
                ADD     SI,AX
                MOV     DH,BYTE PTR CS:[SI]
                                                ;
                MOV     AL,BL
                AND     AL,0FH                  ; 下位４ビット
                LEA     SI,BINTOASC_TB          ; 変換テーブル
                MOV     AH,0
                ADD     SI,AX
                MOV     AL,BYTE PTR CS:[SI]
                MOV     AH,DH
                XCHG    AL,AH
                                                ;
                POP     DX
                POP     BX
                POP     SI
                RET
                                                ;
                                                ;
;  ---------------------------------------------
;         ＨＥＸ→ＡＳＣＩＩ　変換テーブル
;  ---------------------------------------------
                                                ;
BINTOASC_TB     DB      "0123456789ABCDEF"
                                                ;
                                                ;
;  =============================================
;  =                 送信データ表示                =
;  =============================================
                                                ;
TXDATA_DSP:
                MOV     DX,OFFSET TX_DATA_MSG ; 発行ﾃﾞｰﾀ表示
                CALL    MESSAGE
                LEA     DI,SBUF                 ; DI=送信ﾊﾞｯﾌｧ
                MOV     CX,CS:[TXSIZE]          ; CX=送信ｻｲｽﾞ
                CALL    DATA_DSP
                RET
                                                ;
                                                ;
;  =============================================
;  =                 受信データ表示                =
;  =============================================
                                                ;
RXDATA_DSP:
                MOV     DX,OFFSET DSP_RXSIZE  ; 受信ｻｲｽﾞ表示
                CALL    MESSAGE
                MOV     BX,CS:[RXSIZE]
                MOV     AL,BH
                CALL    BIN_ASC
                CALL    BIN_DSP
                MOV     AL,BL
                CALL    BIN_ASC
                CALL    BIN_DSP
                MOV     DX,OFFSET CRLF
                CALL    MESSAGE
                                                ;
                MOV     DX,OFFSET DSP_RXNO    ; 送信元No.表示
                CALL    MESSAGE
                MOV     AL,CS:[RXNO]
                CALL    BIN_ASC
                CALL    BIN_DSP
                MOV     DX,OFFSET CRLF
                CALL    MESSAGE
                                                ;
                MOV     DX,OFFSET DSP_DATA    ; 受信データ表示
                CALL    MESSAGE
                LEA     DI,JBUF                 ; DI=受信ﾊﾞｯﾌｧ
                MOV     CX,CS:[RXSIZE]          ; CX=受信ｻｲｽﾞ
                CALL    DATA_DSP
                RET
                                                ;
;  ---------------------------------------------
;               数値データ表示ＳＵＢ
;  ---------------------------------------------
```

```
                                        ; IN...CX=表示数
DATA_DSP:                               ;       DI=表示ﾃﾞｰﾀ位置
            MOV     AL,DS:[DI]
            CALL    BIN_ASC
            CALL    BIN_DSP             ; ﾃﾞｰﾀ表示
            INC     DI                  ; ﾎﾟｲﾝﾀｰ更新
            LOOP    DATA_DSP
                                        ;
            MOV     DX,OFFSET CRLF
            CALL    MESSAGE
            RET
                                        ;
                                        ;
BIN_DSP:                                ; IN...AX=表示文字ｺｰﾄﾞ
            PUSH    AX
            MOV     DL,AL
            MOV     AH,2
            INT     21H
            POP     AX
            MOV     DL,AH
            MOV     AH,2
            INT     21H
            RET
                                        ;
                                        ;
; ==========================================
; =          Ｉ／Ｏパラメータ              =
; ==========================================
                                        ; = 初期設定 =
INIT        DW      1000H               ; 初期設定ｺﾏﾝﾄﾞｺｰﾄﾞ
            DB      0                   ; ｱﾌﾟﾘｹｰｼｮﾝ･ｿﾌﾄﾓｰﾄﾞ
            DB      1                   ; 使用ﾎﾞｰﾄﾞ枚数
                                        ;
                                        ; = ﾎﾞｰﾄﾞ登録･起動 =
BOARD       DW      1100H               ; ﾎﾞｰﾄﾞ登録･起動ｺﾏﾝﾄﾞｺｰﾄﾞ
            DB      0                   ; ﾎﾞｰﾄﾞ登録No.
            DB      27                  ; 使用ｾｸﾞﾒﾝﾄｱﾄﾞﾚｽNo.27(D0000H)
            DB      0                   ; 割り込みNo.
            DB      0
                                        ;
                                        ; = ＣＨ整数型ＷＲＩＴＥ =
MEWWT       DW      1511H               ; ｺﾏﾝﾄﾞｺｰﾄﾞ
            DB      0                   ; 登録ﾎﾞｰﾄﾞNO.
            DB      0
            DW      OFFSET SBUF         ; 送信ﾊﾞｯﾌｧOFFSET
            DW      SEG SBUF            ; 送信ﾊﾞｯﾌｧSEGMENT
            DW      0
TXSIZE      DW      0                   ; 送信ﾃﾞｰﾀ･ｻｲｽﾞ
            DB      0                   ;
            DB      0,0,0
            DB      0                   ; 制御ｺｰﾄﾞ
            DB      1                   ; 送信先CH
            DB      0                   ; 送信先ID
            DB      1                   ; 送信元CH
            DB      0                   ; 送信元ID
            DB      0
            DB      0                   ; Depth
TXNO        DB      0                   ; 送信先NO.
            DB      14 DUP(0)
            DW      4 DUP(0)
            DW      0                   ;
                                        ;
                                        ;
                                        ; = ＣＨ整数型ＲＥＡＤ =
MEWRD       DW      1412H               ; ｺﾏﾝﾄﾞｺｰﾄﾞ(完了ｾﾝｽﾀｲﾌﾟ)
            DB      0                   ; 登録ﾎﾞｰﾄﾞNO.
            DB      0
            DW      OFFSET JBUF         ; 受信ﾊﾞｯﾌｧOFFSET
MEWRD2      DW      SEG JBUF
            DW      2048                ; 受信ﾊﾞｯﾌｧｻｲｽﾞ
RXSIZE      DW      0                   ; 受信ﾃﾞｰﾀｻｲｽﾞ用
            DB      0                   ;
            DB      0,0,0
            DB      0                   ; 制御ｺｰﾄﾞ
            DB      0                   ; 送信元CH
            DB      0                   ; 送信元ID
            DB      1                   ; 受信CH
            DB      0                   ; 受信ID
            DB      0
            DB      0                   ; Depth
RXNO        DB      0                   ; 送信元NO.
            DB      14 DUP(0)           ; 受信経路用
            DW      4 DUP(0)
            DW      0                   ;
                                        ;
                                        ; = ＣＨ ＯＰＥＮ =
CHOP        DW      1601H               ; ｺﾏﾝﾄﾞｺｰﾄﾞ
```

ボード登録・起動 ～ 割り込みNo. ：パソコン機種別変更箇所（左記はＰＣ９８シリーズの場合）

```
                DB      0                       ; 登録ﾎﾞｰﾄﾞNO.
                DB      1                       ; OPEN CH NO.
                DB      8 DUP(0)                ; CH状態格納用
                                                ;
                                                ;
;  ============================================
;  =          メッセージ ／ バッファ領域          =
;  ============================================
                                                ;
MSG1            DB      '初期設定===>$'
MSG1_OK         DB      'ＯＫ！'
                DB      CR,LF,'$'

MSG3            DB      'ｴﾗｰ.....ｺｰﾄﾞ='
MSG3_1          DB      0,0,0,0
                DB      CR,LF,'$'

MSG4            DB      '使用ボード登録 & 起動===>$'
MSG4_OK         DB      'ＯＫ！'
                DB      CR,LF,'$'

MSG5            DB      'ＣＨ１ ＯＰＥＮ===>$'
MSG5_OK         DB      'ＯＫ！',CR,LF,'$'

TX_DATA_MSG     DB      '発行データ＝$'
TX_MSG          DB      '数値データ発行===>$'
TX_OK_MSG       DB      'ＯＫ！',CR,LF,'$'

RX_MSG          DB      '数値データ受信センス===>$'
RX_OK_MSG       DB      'ＯＫ！',CR,LF,'$'
RX_BD_MSG       DB      '受信センス出来ず！！',CR,LF,'$'

DSP_RXSIZE      DB      '受信ﾃﾞｰﾀｻｲｽﾞ(Hex)=$'
DSP_RXNO        DB      '送信元No.          =$'
DSP_DATA        DB      '受信データ         =$'

CRLF            DB      CR,LF,'$'

SBUF            DB      2048 DUP(0)                     ; 送信ﾊﾞｯﾌｧ
JBUF            DB      2048 DUP(0)                     ; 受信ﾊﾞｯﾌｧ

STACK           DW      256 DUP(0)                      ; ｽﾀｯｸ領域
STPT            DB      0


CODE    ENDS
        END     START
```

ＰＣ／ＡＴ互換機

```
BOARD           DW      1100H                   ; ﾎﾞｰﾄﾞ登録･起動ｺﾏﾝﾄﾞｺｰﾄﾞ
                DB      0                       ; ﾎﾞｰﾄﾞ登録No.
                DB      8                       ; 使用ｾｸﾞﾒﾝﾄｱﾄﾞﾚｽNo.
                DB      10                      ; 割り込みNo.
                DB      0
```

ＦＭＲシリーズ

```
BOARD           DW      1100H                   ; ﾎﾞｰﾄﾞ登録･起動ｺﾏﾝﾄﾞｺｰﾄﾞ
                DB      0                       ; ﾎﾞｰﾄﾞ登録NO.
                DB      7                       ; 基板I/OﾎﾟｰﾄｱﾄﾞﾚｽNo.
                DW      7800H                   ; ｺﾝﾄﾛｰﾙ･ｽﾃｰﾀｽI/Oﾎﾟｰﾄｱﾄﾞﾚｽ
                DB      5                       ; 割り込みNo.
                DB      0
```

Cプログラム（ソフトウェア割り込みint60）

```
/*************************************************************/
/*                    CH文字型 W/R                           */
/*                      (CHCR.C)                             */
/*************************************************************/
/*                        PC98シリーズ                       */
/*      マスター処理:                                        */
/*      64番機へ文字データを送信します。                     */
/*      64番機から返信があれば発行データを順次変更して       */
/*      いきます。                                           */
/*      64番機をCHCR2タイプで起動させて下さい。              */
/*                                                           */
/*       使用コマンド：WRITE=1501H(ﾀｲﾑｱｳﾄ付き)                */
/*                     READ =1402H(完了ｾﾝｽﾀｲﾌﾟ)              */
/*************************************************************/

#include  <dos.h>
#include  <stdio.h>
#include  <stdlib.h>

#define ECODE "D"                   /* DT                */
#define START_N 0                   /* DT 0 から         */
#defineEND_N  10                    /* DT 10 までﾘｰﾄﾞ     */
#define PC_N  64                    /* 局番 64           */

static union REGS inregs;
static union REGS outregs;
struct SREGS segregs;
static char far *p1;
static unsigned char dat[48];
unsigned char sbuf[2048];           /* 送信ﾊﾞｯﾌｧ  */
unsigned char jbuf[2048];           /* 受信ﾊﾞｯﾌｧ  */

void main( void )
{
        int     i;
        int     err;
        long    lwork;
        char    cwork[5];
        char    c_dmy1[4];
        char    *dmy;
        int     datano,rxsize,txsize;

/* ---------------------------*/
/* ------ 初期設定処理 ------*/
/* ---------------------------*/

        dat[0] = 0x00;          /* 初期設定ｺﾏﾝﾄﾞ･ｺｰﾄﾞ=1000H  */
        dat[1] = 0x10;
        dat[2] = 0;             /* ｱﾌﾟﾘｹｰｼｮﾝ･ｿﾌﾄﾓｰﾄﾞ=0       */
        dat[3] = 1;             /* 使用ボード枚数=1          */

        printf("初期設定===>");

        if((err = mew_send()) != 0)
        {
                printf("ｴﾗｰ.....ｺｰﾄﾞ= %x¥n", err);
                exit(-1);
        }
        printf("ＯＫ！¥n");

/* ------------------------------------*/
/* ------ 使用ボード登録 及び起動 ------*/
/* ------------------------------------*/

        dat[0] = 0x00;          /* 使用ボード登録及び起動   */
        dat[1] = 0x11;          /* ｺﾏﾝﾄﾞｺｰﾄﾞ=1100H          */
        dat[2] = 0x00;          /* 登録ボードNo.=0          */
        dat[3] = 27;            /* 使用ｾｸﾞﾒﾝﾄｱﾄﾞﾚｽNo.=27    */
        dat[4] = 0;             /* 割り込みNo.=0            */

        printf("使用ボード登録 & 起動===>");

        if((err = mew_send()) != 0)
```

パソコン機種別変更箇所
（左記はＰＣ９８シリーズの場合）

```
        {
                printf("ｴﾗｰ.....ｺｰﾄﾞ= %x¥n", err);
                exit(-1);
        }
        printf("ＯＫ！¥n");

/* ----------------------------*/
/* ------ ＣＨ１ ＯＰＥＮ ------*/
/* ----------------------------*/

        dat[0] = 0x01;          /* CH OPEN ｺﾏﾝﾄﾞｺｰﾄﾞ=1601H   */
        dat[1] = 0x16;
        dat[2] = 0x00;          /* 登録ﾎﾞｰﾄﾞNo.=0            */
        dat[3] = 0x01;          /* OPEN CH No. =1           */

        printf("ＣＨ１ ＯＰＥＮ===>");

        if((err = mew_send()) != 0)
        {
                printf("ｴﾗｰ...ｺｰﾄﾞ= %x¥n",err);
                exit(-1);
        }
        printf("ＯＫ！¥n");

/* -------------------------------*/
/* ------ ＣＨ文字型 ＷＲＩＴ ------*/
/* -------------------------------*/

        datano = 0;
TXLP:
        memset( sbuf, 0, sizeof( sbuf ) );
        switch(datano)
        {
                case 0:
                        sprintf(sbuf,"AAAAAAAAAA");
                        txsize = 10;
                        break;
                case 1:
                        sprintf(sbuf,"BBBBBBBBBBBBBB");
                        txsize = 15;
                        break;
                case 2:
                        sprintf(sbuf,"CCCCCCCCCCCCCCCCCCCC");
                        txsize = 20;
                        break;
                case 3:
                        sprintf(sbuf,"DDDDDDDDDDDDDDDDDDDDDDDDDDDDDD");
                        txsize = 30;
                        break;
        }
TXLP2:
        printf("発行データ＝%s¥n",sbuf);
        p1 = (char far *)&sbuf[0];

        dat[0] = 0x01;              /* CH文字型WRITE      */
        dat[1] = 0x15;              /* ｺﾏﾝﾄﾞｺｰﾄﾞ=1501H    */
        dat[2] = 0x00;              /* 登録ボードno.=0    */
        dat[3] = 0x00;
        *((unsigned *)&dat[4]) = FP_OFF(p1);        /* 送信ﾊﾞｯﾌｧｵﾌｾｯﾄ     */
        *((unsigned *)&dat[6]) = FP_SEG(p1);        /* 送信ﾊﾞｯﾌｧｾｸﾞﾒﾝﾄ    */
        *((unsigned *)&dat[10]) = txsize;           /* 送信ﾃﾞｰﾀｻｲｽﾞ        */
        dat[16] = 0;            /* 書き込み制御ｺｰﾄﾞ   */
        dat[17] = 1;            /* 送信先CH=1         */
        dat[18] = 0;            /* 送信先ID=0         */
        dat[19] = 1;            /* 送信元CH=1         */
        dat[20] = 0;            /* 送信元ID=0         */
        dat[22] = 0;            /* Depth=0(階層なし)  */
        dat[23] = PC_N;         /* 送信先No.(PC局番)  */

        printf("文字データ発行===>");

        if((err = mew_send()) != 0)
        {
                printf("ｴﾗｰ.....ｺｰﾄﾞ= %x ¥n", err);
                goto TXLP2;
        }
```

```
        printf("ＯＫ！¥n");


/* ---------------------------------*/
/* ------ ＣＨ文字型 ＲＥＡＤ ------*/
/* ---------------------------------*/

        memset( jbuf, 0, sizeof( jbuf ) );
        p1 = (char far *)&jbuf[0];

        dat[0] = 0x02;              /* CH文字型READ     */
        dat[1] = 0x14;              /* ｺﾏﾝﾄﾞｺｰﾄﾞ=1402H  */
        dat[2] = 0x00;              /* 登録ボードno.=0  */
        dat[3] = 0x00;              /* 受信ch no.       */
        *((unsigned *)&dat[4]) = FP_OFF(p1);       /* 受信ﾊﾞｯﾌｧｵﾌｾｯﾄ   */
        *((unsigned *)&dat[6]) = FP_SEG(p1);       /* 受信ﾊﾞｯﾌｧｾｸﾞﾒﾝﾄ */
        *((unsigned *)&dat[8]) = 2048;             /* 受信ﾊﾞｯﾌｧｻｲｽﾞ    */
        dat[16] = 0;                /* 受信制御ｺｰﾄﾞ     */
        dat[19] = 1;                /* 受信CH No.=1     */

        printf("文字データ受信センス===>");

        if((err = mew_send()) != 0)
        {
                if(err == 0x133)
                        printf("受信センス出来ず！！¥n");
                else
                        printf("ｴﾗｰ.....ｺｰﾄﾞ= %x¥n", err);
        }
        else
        {
                printf("ＯＫ！¥n");

                rxsize = dat[10] + dat[11] * 0x100;
                printf("受信ﾃﾞｰﾀｻｲｽﾞ=%d¥n",rxsize);
                printf("送信元No.    =%d¥n",dat[23]);
                printf("受信データ   =%s¥n",jbuf);

                if(datano == 3)
                        datano = 0;
                else
                        datano = datano + 1;
        }
        printf("¥n");
        goto TXLP;
}


/* ----------------------------*/
/* ------ ﾘﾝｸｿﾌﾄ呼び出し------*/
/* ----------------------------*/
int mew_send(void)
{

        p1 = (char far *)&dat[0];

        inregs.x.dx = FP_OFF(p1);
        segregs.ds = FP_SEG(p1);

        int86x( 0x60, &inregs, &outregs, &segregs);
        return(outregs.x.ax);
}
```

```
/*************************************************************/
/*                    ＣＨ文字型 Ｗ／Ｒ                      */
/*                        (CHCR2.C)                          */
/*************************************************************/
/*                    ＰＣ９８シリーズ                       */
/*      スレーブ処理:                                        */
/*      マスターからのデータを受信したら応答を返します。     */
/*      （受信各文字に＋２０ｈした文字を返却します。）       */
/*      本プログラムを起動するパソコンＩ／Ｆボードは         */
/*      ６４番機として下さい。                               */
/*      マスター局側をCHCRタイプで起動させて下さい。         */
/*                                                           */
/*       使用コマンド：WRITE=1501H(ﾀｲﾑｱｳﾄ付き)               */
/*                     READ =1402H(完了ｾﾝｽﾀｲﾌﾟ)              */
/*************************************************************/

#include  <dos.h>
#include  <stdio.h>
#include  <stdlib.h>

#define ECODE "D"           /* DT                  */
#define START_N 0           /* DT 0 から           */
#defineEND_N  10            /* DT 10 までﾘｰﾄﾞ       */

static union REGS inregs;
static union REGS outregs;
struct SREGS segregs;
static char far *p1;
static unsigned char dat[48];
unsigned char sbuf[2048];           /* 送信ﾊﾞｯﾌｧ  */
unsigned char jbuf[2048];           /* 受信ﾊﾞｯﾌｧ  */

void main( void )
{
       int    i;
       int    err;
       long   lwork;
       char   cwork[5];
       char   c_dmy1[4];
       char   *dmy,rxno;
       int    datano,rxsize,lped;

/* -------------------------*/
/* ------ 初期設定処理 ------*/
/* -------------------------*/

       dat[0] = 0x00;       /* 初期設定ｺﾏﾝﾄﾞ･ｺｰﾄﾞ=1000H  */
       dat[1] = 0x10;
       dat[2] = 0;          /* ｱﾌﾟﾘｹｰｼｮﾝ･ｿﾌﾄﾓｰﾄﾞ=0       */
       dat[3] = 1;          /* 使用ボード枚数=1          */

       printf("初期設定===>");

       if((err = mew_send()) != 0)
       {
              printf("ｴﾗｰ.....ｺｰﾄﾞ=  %x¥n", err);
              exit(-1);
       }
       printf("ＯＫ！¥n");

/* ---------------------------------*/
/* ------ 使用ボード登録 及び起動 ------*/
/* ---------------------------------*/

       dat[0] = 0x00;       /* 使用ボード登録及び起動    */
       dat[1] = 0x11;       /* ｺﾏﾝﾄﾞｺｰﾄﾞ=1100H           */
       dat[2] = 0x00;       /* 登録ボードNo.=0           */
       dat[3] = 27;         /* 使用ｾｸﾞﾒﾝﾄｱﾄﾞﾚｽNo.=27    */
       dat[4] = 0;          /* 割り込みNo.=0             */

       printf("使用ボード登録 & 起動===>");

       if((err = mew_send()) != 0)
```

パソコン機種別変更箇所
（左記はＰＣ９８シリーズの場合）

```
        {
                printf("ｴﾗｰ.....ｺｰﾄﾞ=  %x¥n",  err);
                exit(-1);
        }
        printf("ＯＫ！¥n");

/* ----------------------------*/
/* ------ ＣＨ１ ＯＰＥＮ ------*/
/* ----------------------------*/

        dat[0] = 0x01;          /* CH OPEN ｺﾏﾝﾄﾞｺｰﾄﾞ=1601H  */
        dat[1] = 0x16;
        dat[2] = 0x00;          /* 登録ﾎﾞｰﾄﾞNo.=0      */
        dat[3] = 0x01;          /* OPEN CH No. =1     */

        printf("ＣＨ１ ＯＰＥＮ===>");

        if((err = mew_send()) != 0)
        {
                printf("ｴﾗｰ...ｺｰﾄﾞ=  %x¥n",err);
                exit(-1);
        }
        printf("ＯＫ！¥n");

/* --------------------------------*/
/* ------ ＣＨ文字型 ＲＥＡＤ ------*/
/* --------------------------------*/

RXLP:
        lped = 0;
        memset( jbuf, 0, sizeof( jbuf ) );

        do
        {
                p1 = (char far *)&jbuf[0];

                dat[0] = 0x02;                  /* CH文字型READ         */
                dat[1] = 0x14;                  /* ｺﾏﾝﾄﾞｺｰﾄﾞ=1402H      */
                dat[2] = 0x00;                  /* 登録ボードno.=0      */
                dat[3] = 0x00;                  /* 受信ch no.  */
                *((unsigned *)&dat[4]) = FP_OFF(p1);        /* 受信ﾊﾞｯﾌｧｵﾌｾｯﾄ       */
                *((unsigned *)&dat[6]) = FP_SEG(p1);;       /* 受信ﾊﾞｯﾌｧｾｸﾞﾒﾝ       */
                *((unsigned *)&dat[8]) = 2048;              /* 受信ﾊﾞｯﾌｧｻｲｽﾞ        */
                dat[16] = 0;                    /* 受信制御ｺｰﾄﾞ         */
                dat[19] = 1;                    /* 受信CH No.=1         */

                printf("文字データ受信センス===>");

                if((err = mew_send()) != 0)
                {
                        if(err == 0x133)
                                printf("受信センス出来ず！！¥n");
                        else
                                printf("ｴﾗｰ.....ｺｰﾄﾞ=  %x¥n",  err);
                }
                else
                {
                        printf("ＯＫ！¥n");

                        rxsize = dat[10] + dat[11] * 0x100;
                        printf("受信ﾃﾞｰﾀｻｲｽﾞ=%d¥n",rxsize);
                        rxno = dat[23];
                        printf("送信元No.    =%d¥n",rxno);
                        printf("受信データ   =%s¥n",jbuf);
                        memset( sbuf, 0, sizeof( sbuf ) );
                        for(i = 0 ; i < rxsize ; i++)
                        {
                                sbuf[i] = jbuf[i] + 0x20;
                        }
                        lped = 1;
                }
        }
        while(lped == 0);


/* --------------------------------*/
/* ------ ＣＨ文字型 ＷＲＩＴ ------*/
```

```
/* ---------------------------------*/

        printf("発行データ=%s¥n",sbuf);
        p1 = (char far *)&sbuf[0];

        dat[0] = 0x01;              /* CH文字型WRITE     */
        dat[1] = 0x15;              /* ｺﾏﾝﾄﾞｺｰﾄﾞ=1501H   */
        dat[2] = 0x00;              /* 登録ボードno.=0   */
        dat[3] = 0x00;
        *((unsigned *)&dat[6]) = FP_SEG(p1);       /* 送信ﾊﾞｯﾌｧｾｸﾞﾒﾝﾄ   */
        *((unsigned *)&dat[10]) = rxsize;          /* 送信ﾃﾞｰﾀｻｲｽﾞ       */
        dat[16] = 0;                /* 書き込み制御ｺｰﾄﾞ    */
        dat[17] = 1;                /* 送信先CH=1          */
        dat[18] = 0;                 /* 送信先ID=0         */
        dat[19] = 1;                /* 送信元CH=1          */
        dat[20] = 0;                /* 送信元ID=0          */
        dat[22] = 0;                /* Depth=0(階層なし)   */
        dat[23] = rxno;             /* 送信先No.(PC局番)   */

        printf("文字データ発行===>");

        if((err = mew_send()) != 0)
        {
                printf("ｴﾗｰ.....ｺｰﾄﾞ= %x ¥n", err);
                goto RXLP;
        }

        printf("ＯＫ！¥n");
        goto RXLP;
}


/* ---------------------------*/
/* ------ ﾘﾝｸｿﾌﾄ呼び出し------*/
/* ---------------------------*/
int mew_send(void)
{

        p1 = (char far *)&dat[0];

        inregs.x.dx = FP_OFF(p1);
        segregs.ds = FP_SEG(p1);

        int86x( 0x60, &inregs, &outregs, &segregs);
        return(outregs.x.ax);
}
```

```
/*************************************************************/
/*                      ＣＨ整数型 Ｗ／Ｒ                    */
/*                        (CHIT.C)                           */
/*************************************************************/
/*                        ＰＣ９８シリー                     */
/*      マスター処理:                                        */
/*      ６４番機へ数値データを送信します。                   */
/*      ６４番機から返信があれば発行データを順次変更して     */
/*      いきます。                                           */
/*      ６４番機をCHIT2タイプで起動させて下さい。           */
/*                                                           */
/*      使用コマンド：WRITE=1511H(ﾀｲﾑｱｳﾄ付き)                */
/*                    READ =1412H(完了ｾﾝｽﾀｲﾌﾟ)               */
/*************************************************************/
/
```

```
#include  <dos.h>
#include  <stdio.h>
#include  <stdlib.h>

#define ECODE "D"           /* DT                  */
#define START_N 0           /* DT 0 から           */
#defineEND_N  10            /* DT 10 までﾘｰﾄﾞ      */
#define PC_N  64            /* 局番 64             */

static union REGS inregs;
static union REGS outregs;
struct SREGS segregs;
static char far *p1;
static unsigned char dat[48];
unsigned char sbuf[2048];           /* 送信ﾊﾞｯﾌｧ  */
unsigned char jbuf[2048];           /* 受信ﾊﾞｯﾌｧ  */

void main( void )
{
       int    i;
       int    err;
       long   lwork;
       char   cwork[5];
       char   c_dmy1[4];
       char   *dmy;
       int    datano,rxsize,datacd,txsize;

/* -------------------------*/
/* ------ 初期設定処理 ------*/
/* -------------------------*/

       dat[0] = 0x00;       /* 初期設定ｺﾏﾝﾄﾞ･ｺｰﾄﾞ=1000H  */
       dat[1] = 0x10;
       dat[2] = 0;          /* ｱﾌﾟﾘｹｰｼｮﾝ･ｿﾌﾄﾓｰﾄﾞ=0       */
       dat[3] = 1;          /* 使用ボード枚数=1          */

       printf("初期設定===>");

       if((err = mew_send()) != 0)
       {
              printf("ｴﾗｰ.....ｺｰﾄﾞ=  %x¥n",  err);
              exit(-1);
       }
       printf("ＯＫ！¥n");

/* ---------------------------------*/
/* ------ 使用ボード登録 及び起動 ------*/
/* ---------------------------------*/

       dat[0] = 0x00;       /* 使用ボード登録及び起動   */
       dat[1] = 0x11;       /* ｺﾏﾝﾄﾞｺｰﾄﾞ=1100H          */
       dat[2] = 0x00;       /* 登録ボードno.=0          */
       dat[3] = 27;         /* 使用ｾｸﾞﾒﾝﾄｱﾄﾞﾚｽNo=27     */
       dat[4] = 0;          /* 割り込みレベルno.=0      */

       printf("使用ボード登録 & 起動===>");

       if((err = mew_send()) != 0)
       {
              printf("ｴﾗｰ.....ｺｰﾄﾞ=  %x¥n",  err);
              exit(-1);
       }
       printf("ＯＫ！¥n");

/* ---------------------------*/
/* ------ ＣＨ１ ＯＰＥＮ ------*/
/* ---------------------------*/

       dat[0] = 0x01;       /* CH OPEN ｺﾏﾝﾄﾞｺｰﾄﾞ=1601H  */
       dat[1] = 0x16;
       dat[2] = 0x00;       /* 登録ﾎﾞｰﾄﾞNo.=0           */
       dat[3] = 0x01;       /* OPEN CH No. =1           */

       printf("ＣＨ１ ＯＰＥＮ===>");

       if((err = mew_send()) != 0)
```

パソコン機種別変更箇所
（左記はＰＣ９８シリーズの場合）

```
        {
                printf("ｴﾗｰ...ｺｰﾄﾞ=  %x¥n",err);
                exit(-1);
        }
        printf("ＯＫ！¥n");

/* ---------------------------------*/
/* ------ ＣＨ整数型 ＷＲＩＴ ------*/
/* ---------------------------------*/

        datano = 0;
TXLP:
        memset( sbuf, 0, sizeof( sbuf ) );
        switch(datano)
        {
                case 0:
                        datacd = 0;
                        txsize = 10;
                        break;
                case 1:
                        datacd = 10;
                        txsize = 20;
                        break;
                case 2:
                        datacd = 30;
                        txsize = 30;
                        break;
                case 3:
                        datacd = 70;
                        txsize = 30;
                        break;
        }

        for(i = 0 ; i < txsize ; i++)
        {
                sbuf[i] = datacd + i;
        }
TXLP2:
        printf("発行データ=");
        for(i = 0 ; i < txsize ; i++)
        {
                printf("%d",sbuf[i]);
        }
        printf("¥n");

        p1 = (char far *)&sbuf[0];

        dat[0] = 0x11;          /* CH整数型WRITE      */
        dat[1] = 0x15;          /* ｺﾏﾝﾄﾞｺｰﾄﾞ=1511H    */
        dat[2] = 0x00;          /* 登録ボードno.=0    */
        dat[3] = 0x00;
        *((unsigned *)&dat[4]) = FP_OFF(p1);        /* 送信ﾊﾞｯﾌｧｵﾌｾｯﾄ     */
        *((unsigned *)&dat[6]) = FP_SEG(p1);        /* 送信ﾊﾞｯﾌｧｾｸﾞﾒﾝﾄ    */
        *((unsigned *)&dat[10]) = txsize;           /* 送信ﾃﾞｰﾀｻｲｽ        */
        dat[16] = 0;            /* 書き込み制御ｺｰﾄﾞ   */
        dat[17] = 1;            /* 送信先CH=1  */
        dat[18] = 0;            /* 送信先ID=0  */
        dat[19] = 1;            /* 送信元CH=1  */
        dat[20] = 0;            /* 送信元ID=0  */
        dat[22] = 0;            /* Depth=0(階層なし)  */
        dat[23] = PC_N;         /* 送信先No.(PC局番)  */

        printf("数値データ発行===>");

        if((err = mew_send()) != 0)
        {
                printf("ｴﾗｰ.....ｺｰﾄﾞ=  %x ¥n", err);
                goto TXLP2;
        }

        printf("ＯＫ！¥n");


/* ---------------------------------*/
/* ------ ＣＨ整数型 ＲＥＡＤ ------*/
/* ---------------------------------*/
```

```
        memset( jbuf, 0, sizeof( jbuf ) );
        p1 = (char far *)&jbuf[0];

        dat[0] = 0x12;                  /* CH整数型READ     */
        dat[1] = 0x14;                  /* ｺﾏﾝﾄﾞｺｰﾄﾞ=1412H  */
        dat[2] = 0x00;                  /* 登録ボードno.=0  */
        dat[3] = 0x00;                  /* 受信ch no.       */
        *((unsigned *)&dat[4]) = FP_OFF(p1);        /* 受信ﾊﾞｯﾌｧｵﾌｾｯﾄ   */
        *((unsigned *)&dat[6]) = FP_SEG(p1);        /* 受信ﾊﾞｯﾌｧｾｸﾞﾒﾝﾄ */
        *((unsigned *)&dat[8]) = 2048;              /* 受信ﾊﾞｯﾌｧｻｲｽﾞ    */
        dat[16] = 0;                    /* 受信制御ｺｰﾄﾞ     */
        dat[19] = 1;                    /* 受信CH No.=1     */

        printf("数値データ受信センス===>");

        if((err = mew_send()) != 0)
        {
                if(err == 0x133)
                        printf("受信センス出来ず！！¥n");
                else
                        printf("ｴﾗｰ.....ｺｰﾄﾞ= %x¥n", err);
        }
        else
        {
                printf("ＯＫ！¥n");

                rxsize = dat[10] + dat[11] * 0x100;
                printf("受信ﾃﾞｰﾀｻｲｽﾞ=%d¥n",rxsize);
                printf("送信元No.    =%d¥n",dat[23]);
                printf("受信データ   =");
                for(i = 0 ; i < rxsize ; i++)
                {
                        printf("%d",jbuf[i]);
                }
                printf("¥n");

                if(datano == 3)
                        datano = 0;
                else
                        datano = datano + 1;
        }
        printf("¥n");
        goto TXLP;
}


/* ---------------------------*/
/* ------ ﾘﾝｸｿﾌﾄ呼び出し------*/
/* ---------------------------*/
int mew_send(void)
{

        p1 = (char far *)&dat[0];

        inregs.x.dx = FP_OFF(p1);
        segregs.ds = FP_SEG(p1);

        int86x( 0x60, &inregs, &outregs, &segregs);
        return(outregs.x.ax);
}
```

```
/*****************************************************************/
/*                  ＣＨ整数型 Ｗ／Ｒ                            */
/*                          (CHIT2.C)                            */
/*****************************************************************/
/*                    ＰＣ９８シリーズ                           */
/*      スレーブ処理:                                            */
/*      マスターからのデータを受信したら応答を返します。         */
/*      （受信データをそのまま返却します。）                     */
/*      本プログラムを起動するパソコンＩ／Ｆボードは             */
/*      ６４番機として下さい。                                   */
/*      マスター局側をCHITタイプで起動させて下さい。             */
/*                                                               */
/*       使用コマンド：WRITE=1511H(ﾀｲﾑｱｳﾄ付き)                   */
/*                     READ =1412H(完了ｾﾝｽﾀｲﾌﾟ)                  */
/*****************************************************************/

#include  <dos.h>
#include  <stdio.h>
#include  <stdlib.h>

#define ECODE "D"           /* DT                  */
#define START_N 0           /* DT 0 から           */
#defineEND_N  10            /* DT 10 までﾘｰﾄﾞ      */

static union REGS inregs;
static union REGS outregs;
struct SREGS segregs;
static char far *p1;
static unsigned char dat[48];
unsigned char sbuf[2048];           /* 送信ﾊﾞｯﾌｧ  */
unsigned char jbuf[2048];           /* 受信ﾊﾞｯﾌｧ  */

void main( void )
{
       int    i;
       int    err;
       long   lwork;
       char   cwork[5];
       char   c_dmy1[4];
       char   *dmy,rxno;
       int    datano,rxsize,lped;

/* -------------------------*/
/* ------ 初期設定処理 ------*/
/* -------------------------*/

       dat[0] = 0x00;       /* 初期設定ｺﾏﾝﾄﾞ･ｺｰﾄﾞ=1000H  */
       dat[1] = 0x10;
       dat[2] = 0;          /* ｱﾌﾟﾘｹｰｼｮﾝ･ｿﾌﾄﾓｰﾄﾞ=0       */
       dat[3] = 1;          /* 使用ボード枚数=1   */

       printf("初期設定===>");

       if((err = mew_send()) != 0)
       {
              printf("ｴﾗｰ.....ｺｰﾄﾞ=  %x¥n",  err);
              exit(-1);
       }
       printf("ＯＫ！¥n");

/* ---------------------------------*/
/* ------ 使用ボード登録 及び起動 ------*/
/* ---------------------------------*/

       dat[0] = 0x00;       /* 使用ボード登録及び起動   */
       dat[1] = 0x11;       /* ｺﾏﾝﾄﾞ･ｺｰﾄﾞ=1100H        */
       dat[2] = 0x00;       /* 登録ボードNo.=0          */
       dat[3] = 27;         /* 使用ｾｸﾞﾒﾝﾄｱﾄﾞﾚｽNo.=27    */
       dat[4] = 0;          /* 割り込みNo.=0            */

       printf("使用ボード登録 & 起動===>");

       if((err = mew_send()) != 0)
```

パソコン機種別変更箇所
（左記はＰＣ９８シリーズの場合）

```
        {
                printf("ｴﾗｰ.....ｺｰﾄﾞ= %x¥n", err);
                exit(-1);
        }
        printf("ＯＫ！¥n");

/* ----------------------------*/
/* ------ ＣＨ１ ＯＰＥＮ ------*/
/* ----------------------------*/

        dat[0] = 0x01;          /* CH OPEN ｺﾏﾝﾄﾞｺｰﾄﾞ=1601H   */
        dat[1] = 0x16;
        dat[2] = 0x00;          /* 登録ﾎﾞｰﾄﾞNo.=0     */
        dat[3] = 0x01;          /* OPEN CH No. =1     */

        printf("ＣＨ１ ＯＰＥＮ===>");

        if((err = mew_send()) != 0)
        {
                printf("ｴﾗｰ...ｺｰﾄﾞ= %x¥n",err);
                exit(-1);
        }
        printf("ＯＫ！¥n");

/* --------------------------------*/
/* ------ ＣＨ整数型 ＲＥＡＤ ------*/
/* --------------------------------*/

RXLP:
        lped = 0;
        memset( jbuf, 0, sizeof( jbuf ) );

        do
        {
                p1 = (char far *)&jbuf[0];

                dat[0] = 0x12;          /* CH整数型READ       */
                dat[1] = 0x14;          /* ｺﾏﾝﾄﾞｺｰﾄﾞ=1412H    */
                dat[2] = 0x00;          /* 登録ボードno.=0    */
                dat[3] = 0x00;          /* 受信ch no. */
                *((unsigned *)&dat[4]) = FP_OFF(p1);        /* 受信ﾊﾞｯﾌｧｵﾌｾｯﾄ     */
                *((unsigned *)&dat[6]) = FP_SEG(p1);        /* 受信ﾊﾞｯﾌｧｾｸﾞﾒﾝﾄ    */
                *((unsigned *)&dat[8]) = 2048;              /* 受信ﾊﾞｯﾌｧｻｲｽﾞ       */
                dat[16] = 0;                    /* 受信制御ｺｰﾄﾞ        */
                dat[19] = 1;                    /* 受信CH No.=1        */

                printf("数値データ受信センス===>");

                if((err = mew_send()) != 0)
                {
                        if(err == 0x133)
                                printf("受信センス出来ず！！¥n");
                        else
                                printf("ｴﾗｰ.....ｺｰﾄﾞ= %x¥n", err);
                }
                else
                {
                        printf("ＯＫ！¥n");

                        rxsize = dat[10] + dat[11] * 0x100;
                        printf("受信ﾃﾞｰﾀｻｲｽﾞ=%d¥n",rxsize);
                        rxno = dat[23];
                        printf("送信元No.    =%d¥n",rxno);
                        printf("受信データ   =");
                        memset( sbuf, 0, sizeof( sbuf ) );
                        for(i = 0 ; i < rxsize ; i++)
                        {
                                printf("%d",jbuf[i]);
                                sbuf[i] = jbuf[i];
                        }
                        printf("¥n");
                        lped = 1;
                }
        }
        while(lped == 0);


/* --------------------------------*/
/* ------ ＣＨ数値型 ＷＲＩＴ ------*/
```

```
/* ---------------------------------*/

        printf("発行データ=");
        for(i = 0 ; i < rxsize ; i++)
        {
                printf("%d",sbuf[i]);
        }
        printf("¥n");
        p1 = (char far *)&sbuf[0];

        dat[0] = 0x11;                  /* CH整数型WRITE     */
        dat[1] = 0x15;                  /* ｺﾏﾝﾄﾞｺｰﾄﾞ=1511H   */
        dat[2] = 0x00;/* 登録ボードno.=0   */
        dat[3] = 0x00;
        *((unsigned *)&dat[4]) = FP_OFF(p1);        /* 送信ﾊﾞｯﾌｧｵﾌｾｯﾄ    */
        *((unsigned *)&dat[6]) = FP_SEG(p1);        /* 送信ﾊﾞｯﾌｧｾｸﾞﾒﾝﾄ   */
        *((unsigned *)&dat[10]) = rxsize;           /* 送信ﾃﾞｰﾀｻｲｽﾞ       */
        dat[16] = 0;            /* 書き込み制御ｺｰﾄﾞ  */
        dat[17] = 1;            /* 送信先CH=1        */
        dat[18] = 0;            /* 送信先ID=0        */
        dat[19] = 1;            /* 送信元CH=1        */
        dat[20] = 0;            /* 送信元ID=0        */
        dat[22] = 0;            /* Depth=0(階層なし)*/
        dat[23] = rxno;         /* 送信先No.(PC局番)*/

        printf("数値データ発行===>");

        if((err = mew_send()) != 0)
        {
                printf("ｴﾗｰ.....ｺｰﾄﾞ= %x ¥n", err);
                goto RXLP;
        }

        printf("ＯＫ！¥n");
        goto RXLP;
}


/* ----------------------------*/
/* ------ ﾘﾝｸｿﾌﾄ呼び出し------*/
/* ----------------------------*/
int mew_send(void)
{

        p1 = (char far *)&dat[0];

        inregs.x.dx = FP_OFF(p1);
        segregs.ds = FP_SEG(p1);

        int86x( 0x60, &inregs, &outregs, &segregs);
        return(outregs.x.ax);
}
```

パソコン機種別変更箇所

ＰＣ／ＡＴ互換機

```
        dat[0] = 0x00;          /* 使用ボード登録及び起動    */
        dat[1] = 0x11;          /* ｺﾏﾝﾄﾞｺｰﾄﾞ=1100H           */
        dat[2] = 0x00;          /* 登録ボードNo.=0           */
        dat[3] = 8;             /* 使用ｾｸﾞﾒﾝﾄｱﾄﾞﾚｽNo.=8      */
        dat[4] = 10;            /* 割り込みNo.=10            */
```

ＦＭＲシリーズ

```
        dat[0] = 0x00;          /* 使用ボード登録及び起動    */
        dat[1] = 0x11;          /* ｺﾏﾝﾄﾞｺｰﾄﾞ=1100H           */
        dat[2] = 0x00;          /* 登録ボードNo.=0           */
        dat[3] = 7;             /* 基板I/OﾎﾟｰﾄｱﾄﾞﾚｽNo.=7     */
        dat[4] = 0x00;          /* ｽﾃｰﾀｽ･ｺﾝﾄﾛｰﾙI/Oﾎﾟｰﾄｱﾄﾞﾚｽ */
        dat[5] = 0x78;          /* =7800H                    */
        dat[6] = 5;             /* 割り込みNo.=5      */
```

3－7　登録受信要求解除の機能コマンド

3-7-1 登録受信要求タイプコマンド（タイムアウト無し）解除

コンピュータリンク、データ転送、コンピュータ間通信／登録受信タイムアウト無しタイプ解除（コマンドコード&H1460）

機　能

指定された登録受信タイムアウト無しタイプの登録解除を行います。

解　説

・機能コマンドを実行するには、各種プログラミング言語でI/Oパラメータのセグメントアドレスとオフセットアドレスをds:dxレジスタにセットし、割り込みを実行するユーザープログラムを作成してください。
・コマンド実行後の結果（リターン値）は、axレジスタに格納されます。
　0：正常終了時
　その他：エラーコード（「3－9」参照）

注　意

・本コマンドで解除できるのは、登録受信タイムアウト無しタイプで登録したものだけです。
登録時にI/Oパラメータ（アドレス2Ch）に格納される登録No.を指定してください。

文　例

●MS-DOSファンクションコール（int21h）

```
;----- 登録受信タイムアウト無しタイプ解除 ----
R_END:
        mov     bx,[FILHDL]
        mov     ah,44h
        mov     al,03h
        mov     dx,offset R_END
        int     21h
;------- I/Oﾊﾟﾗﾒｰﾀ ------
R_END   dw      1460h
        dw      0,0,21 dup(?)
```

●ソフトウェア割り込み（int60h）

```
;----- 登録受信タイムアウト無しタイプ解除 ----
PCRRD:
        mov     dx,offset R_END
        int     60h
;------- I/Oﾊﾟﾗﾒｰﾀ ------
R_END   dw      1460h
        dw      0,0,21 dup(?)
```

I/Oパラメータ

△：設定する［渡す値］
▼：実行後に格納される［得る値］

dsレジスタ ←I/Oパラメータの先頭セグメントアドレス
dxレジスタ ←I/Oパラメータの先頭オフセットアドレス

0	△［コマンドコード］&H1460
2	△［登録ボードNo.］0～3（使用するリンクボードの登録No.） △設定値：00固定で使用してください。
4	△［受信登録No.］登録受信要求（タイムアウト無し）タイプコマンドの登録No.を指定します。
6	
8	
A	
C	
E	
10	
12	
14	
16	
18	
1A	
1C	
1E	
20	
22	
24	
26	
28	
2A	
2C	
2E	

3-7-2 登録受信解除のプログラム例

アセンブラプログラム（MS-DOSシステムコールint21）

```
; ********************************************************
; *        ＣＨ文字型 Ｗ／Ｒ 登録受信及び解除タイプ      *
; *                    <ACHCR3.ASM>                      *
; ********************************************************
; *                    ＰＣ９８シリーズ                  *
; *  スレーブ処理:                                       *
; *  マスターからのデータを受信したら応答を返します。    *
; *   （受信各文字に＋２０ｈした文字を返却します。）     *
; *   データの受信は、登録タイムアウト無しタイプで行い   *
; *   ます。                                             *
; *   受信完了センスで１０回連続センス出来なかった場合、*
; *  一旦登録を解除します。                              *
; *  本プログラムを起動するパソコンＩ／Ｆボードは        *
; *  ６４番機として下さい。                              *
; *  マスター局側をCHCRﾀｲﾌﾟで起動させて下さい。          *
; *                                                      *
; *   使用コマンド：WRITE=1501H(ﾀｲﾑｱｳﾄ付き)              *
; *                 READ =1404H(登録受信ﾀｲﾑｱｳﾄ無しﾀｲﾌﾟ)  *
; ********************************************************
                                        ;
CODE    SEGMENT
        ASSUME CS:CODE,DS:CODE,ES:CODE,SS:CODE


CR              EQU     0DH             ; CRｺｰﾄﾞ
LF              EQU     0AH             ; LFｺｰﾄﾞ
                                        ;
                                        ;
START:
                MOV     AX,CS
                MOV     DS,AX
                MOV     ES,AX
                MOV     SS,AX
                MOV     AX,OFFSET STPT
                MOV     SP,AX
                                        ;
;  ---------------------------------------------
;                 初 期 設 定
;  ---------------------------------------------
INI:
                MOV     DX,OFFSET MSG1  ; 初期設定メッセージ表示
                CALL    MESSAGE
                MOV     DX,OFFSET INIT  ; 初期設定用I/Oﾊﾟﾗﾒｰﾀ ｱﾄﾞﾚｽ設定
                CALL    COMMAND         ; ﾘﾝｸｿﾌﾄ呼出
                JNB     INI_OK
                JMP     DOS             ; ｴﾗｰ時 MS-DOSへ戻る
                                        ;
INI_OK:
                MOV     DX,OFFSET OK_MSG ; 初期設定ＯＫﾒｯｾｰｼﾞ 表示
                CALL    MESSAGE
                                        ;
;  ---------------------------------------------
;        ボ ー ド 登 録 ・ 起 動
;  ---------------------------------------------
                                        ;
                MOV     DX,OFFSET MSG4  ; ﾎﾞｰﾄﾞ登録･起動メッセージ表示
                CALL    MESSAGE
                MOV     DX,OFFSET BOARD ; ﾎﾞｰﾄﾞ登録用I/Oﾊﾟﾗﾒｰﾀ ｱﾄﾞﾚｽ
                CALL    COMMAND         ; ﾘﾝｸｿﾌﾄ呼出
                JNB     BDST_OK
                JMP     DOS             ; ｴﾗｰ時 MS-DOSへ戻る
                                        ;
BDST_OK:
                MOV     DX,OFFSET OK_MSG ; ﾎﾞｰﾄﾞ登録&起動OKﾒｯｾｰｼﾞ 表示
                CALL    MESSAGE
                                        ;
;  ---------------------------------------------
;              CH－1  OPEN
```

```
; ------------------------------------------
                                        ;
CH1OP:
            MOV     DX,OFFSET MSG5      ; ＣＨ－１ ＯＰＥＮメッセージ
            CALL    MESSAGE
            MOV     DX,OFFSET CHOP      ; CH OPEN I/Oﾊﾟﾗﾒｰﾀｱﾄﾞﾚｽ
            CALL    COMMAND
            JNB     CH1OP_OK
            JMP     DOS                 ; ｴﾗｰ時 MS-DOSへ戻る
                                        ;
CH1OP_OK:
            MOV     DX,OFFSET OK_MSG    ; CH-1 OPEN OKﾒｯｾｰｼﾞ 表示
            CALL    MESSAGE
                                        ;
; ------------------------------------------
;           ＣＨ文字型 ＲＥＡＤ登録
; ------------------------------------------
                                        ;
RXLP:
            MOV     DX,OFFSET RX_ENT_MSG ; 文字型登録受信ﾒｯｾｰｼﾞ 表示
            CALL    MESSAGE
            MOV     DX,OFFSET MEWRD     ; READ用I/Oﾊﾟﾗﾒｰﾀ ｱﾄﾞﾚｽ
            CALL    COMMAND
            JNB     ENT_OK              ; 登録OK===>ENT_OK へ
            JMP     DOS                 ; ｴﾗｰ時 MS-DOS へ戻る
                                        ;
ENT_OK:
            MOV     DX,OFFSET OK_MSG    ; 登録OKﾒｯｾｰｼﾞ 表示
            CALL    MESSAGE
            MOV     AX,CS:[RXENTNO]
            MOV     CS:[ENTRYNO],AX     ; 登録受信NO.保管
            MOV     CS:[RXSENSE],0      ; 受信ｾﾝｽｶｳﾝﾀｰ0ｸﾘｱ
                                        ;
; ------------------------------------------
;         ＣＨ文字型受信センス処理
; ------------------------------------------
                                        ;
SENSECK:                                ;
            INC     CS:[RXSENSE]        ; 受信ｾﾝｽｶｳﾝﾀｰ+1
            MOV     DX,OFFSET RX_MSG    ; 文字型ﾃﾞｰﾀ受信ｾﾝｽﾒｯｾｰｼﾞ 表示
            CALL    MESSAGE
            MOV     AL,CS:[RXSENSE]     ; AL=ｾﾝｽｶｳﾝﾀｰ
            CALL    BIN_ASC             ; BIN-->ASCII 変換
            CALL    BIN_DSP             ; ｾﾝｽ回数表示
            MOV     DX,OFFSET RX_MSG2
            CALL    MESSAGE
                                        ;
            CMP     WORD PTR CS:[ENDF],0 ; 受信完了?
            JNE     RX_ED               ; YES===.RX_ED へ
                                        ; ...NO
            MOV     DX,OFFSET RX_BD_MSG ; 受信ｾﾝｽ出来ずﾒｯｾｰｼﾞ 表示
            CALL    MESSAGE
            CMP     CS:[RXSENSE],10     ; 10回ｾﾝｽ完了?
            JB      SENSECK             ; NO ===>SENSECK へ
                                        ; ...YES
; ------------------------------------------
;             登録受信解除処理
; ------------------------------------------
                                        ;
            MOV     DX,OFFSET RX_KAI_MSG ; 登録解除ﾒｯｾｰｼﾞ 表示
            CALL    MESSAGE
            MOV     DX,OFFSET ENTKAI    ; 登録解除
            CALL    COMMAND             ; ﾘﾝｸｿﾌﾄ呼出
            JNB     ENTKAI_OK           ; 解除OK===>ENTKAI_OK へ
            JMP     DOS                 ; ｴﾗｰ時 MS-DOSへ戻る
ENTKAI_OK:
            MOV     DX,OFFSET OK_MSG    ; 登録解除OKﾒｯｾｰｼﾞ 表示
            CALL    MESSAGE
            MOV     DX,OFFSET CRLF      ; 改行復帰
            CALL    MESSAGE
            JMP     RXLP                ; ===>受信登録へ
                                        ;
; ------------------------------------------
;               受信完了処理
; ------------------------------------------
RX_ED:                                  ;
```

```
                CMP     WORD PTR CS:[ENDF],1  ; 正常終了?
                JE      RD_OK                 ; YES===>RD_OK へ
                                              ; ...NO
                MOV     AX,CS:[ENDF]
                CALL    CMD_2                 ; ｴﾗｰｺｰﾄﾞ表示
                MOV     DX,OFFSET CRLF        ; 改行復帰
                CALL    MESSAGE
                JMP     RXLP                  ; ===>受信登録へ
                                              ;
RD_OK:                                        ; == 受信センス時 ==
                MOV     DX,OFFSET OK_MSG      ; ＯＫﾒｯｾｰｼﾞ表示
                CALL    MESSAGE
                                              ;
                CALL    RXDATA_DSP            ; 受信内容表示
                                              ;
                MOV     CX,CS:[RXSIZE]
                LEA     SI,JBUF
                LEA     DI,SBUF
RD_LP:
                MOV     AL,CS:[SI]            ; AL=受信ﾃﾞｰﾀ
                ADD     AL,20H                ; 受信ﾃﾞｰﾀ+20H
                MOV     CS:[DI],AL            ; <==送信ﾊﾞｯﾌｧへ転送
                INC     SI
                INC     DI
                LOOP    RD_LP
                                              ;
                MOV     BYTE PTR CS:[DI],CR
                MOV     BYTE PTR CS:[DI+1],LF
                MOV     BYTE PTR CS:[DI+2],'$'
                                              ;
;  ---------------------------------------------
;               ＣＨ文字型 ＷＲＩＴＥ
;  ---------------------------------------------
                                              ;
                MOV     AX,CS:[RXSIZE]
                MOV     CS:[TXSIZE],AX        ; 送信ﾃﾞｰﾀ･ｻｲｽﾞ ｾｯﾄ
                MOV     AL,CS:[RXNO]
                MOV     CS:[TXNO],AL          ; 送信先No.ｾｯﾄ
                MOV     DX,OFFSET TX_DATA_MSG ; 発行ﾃﾞｰﾀ表示
                CALL    MESSAGE
                MOV     DX,OFFSET SBUF
                CALL    MESSAGE
                                              ;
                MOV     DX,OFFSET TX_MSG      ; 文字ﾃﾞｰﾀ発行ﾒｯｾｰｼﾞ表示
                CALL    MESSAGE
                MOV     DX,OFFSET MEWWT       ; WRITEI/Oﾊﾟﾗﾒｰﾀ ｱﾄﾞﾚｽ
                CALL    COMMAND
                JB      TX_ED                 ; ｴﾗｰ時 再送
                                              ;
                MOV     DX,OFFSET OK_MSG      ; ＯＫﾒｯｾｰｼﾞ表示
                CALL    MESSAGE
TX_ED:
                MOV     DX,OFFSET CRLF        ; 改行復帰
                CALL    MESSAGE
                JMP     RXLP
                                              ;
;  =============================================
;  =          ＭＳ－ＤＯＳへ戻る処理          =
;  =============================================
                                              ;
DOS:
                MOV     AH,4CH
                INT     21H
                                              ;
                                              ;
;  =============================================
;  =          メッセージの画面表示処理          =
;  =============================================
                                              ;
MESSAGE:
                MOV     AH,9                  ; DXｱﾄﾞﾚｽから'$'迄を画面表示
                INT     21H
                RET
                                              ;
                                              ;
;  =============================================
```

```
; =          リンクソフト呼出処理              =
; ============================================
                                        ;
COMMAND:
            INT     60H                 ; ｼｽﾃﾑｺｰﾙ
            CMP     AX,0                ; 正常終了?
            JE      CMD_ED              ; YES===>CMD_ED へ
                                        ;
            CMP     AX,133H             ; 受信ｾﾝｽ出来ず?
            JE      CMD_ER              ; YES===>CMD_ER へ
                                        ; ...NO
CMD_2:
            PUSH    AX                  ; ｴﾗｰｺｰﾄﾞ待避
            CALL    BI2ASC              ; ｴﾗｰｺｰﾄﾞ ASCII変換
            MOV     DX,OFFSET MSG3      ; ｴﾗｰﾒｯｾｰｼﾞ画面表示
            CALL    MESSAGE
            POP     AX                  ; ｴﾗｰｺｰﾄﾞ復帰
CMD_ER:
            STC
CMD_ED:
            RET
                                        ;
                                        ;
; ============================================
; =       エラーコードのＡＳＣＩＩ変換       =
; ============================================
                                        ;
BI2ASC:
            MOV     BL,AH
            MOV     AH,AL
            MOV     BH,BL
            AND     AL,0FH
            AND     BL,0FH
            MOV     CL,4
            SHR     AH,CL
            SHR     BH,CL
            ADD     BX,3030H
            ADD     AX,3030H
            CMP     AL,3AH
            JB      B0
            ADD     AL,7
B0:
            CMP     AH,3AH
            JB      B1
            ADD     AH,7
B1:
            CMP     BL,3AH
            JB      B2
            ADD     BL,7
B2:
            CMP     BH,3AH
            JB      B3
            ADD     BH,7
B3:
            MOV     [MSG3_1+0],BH
            MOV     [MSG3_1+1],BL
            MOV     [MSG3_1+2],AH
            MOV     [MSG3_1+3],AL
            RET
                                        ;
                                        ;
; ============================================
; =        ＢＩＮ==>ＡＳＣＩＩ変換         =
; ============================================
                                        ;
BIN_ASC:                                ; IN...AL=BINｺｰﾄﾞ
            PUSH    SI                  ; OUT..AX=ASCIIｺｰﾄﾞ
            PUSH    BX
            PUSH    DX
                                        ;
            MOV     BL,AL
            AND     AL,0F0H             ; 上位４ビット
            SHR     AL,1
            SHR     AL,1
            SHR     AL,1
            SHR     AL,1
```

```
                LEA     SI,BINTOASC_TB          ; 変換テーブル
                MOV     AH,0
                ADD     SI,AX
                MOV     DH,BYTE PTR CS:[SI]
                                                ;
                MOV     AL,BL
                AND     AL,0FH                  ; 下位４ビット
                LEA     SI,BINTOASC_TB          ; 変換テーブル
                MOV     AH,0
                ADD     SI,AX
                MOV     AL,BYTE PTR CS:[SI]
                MOV     AH,DH
                XCHG    AL,AH
                                                ;
                POP     DX
                POP     BX
                POP     SI
                RET
                                                ;
                                                ;
;  ---------------------------------------------
;         ＨＥＸ→ＡＳＣＩＩ　変換テーブル
;  ---------------------------------------------
                                                ;
BINTOASC_TB     DB      "0123456789ABCDEF"
                                                ;
                                                ;
;  =============================================
;  =                受信データ表示                =
;  =============================================
                                                ;
RXDATA_DSP:
                MOV     DX,OFFSET DSP_RXSIZE    ; 受信ｻｲｽﾞ表示
                CALL    MESSAGE
                MOV     BX,CS:[RXSIZE]
                LEA     DI,JBUF
                MOV     BYTE PTR CS:[DI+BX],CR
                MOV     BYTE PTR CS:[DI+BX+1],LF
                MOV     BYTE PTR CS:[DI+BX+2],'$'
                MOV     AL,BH
                CALL    BIN_ASC
                CALL    BIN_DSP
                MOV     AL,BL
                CALL    BIN_ASC
                CALL    BIN_DSP
                MOV     DX,OFFSET CRLF
                CALL    MESSAGE
                                                ;
                MOV     DX,OFFSET DSP_RXNO      ; 送信元No. 表示
                CALL    MESSAGE
                MOV     AL,CS:[RXNO]
                CALL    BIN_ASC
                CALL    BIN_DSP
                MOV     DX,OFFSET CRLF
                CALL    MESSAGE
                                                ;
                MOV     DX,OFFSET DSP_DATA      ; 受信データ表示
                CALL    MESSAGE
                MOV     DX,OFFSET JBUF
                CALL    MESSAGE
                                                ;
                RET
                                                ;
                                                ;
BIN_DSP:                                        ; IN...AX=表示文字ｺｰﾄﾞ
                PUSH    AX
                MOV     DL,AL
                MOV     AH,2
                INT     21H
                POP     AX
                MOV     DL,AH
                MOV     AH,2
                INT     21H
                RET
                                                ;
                                                ;
```

```
;  ============================================
;  =            I／Oパラメータ                =
;  ============================================
                                        ; = 初期設定 =
INIT        DW      1000H               ; 初期設定ｺﾏﾝﾄﾞｺｰﾄﾞ
            DB      0                   ; ｱﾌﾟﾘｹｰｼｮﾝ･ｿﾌﾄﾓｰﾄﾞ
            DB      1                   ; 使用ﾎﾞｰﾄﾞ枚数
                                        ;
                                        ; = ﾎﾞｰﾄﾞ登録･起動 =
BOARD       DW      1100H               ; ﾎﾞｰﾄﾞ登録･起動ｺﾏﾝﾄﾞｺｰﾄﾞ
            DB      0                   ; ﾎﾞｰﾄﾞ登録No.
            DB      27                  ; 使用ｾｸﾞﾒﾝﾄｱﾄﾞﾚｽNo.27(D0000H)
            DB      0                   ; 割り込みNo.
            DB      0
                                        ;
                                        ; = ＣＨ文字型ＷＲＩＴＥ =
MEWWT       DW      1501H               ; ｺﾏﾝﾄﾞｺｰﾄﾞ
            DB      0                   ; 登録ﾎﾞｰﾄﾞNO.
            DB      0
            DW      OFFSET SBUF         ; 送信ﾊﾞｯﾌｧOFFSET
            DW      SEG SBUF            ; 送信ﾊﾞｯﾌｧSEGMENT
            DW      0
TXSIZE      DW      0                   ; 送信ﾃﾞｰﾀ･ｻｲｽﾞ
            DB      0                   ;
            DB      0,0,0
            DB      0                   ; 制御ｺｰﾄﾞ
            DB      1                   ; 送信先CH
            DB      0                   ; 送信先ID
            DB      1                   ; 送信元CH
            DB      0                   ; 送信元ID
            DB      0
            DB      0                   ; Depth
TXNO        DB      0                     ; 送信先NO.
            DB      14 DUP(0)
            DW      4 DUP(0)
            DW      0                   ;
                                        ;
                                        ;
                                        ; = ＣＨ文字型ＲＥＡＤ =
MEWRD       DW      1404H               ; ｺﾏﾝﾄﾞｺｰﾄﾞ(登録受信ﾀｲﾌﾟ)
            DB      0                   ; 登録ﾎﾞｰﾄﾞNO.
            DB      0
            DW      OFFSET JBUF         ; 受信ﾊﾞｯﾌｧOFFSET
MEWRD2      DW      SEG JBUF
            DW      2048                ; 受信ﾊﾞｯﾌｧｻｲｽﾞ
RXSIZE      DW      0                   ; 受信ﾃﾞｰﾀｻｲｽﾞ用
            DB      0                   ;
            DB      0,0,0
            DB      0                   ; 制御ｺｰﾄﾞ
            DB      0                   ; 送信元CH
            DB      0                   ; 送信元ID
            DB      1                   ; 受信CH
            DB      0                   ; 受信ID
            DB      0
            DB      0                   ; Depth
RXNO        DB      0                   ; 送信元NO.
            DB      14 DUP(0)           ; 受信経路用
            DW      3 DUP(0)
RXENTNO     DW      0                   ; 登録受信NO.
ENDF        DW      0                   ; 完了ﾌﾗｸﾞ
                                        ;
                                        ; = ＣＨ ＯＰＥＮ =
CHOP        DW      1601H               ; ｺﾏﾝﾄﾞｺｰﾄﾞ
            DB      0                   ; 登録ﾎﾞｰﾄﾞNO.
            DB      1                   ; OPEN CH NO.
            DB      8 DUP(0)            ; CH状態格納用
                                        ;
                                        ; = 登録受信解除 =
ENTKAI      DW      1460H               ; ｺﾏﾝﾄﾞｺｰﾄﾞ
            DB      0                   ; 登録ﾎﾞｰﾄﾞNO.
            DB      0                   ; 0固定
            DW      0                   ; 登録NO.
                                        ;
                                        ;
;  ============================================
;  =        メッセージ ／ バッファ領域        =
;  ============================================
```

パソコン機種別変更箇所
（左記はＰＣ９８シリーズの場合）

```
                                        ;
MSG1            DB      '初期設定===>$'

MSG3            DB      'ｴﾗｰ.....ｺｰﾄﾞ='
MSG3_1          DB      0,0,0,0
                DB      CR,LF,'$'

MSG4            DB      '使用ボード登録 & 起動===>$'

MSG5            DB      'ＣＨ１　ＯＰＥＮ===>$'

TX_DATA_MSG     DB      '発行データ＝$'
TX_MSG          DB      '文字データ発行===>$'

RX_ENT_MSG      DB      'ＣＨ（文字型）登録受信===>$'

RX_MSG          DB      '文字データ受信センス（$'
RX_MSG2         DB      '回目）===>$'
RX_BD_MSG       DB      '受信センス出来ず！！',CR,LF,'$'
RX_KAI_MSG      DB      '登録受信解除===>$'

DSP_RXSIZE      DB      '受信ﾃﾞｰﾀｻｲｽﾞ(Hex)=$'
DSP_RXNO        DB      '送信元No.          =$'
DSP_DATA        DB      '受信データ         =$'

OK_MSG          DB      'ＯＫ！',CR,LF,'$'
CRLF            DB      CR,LF,'$'

SBUF            DB      2048 DUP(0)             ; 送信ﾊﾞｯﾌｧ
JBUF            DB      2048 DUP(0)             ; 受信ﾊﾞｯﾌｧ

RXSENSE         DB      0                       ; 受信ｾﾝｽ用ｶｳﾝﾀｰ
ENTRYNO         DW      0                       ; 登録受信NO.保管用

STACK           DW      256 DUP(0)              ; ｽﾀｯｸ領域
STPT            DB      0


                CODE    ENDS
                END     START
```

パソコン機種別変更箇所

ＰＣ／ＡＴ互換機

```
BOARD           DW      1100H                   ; ﾎﾞｰﾄﾞ登録･起動ｺﾏﾝﾄﾞｺｰﾄﾞ
                DB      0                       ; ﾎﾞｰﾄﾞ登録No.
                DB      8                       ; 使用ｾｸﾞﾒﾝﾄｱﾄﾞﾚｽNo.
                DB      10                      ; 割り込みNo.
                DB      0
```

ＦＭＲシリーズ

```
BOARD           DW      1100H                   ; ﾎﾞｰﾄﾞ登録･起動ｺﾏﾝﾄﾞｺｰﾄﾞ
                DB      0                       ; ﾎﾞｰﾄﾞ登録No.
                DB      7                       ; 基板I/OﾎﾟｰﾄｱﾄﾞﾚｽNo.
                DW      7800H                   ; ｺﾝﾄﾛｰﾙ･ｽﾃｰﾀｽI/Oﾎﾟｰﾄｱﾄﾞﾚｽ
                DB      5                       ; 割り込みNo.
                DB      0
```

Ｃプログラム（ソフトウェア割り込みint60）

```
/*****************************************************************/
/*          ＣＨ文字型 Ｗ／Ｒ 登録受信及び解除タイプ             */
/*                          (CHCR3.C)                            */
/*****************************************************************/
/*                              ＰＣ９８シリーズ                 */
/*      スレーブ処理:                                            */
/*      マスターからのデータを受信したら応答を返します。         */
/*      （受信各文字に＋２０ｈした文字を返却します。）           */
/*      データの受信は、登録タイムアウト無しタイプで行い         */
/*      ます。                                                   */
/*      受信完了センスで１０回連続センス出来なかった場合、       */
/*      一旦登録を解除します。                                   */
/*      本プログラムを起動するパソコンＩ／Ｆボードは             */
/*      ６４番機として下さい。                                   */
/*      マスター局側をCHCRタイプで起動させて下さい。             */
/*                                                               */
/*       使用コマンド：WRITE=1501H(ﾀｲﾑｱｳﾄ付き)                   */
/*                     READ =1404H(登録受信ﾀｲﾑｱｳﾄ無しﾀｲﾌﾟ)        */
/*****************************************************************/

#include  <dos.h>
#include  <stdio.h>
#include  <stdlib.h>

#define ECODE "D"           /* DT                  */
#define START_N 0           /* DT 0 から           */
#defineEND_N  10            /* DT 10 までﾘｰﾄﾞ       */

static union REGS inregs;
static union REGS outregs;
struct SREGS segregs;
static char far *p1;
static unsigned char dat[48];
unsigned char sbuf[2048];           /* 送信ﾊﾞｯﾌｧ  */
unsigned char jbuf[2048];           /* 受信ﾊﾞｯﾌｧ  */

void main( void )
{
       int    i;
       int    err;
       long   lwork;
       char   cwork[5];
       char   c_dmy1[4];
       char   *dmy,rxno;
       int    datano,rxsize,lped,rxsense,endf,entryno;

/* -------------------------*/
/* ------ 初期設定処理 ------*/
/* -------------------------*/

       dat[0] = 0x00;       /* 初期設定ｺﾏﾝﾄﾞ･ｺｰﾄﾞ=1000H  */
       dat[1] = 0x10;
       dat[2] = 0;          /* ｱﾌﾟﾘｹｰｼｮﾝ･ｿﾌﾄﾓｰﾄﾞ=0       */
       dat[3] = 1;          /* 使用ボード枚数=1          */

       printf("初期設定===>");

       if((err = mew_send()) != 0)
       {
              printf("ｴﾗｰ.....ｺｰﾄﾞ= %x¥n", err);
              exit(-1);
       }
       printf("ＯＫ！¥n");

/* ------------------------------------*/
/* ------ 使用ボード登録 及び起動 ------*/
/* ------------------------------------*/
```

```
        dat[0] = 0x00;          /* 使用ボード登録及び起動    */
        dat[1] = 0x11;          /* ｺﾏﾝﾄﾞｺｰﾄﾞ=1100H           */
        dat[2] = 0x00;          /* 登録ボードNo.=0           */
        dat[3] = 27;            /* 使用ｾｸﾞﾒﾝﾄｱﾄﾞﾚｽNo.=27    */
        dat[4] = 0;             /* 割り込みNo.=0             */

        printf("使用ボード登録 & 起動===>");

        if((err = mew_send()) != 0)
        {
                printf("ｴﾗｰ.....ｺｰﾄﾞ= %x¥n", err);
                exit(-1);
        }
        printf("ＯＫ！¥n");

/* ----------------------------*/
/* ------ ＣＨ１ ＯＰＥＮ ------*/
/* ----------------------------*/

        dat[0] = 0x01;          /* CH OPEN ｺﾏﾝﾄﾞｺｰﾄﾞ=1601H   */
        dat[1] = 0x16;
        dat[2] = 0x00;          /* 登録ﾎﾞｰﾄﾞNo.=0     */
        dat[3] = 0x01;          /* OPEN CH No. =1     */

        printf("ＣＨ１ ＯＰＥＮ===>");

        if((err = mew_send()) != 0)
        {
                printf("ｴﾗｰ...ｺｰﾄﾞ= %x¥n",err);
                exit(-1);
        }
        printf("ＯＫ！¥n");

/* ------------------------------------*/
/* ------ ＣＨ文字型 ＲＥＡＤ登録 ------*/
/* ------------------------------------*/

RXLP:
        lped = 0;
        memset( jbuf, 0, sizeof( jbuf ) );

        do
        {
                p1 = (char far *)&jbuf[0];

                dat[0] = 0x04;          /* CH文字型READ       */
                dat[1] = 0x14;          /* ｺﾏﾝﾄﾞｺｰﾄﾞ=1404H    */
                dat[2] = 0x00;          /* 登録ボードno.=0    */
                dat[3] = 0x00;          /* 受信ch no.  */
                *((unsigned *)&dat[4]) = FP_OFF(p1);        /* 受信ﾊﾞｯﾌｧｵﾌｾｯﾄ     */
                *((unsigned *)&dat[6]) = FP_SEG(p1);         /* 受信ﾊﾞｯﾌｧｾｸﾞﾒﾝﾄ   */
                *((unsigned *)&dat[8]) = 2048;              /* 受信ﾊﾞｯﾌｧｻｲｽﾞ       */
                dat[16] = 0;                    /* 受信制御ｺｰﾄﾞ        */
                dat[19] = 1;                    /* 受信CH No.=1        */

                printf("ＣＨ（文字型）登録受信===>");

                if((err = mew_send()) != 0)
                {
                        printf("ｴﾗｰ.....ｺｰﾄﾞ= %x¥n", err);
                        exit(-1);
                }
                else
                {
                        printf("ＯＫ！¥n");
                        entryno = *((unsigned *)&dat[44]);          /* 登録No.     */
                }

                rxsense = 0;

                do
                {
                        ++rxsense;
                        printf("文字データ受信センス(%d回目)===>",rxsense);
                        endf = *((unsigned *)&dat[46]);
                        if(endf == 0)                           /* 受信完了ｾﾝｽ*/
```

パソコン機種別変更箇所
（左記はＰＣ９８シリーズの場合）

```
                {
                        printf("受信センス出来ず！！¥n");
                        if(rxsense >= 10)
                        {
                                dat[0] = 0x60;          /* 登録解除    */
                                dat[1] = 0x14;          /* ｺﾏﾝﾄﾞ=1460H */
                                dat[2] = 0x00;          /* 登録ﾎﾞｰﾄﾞ No.      */
                                dat[3] = 0x00;          /* 0固定       */
                                    *((unsigned *)&dat[4])=entryno;
                                printf("登録受信解除===>");
                                if((err = mew_send()) != 0)
                                {
                                        printf("ｴﾗｰ.....ｺｰﾄﾞ= %x¥n", err);
                                        exit(-1);
                                }
                                else
                                {
                                        printf("ＯＫ！¥n¥n");
                                        lped = 1;
                                }
                        }
                }
                  else
                {
                        if(endf == 1)
                        {
                                printf("ＯＫ！¥n");
                                rxsize = dat[10] + dat[11] * 0x100;
                                printf("受信ﾃﾞｰﾀｻｲｽﾞ=%d¥n",rxsize);
                                rxno = dat[23];
                                printf("送信元No.    =%d¥n",rxno);
                                printf("受信データ   =%s¥n",jbuf);
                                memset( sbuf, 0, sizeof( sbuf ) );
                                for(i = 0 ; i < rxsize ; i++)
                                {
                                        sbuf[i] = jbuf[i] + 0x20;
                                }
                                lped = 2;
                        }
                        else
                        {
                                printf("ｴﾗｰ.....ｺｰﾄﾞ= %x¥n¥n",endf);
                                lped = 1;
                        }
                }
        }
        while(lped == 0);
        --lped;
}
while(lped == 0);


/* ---------------------------------*/
/* ------ ＣＨ文字型 ＷＲＩＴ ------*/
/* ---------------------------------*/

        printf("発行データ=%s¥n",sbuf);
        p1 = (char far *)&sbuf[0];

        dat[0] = 0x01;          /* CH文字型WRITE     */
        dat[1] = 0x15;          /* ｺﾏﾝﾄﾞｺｰﾄﾞ=1501H   */
        dat[2] = 0x00;          /* 登録ボードno.=0   */
        dat[3] = 0x00;
        *((unsigned *)&dat[4]) = FP_OFF(p1);        /* 送信ﾊﾞｯﾌｧｵﾌｾｯﾄ    */
        *((unsigned *)&dat[6]) = FP_SEG(p1);        /* 送信ﾊﾞｯﾌｧｾｸﾞﾒﾝﾄ   */
        *((unsigned *)&dat[10]) = rxsize;           /* 送信ﾃﾞｰﾀｻｲｽﾞ       */
        dat[16] = 0;                /* 書き込み制御ｺｰﾄﾞ  */
        dat[17] = 1;                /* 送信先CH=1        */
        dat[18] = 0;                /* 送信先ID=0        */
        dat[19] = 1;                /* 送信元CH=1        */
        dat[20] = 0;                /* 送信元ID=0        */
        dat[22] = 0;                /* Depth=0(階層なし)*/
        dat[23] = rxno;             /* 送信先No.(PC局番)*/

        printf("文字データ発行===>");
```

```
        if((err = mew_send()) != 0)
        {
                printf("ｴﾗｰ.....ｺｰﾄﾞ= %x ¥n", err);
                goto RXLP;
        }

        printf("ＯＫ！¥n¥n");
        goto RXLP;
}


/* ---------------------------*/
/* ------ ﾘﾝｸｿﾌﾄ呼び出し------*/
/* ---------------------------*/
int mew_send(void)
{

        p1 = (char far *)&dat[0];

        inregs.x.dx = FP_OFF(p1);
        segregs.ds = FP_SEG(p1);

        int86x( 0x60, &inregs, &outregs, &segregs);
        return(outregs.x.ax);
}
```

ＰＣ／ＡＴ互換機

```
        dat[0] = 0x00;          /* 使用ボード登録及び起動    */
        dat[1] = 0x11;          /* ｺﾏﾝﾄﾞｺｰﾄﾞ=1100H           */
        dat[2] = 0x00;          /* 登録ボードNo.=0           */
        dat[3] = 8;             /* 使用ｾｸﾞﾒﾝﾄｱﾄﾞﾚｽNo.=8      */
        dat[4] = 10;            /* 割り込みNo.=10     */
```

ＦＭＲシリーズ

```
        dat[0] = 0x00;          /* 使用ボード登録及び起動    */
        dat[1] = 0x11;          /* ｺﾏﾝﾄﾞｺｰﾄﾞ=1100H           */
        dat[2] = 0x00;          /* 登録ボードNo.=0           */
        dat[3] = 7;             /* 基板I/OﾎﾟｰﾄｱﾄﾞﾚｽNo.=7     */
        dat[4] = 0x00;          /* ｽﾃｰﾀｽ･ｺﾝﾄﾛｰﾙI/Oﾎﾟｰﾄｱﾄﾞﾚｽ  */
        dat[5] = 0x78;          /* =7800H                    */
        dat[6] = 5;             /* 割り込みNo.=5             */
```

3－8 コントロール／ステータスレジスタのアクセス

3-8-1 コントロールレジスタWRITE

コントロールレジスタ書き込み（コマンドコード&H1200）

機 能

指定バッファの内容でコントロールレジスタへの書き込みを行います。

解 説

・機能コマンドを実行するには、各種プログラミング言語でI/Oパラメータのセグメントアドレスとオフセットアドレスをds:dxレジスタにセットし、割り込みを実行するユーザープログラムを作成してください。
・コマンド実行後の結果（リターン値）は、axレジスタに格納されます。
　0：正常終了時
　その他：エラーコード（「3－9」参照）

文 例

●MS-DOSファンクションコール（int21h）

```
;----- コントロールレジスタWRITE ----
CRGW:
        mov     bx,[FILHDL]
        mov     ah,44h
        mov     al,03h
        mov     dx,offset CRGW
        int     21h
;------- I/Oﾊﾟﾗﾒｰﾀ ------
CRGW    dw      1200h
        db      0,0
        dw      632,1,offset WBUF,seg WBUF,18 dup(?)
WBUF    db      20
```

●ソフトウェア割り込み（int60h）

```
;----- コントロールレジスタWRITE ----
CRGW:
        mov     dx,offset CRGW
        int     60h
;------- I/Oﾊﾟﾗﾒｰﾀ ------
CRGW    dw      1200h
        db      0,0
        dw      632,1,offset WBUF,seg WBUF,18 dup(?)
WBUF    db      20
```

I/Oパラメータ

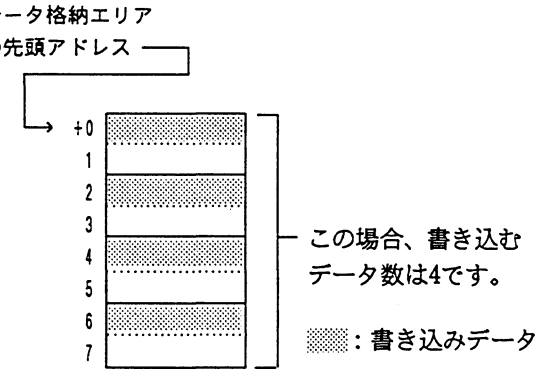
△：設定する［渡す値］
▼：実行後に格納される［得る値］

dsレジスタ ←I/Oパラメータの先頭セグメントアドレス
dxレジスタ ←I/Oパラメータの先頭オフセットアドレス

0		△［コマンドコード］&H1200
2		△［登録ボードNo.］0～3（使用するリンクボードの登録ボードNo.）
		△ 設定値：00固定で使用してください。
4		△［書き込み開始レジスタNo.］コントロールレジスタの書き込み開始アドレス（10進）を指定。
6		△［書き込みデータ数］書き込むデータ数を指定します。
8		△［書き込みデータ格納バッファアドレス（オフセット）］
A		△［書き込みデータ格納バッファアドレス（セグメント）］
C		
E		
10		
12		
14		
16		
18		
1A		
1C		
1E		
20		
22		
24		
26		
28		
2A		
2C		
2E		

注　意

・書き込みデータの格納方法は、各1ワードエリアのLOWバイト側にセットします。



・コントロールレジスタ以外への書き込みはできません。
・コントロールレジスタ範囲を越えて書き込みは行うと、エラー106Hが出力されます。この場合、コントロールレジスタ内への書き込みは行われます。

3-8-2 コントロール／ステータスレジスタREAD

ステータスレジスタ・コントロールレジスタ読み出し（コマンドコード&H1300）

機　能

指定されたデータ格納バッファに、ステータスレジスタおよびコントロールレジスタを読み出します。

解　説

・機能コマンドを実行するには、各種プログラミング言語でI/Oパラメータのセグメントアドレスとオフセットアドレスをds:dxレジスタにセットし、割り込みを実行するユーザープログラムを作成してください。
・コマンド実行後の結果（リターン値）は、axレジスタに格納されます。
　0：正常終了時
　その他：エラーコード（「３－９」参照）

文　例

●MS-DOSファンクションコール（int21h）

```
;--- コントロール／ステータスレジスタREAD ----
CSRGR:
        mov     bx,[FILHDL]
        mov     ah,44h
        mov     al,03h
        mov     dx,offset CSRGR
        int     21h
;------- I/Oﾊﾟﾗﾒｰﾀ ------
CSRGR   dw      1300h
        db      0,0
        dw      616,32,ofset RBUF,seg RBUF,18 dup(?)
RBUF    db      32 dub(?)
```

●ソフトウェア割り込み（int40h以降）

```
;--- コントロール／ステータスレジスタREAD ---
CSRGR:
        mov     dx,offset CSRGR
        int     40h
;------- I/Oﾊﾟﾗﾒｰﾀ ------
CSRGR   dw      1300h
        db      0,0
        dw      616,32,ofset RBUF,seg RBUF,18 dup(?)
```

I/Oパラメータ

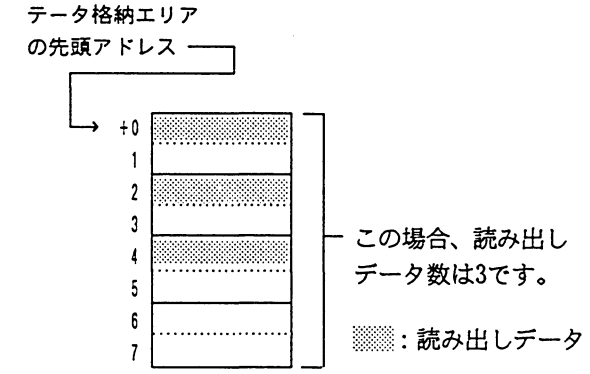
△：設定する［渡す値］
▼：実行後に格納される［得る値］

dsﾚｼﾞｽﾀ ←I/Oパラメータの先頭セグメントアドレス
dxﾚｼﾞｽﾀ ←I/Oパラメータの先頭オフセットアドレス

0	△［コマンドコード］設定値：&H1300
2	△［登録ボードNo.］格納値：0～3　使用するリンクボードの登録ボードNo.を設定します。
	△ 設定値：00固定で使用してください。
4	△［読み出し開始レジスタNo.］コントロールレジスタの読み出し開始アドレスを指定します。
6	△［読み出しデータ数］読み出すデータ数をバイト数で指定します。
8	△［読み出しデータ格納バッファアドレス（オフセット）］
A	△［読み出しデータ格納バッファアドレス（セグメント）］
C	
E	
10	
12	
14	
16	
18	
1A	
1C	
1E	
20	
22	
24	
26	
28	
2A	
2C	
2E	

注　意

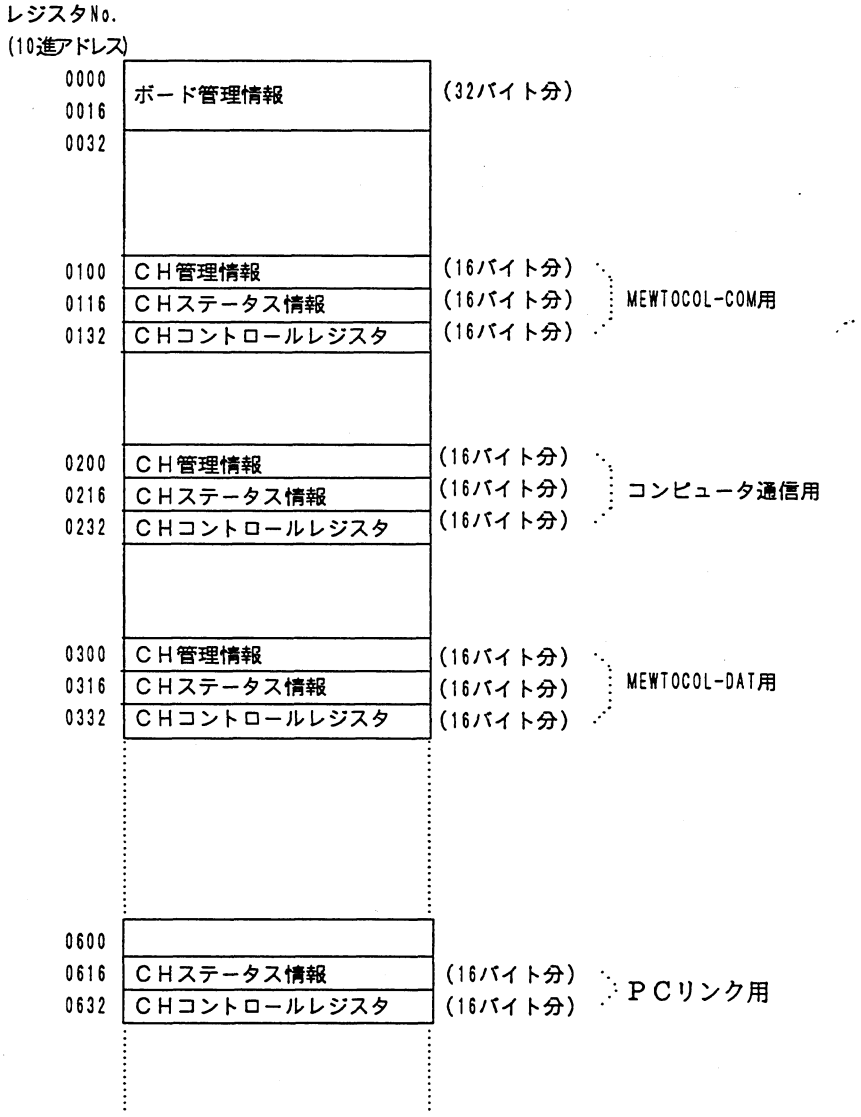
・読み出したデータは、各1ワードエリアのLOWバイト側にセットされます。



・サポート外レジスタNo.を越えての読み出しはできません。

3-8-3 コントロール／ステータスレジスタの構成

ステータスレジスタ、コントロールレジスタの構成は以
下の通りです。



(1)ボード管理情報

レジスタNo. (10進アドレス)	7 6 5 4 3 2 1 0	
0000	割り込みNo.	
0001	セグメントNo.	
0002 0003	セグメント値 （メモリアドレス）	
0004	割り込み処理中フラグ	→ 1：処理中
0005		
0006	リンクボードNo.	
0007	ネットワーク加入局数	
0008	8 7 6 5 4 3 2 1	ネットワーク加入情報 ＯＮビット局と通信可能。 （自局No.ビット位置もＯＮします。）
0009	16 15 14 13 12 11 10 9	
0010	24 23 22 21 20 19 18 17	
0011	32 31 30 29 28 27 26 25	
0012	40 39 38 37 36 35 34 33	
0013	48 47 46 45 44 43 42 41	
0014	56 55 54 53 52 51 50 49	
0015	64 63 62 61 60 59 58 57	
0016	リンクボード動作状態	→ 0：動作正常　1：制御ボード部異常　2：通信系異常
0017	リンクボード通信状態	→ 0：通信正常　その他：通信異常
0018	ＰＣリンク実施局数	
0019	登録受信タイプ登録残数	
0020	ボード内現在受信数	
0021 0023 0024 0025 0026 0027 0028 0029 0030 0031		

(2)ＣＨ管理情報

レジスタNo. (10進アドレス)		
0n00	割り込み許可／禁止状態	→ 0：禁止　1：許可
0n01	割り込みアドレス登録済みフラグ	→ 0：未登録　1：登録済み
0n02 0n03	割り込み先アドレス （オフセット）	
0n04 0n05	割り込み先アドレス （セグメント）	
0n06	00	
0n07	00	
0n08 0n09 0n10 0n11 0n12 0n13 0n14 0n15	未使用	

n=1：MEWTOCOL-COM用
n=2：コンピュータ通信用
n=3：MEWTOCOL-DAT用
n=6：ＰＣリンク用

(3)ＣＨステータス情報

■コンピュータリンク、データ転送、コンピュータ間通信

レジスタNo.
(10進アドレス)

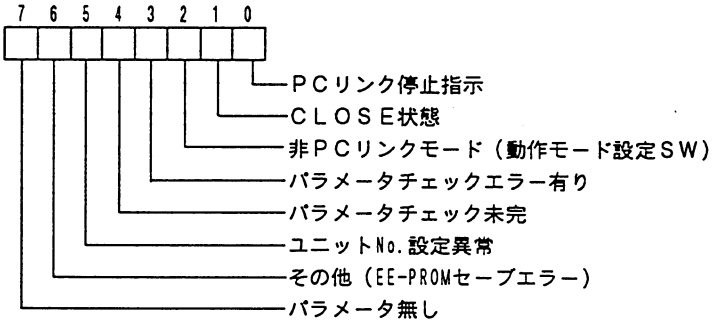
On16	受信回数
On17	00
On18	
On19	
On20	
On21	
On22	
On23	
On24	送信完了フラグ
On25	送信結果
On26	
On27	未使用
On28	
On29	
On30	
On31	

n=1：MEWTOCOL-COM用
n=2：コンピュータ通信用
n=3：MEWTOCOL-DAT用

■ＰＣリンク

レジスタNo.
(10進アドレス)

0616	受信完了フラグ	
0617	受信完了結果	
0618	8 7 6 5 4 3 2 1	┐
0619	16 15 14 13 12 11 10 9	┘ 受信UNIT No.
0620	8 7 6 5 4 3 2 1	┐
0621	16 15 14 13 12 11 10 9	┘ ＰＣリンク通信保証　ON：ＰＣリンク実施
0622	8 7 6 5 4 3 2 1	┐
0623	16 15 14 13 12 11 10 9	┘ ＰＣ動作モード　ON：RUN
0624		
0625		
0626		
0627		
0628		
0629	ＰＣリンク実施局数	
0630	ＰＣリンク動作状態	→ 0：停止　1：起動
0631	ＰＣリンク停止要因	→



(4)ＣＨコントロールレジスタ

■コンピュータリンク、データ転送、コンピュータ間通信

レジスタNo.
(10進アドレス)

On32	送信完了待ちタイムアウト値
On33	未使用
On34	
On35	
On36	
On37	
On38	
On39	
On40	
On41	
On42	
On43	未使用
On44	
On45	
On46	
On47	

→ 2～58秒の範囲で指定（秒単位）　初期値：10秒

n=1：MEWTOCOL-COM用
n=2：コンピュータ通信用
n=3：MEWTOCOL-DAT用

■ＰＣリンク

レジスタNo.
(10進アドレス)

0632	送信完了待ちタイムアウト値
0633	未使用
0634	
0635	
0636	
0637	
0638	
0639	
0640	
0641	
0642	
0643	
0644	未使用
0645	
0646	
0647	

→ 2～58秒の範囲で指定（秒単位）　初期値：10秒

3-8-4 コントロール／ステータスレジスタのアクセスのプログラム例

アセンブラプログラム（MS-DOSシステムコールint21）

```
; ************************************************************
; *       コントロール／ステータスレジスタＲＤ／ＷＴ         *
; *                    <ACTST.ASM>                           *
; ************************************************************
; *                  ＰＣ９８シリーズ                        *
; *                                                          *
; *   ｺﾝﾄﾛｰﾙﾚｼﾞｽﾀWRITE:                                       *
; *   ｺﾝﾋﾟｭｰﾀﾘﾝｸ用送受信ﾀｲﾑｱｳﾄ値を10秒から3秒に変更します。    *
; *                                                          *
; *   ｽﾃｰﾀｽﾚｼﾞｽﾀREAD:                                         *
; *   ﾘﾝｸﾎﾞｰﾄﾞNO. 及びﾈｯﾄﾜｰｸ加入局情報を読出します。           *
; *                                                          *
; *    使用コマンド：WRITE=1200H                             *
; *                  READ =1300H                             *
; ************************************************************
                                        ;
CODE    SEGMENT
        ASSUME CS:CODE,DS:CODE,ES:CODE,SS:CODE


CR              EQU     0DH             ; CRｺｰﾄﾞ
LF              EQU     0AH             ; LFｺｰﾄﾞ
                                        ;
                                        ;
START:
                MOV     AX,CS
                MOV     DS,AX
                MOV     ES,AX
                MOV     SS,AX
                MOV     AX,OFFSET STPT
                MOV     SP,AX
                                        ;
;  ---------------------------------------
;                 初 期 設 定
;  ---------------------------------------
INI:
                MOV     DX,OFFSET MSG1  ; 初期設定メッセージ表示
                CALL    MESSAGE
                MOV     DX,OFFSET INIT  ; 初期設定用I/Oﾊﾟﾗﾒｰﾀ ｱﾄﾞﾚｽ設定
                CALL    COMMAND         ; ﾘﾝｸｿﾌﾄ呼出
                JNB     INI_OK
                JMP     DOS             ; ｴﾗｰ時 MS-DOSへ戻る
                                        ;
INI_OK:
                MOV     DX,OFFSET OK_MSG ; 初期設定ＯＫﾒｯｾｰｼﾞ 表示
                CALL    MESSAGE
                                        ;
;  ---------------------------------------
;        ボ ー ド 登 録 ・ 起 動
;  ---------------------------------------
                                        ;
                MOV     DX,OFFSET MSG4  ; ﾎﾞｰﾄﾞ登録･起動メッセージ表示
                CALL    MESSAGE
                MOV     DX,OFFSET BOARD ; ﾎﾞｰﾄﾞ登録用I/Oﾊﾟﾗﾒｰﾀ ｱﾄﾞﾚｽ
                CALL    COMMAND         ; ﾘﾝｸｿﾌﾄ呼出
                JNB     BDST_OK
                JMP     DOS             ; ｴﾗｰ時 MS-DOSへ戻る
                                        ;
BDST_OK:
                MOV     DX,OFFSET OK_MSG ; ﾎﾞｰﾄﾞ登録&起動OKﾒｯｾｰｼﾞ 表示
                CALL    MESSAGE

;  ---------------------------------------
;        コントロールレジスタＷＲＩＴＥ
;  ---------------------------------------
                                        ;
                MOV     DX,OFFSET CTWT_MSG ; ｺﾝﾄﾛｰﾙﾚｼﾞｽﾀWRITEﾒｯｾｰｼﾞ 表示
                CALL    MESSAGE
                MOV     DX,OFFSET CTWT
```

```
                CALL    COMMAND                 ; ﾘﾝｸｿﾌﾄ呼出
                JNB     CTWT_OK
                JMP     DOS                     ; ｴﾗｰ時 MS-DOSへ戻る
                                                ;
CTWT_OK:
                MOV     DX,OFFSET  OK_MSG       ; WRITE OKﾒｯｾｰｼﾞ 表示
                CALL    MESSAGE
                                                ;
; ---------------------------------------------
;           ステータスレジスタＲＥＡＤ
; ---------------------------------------------
                                                ;
LP:
                MOV     DX,OFFSET  STRD_MSG     ; ｽﾃｰﾀｽﾚｼﾞｽﾀREADﾒｯｾｰｼﾞ 表示
                CALL    MESSAGE
                MOV     DX,OFFSET  STRD
                CALL    COMMAND                 ; ﾘﾝｸｿﾌﾄ呼出
                JNB     STRD_OK
                JMP     DOS                     ; ｴﾗｰ時 MS-DOSへ戻る
                                                ;
STRD_OK:
                MOV     DX,OFFSET  OK_MSG       ; READ OKﾒｯｾｰｼﾞ 表示
                CALL    MESSAGE
                                                ;
                CALL    RD_DATA_DSP             ; READﾃﾞｰﾀ表示
                JMP     LP
                                                ;
                                                ;
; ===============================================
; =           ＭＳ－ＤＯＳへ戻る処理           =
; ===============================================
                                                ;
DOS:
                MOV     AH,4CH
                INT     21H
                                                ;
                                                ;
; ===============================================
; =          メッセージの画面表示処理           =
; ===============================================
                                                ;
MESSAGE:
                MOV     AH,9                    ; DXｱﾄﾞﾚｽから'$'迄を画面表示
                INT     21H
                RET
                                                ;
                                                ;
; ===============================================
; =           リンクソフト呼出処理              =
; ===============================================
                                                ;
COMMAND:
                INT     60H                     ; ｼｽﾃﾑｺｰﾙ
                CMP     AX,0                    ; AX>0時ｴﾗｰ
                JE      CMD2                    ;
                                                ;
                CALL    BI2ASC                  ; ｴﾗｰｺｰﾄﾞ ASCII変換
                MOV     DX,OFFSET  MSG3         ; ｴﾗｰﾒｯｾｰｼﾞ 画面表示
                CALL    MESSAGE
                STC
CMD2:
                RET
                                                ;
                                                ;
; ===============================================
; =        エラーコードのＡＳＣＩＩ変換        =
; ===============================================
                                                ;
BI2ASC:
                MOV     BL,AH
                MOV     AH,AL
                MOV     BH,BL
                AND     AL,0FH
                AND     BL,0FH
                MOV     CL,4
                SHR     AH,CL
```

```
                SHR     BH,CL
                ADD     BX,3030H
                ADD     AX,3030H
                CMP     AL,3AH
                JB      B0
                ADD     AL,7
B0:
                CMP     AH,3AH
                JB      B1
                ADD     AH,7
B1:
                CMP     BL,3AH
                JB      B2
                ADD     BL,7
B2:
                CMP     BH,3AH
                JB      B3
                ADD     BH,7
B3:
                MOV     [MSG3_1+0],BH
                MOV     [MSG3_1+1],BL
                MOV     [MSG3_1+2],AH
                MOV     [MSG3_1+3],AL
                RET
                                        ;
                                        ;
; ========================================
; =       ＢＩＮ==>ＡＳＣＩＩ変換       =
; ========================================
                                        ;
BIN_ASC:                                ; IN...AL=BINｺｰﾄﾞ
                PUSH    SI              ; OUT..AX=ASCIIｺｰﾄﾞ
                PUSH    BX
                PUSH    DX
                                        ;
                MOV     BL,AL
                AND     AL,0F0H         ; 上位４ビット
                SHR     AL,1
                SHR     AL,1
                SHR     AL,1
                SHR     AL,1
                LEA     SI,BINTOASC_TB  ; 変換テーブル
                MOV     AH,0
                ADD     SI,AX
                MOV     DH,BYTE PTR CS:[SI]
                                        ;
                MOV     AL,BL
                AND     AL,0FH          ; 下位４ビット
                LEA     SI,BINTOASC_TB  ; 変換テーブル
                MOV     AH,0
                ADD     SI,AX
                MOV     AL,BYTE PTR CS:[SI]
                MOV     AH,DH
                XCHG    AL,AH
                                        ;
                POP     DX
                POP     BX
                POP     SI
                RET
                                        ;
                                        ;
; ----------------------------------------
;       ＨＥＸ→ＡＳＣＩＩ　変換テーブル
; ----------------------------------------
                                        ;
BINTOASC_TB     DB      "0123456789ABCDEF"
                                        ;
                                        ;
; ----------------------------------------
;          ステータスＲＥＡＤ内容表示
; ----------------------------------------
                                        ;
RD_DATA_DSP:
                LEA     DI,RBUF
                MOV     DX,OFFSET ST1_MSG       ; ﾘﾝｸﾎﾞｰﾄﾞNO. 表示
                CALL    MESSAGE
```

```
                MOV     AL,DS:[DI]
                CALL    BIN_ASC
                CALL    BIN_DSP
                MOV     DX,OFFSET CRLF          ; 改行･復帰
                CALL    MESSAGE
                                                ;
                MOV     DX,OFFSET ST2_MSG       ; ﾈｯﾄﾜｰｸ加入局数表示
                CALL    MESSAGE
                MOV     AL,DS:[DI+2]
                CALL    BIN_ASC
                CALL    BIN_DSP
                MOV     DX,OFFSET CRLF          ; 改行･復帰
                CALL    MESSAGE
                                                ;
                MOV     DX,OFFSET ST3_MSG       ; ﾈｯﾄﾜｰｸ加入局情報表示
                CALL    MESSAGE
                MOV     CX,8
                MOV     BX,4
RD_DATA_LP:
                MOV     AL,DS:[DI+BX]
                CALL    BIN_ASC
                CALL    BIN_DSP
                MOV     AX,2020H
                CALL    BIN_DSP
                ADD     BX,2
                LOOP    RD_DATA_LP
                                                ;
                MOV     DX,OFFSET CRLF          ; 改行･復帰
                CALL    MESSAGE
                                                ;
                MOV     DX,OFFSET ST4_MSG       ; ﾘﾝｸﾎﾞｰﾄﾞ動作状態表示
                CALL    MESSAGE
                MOV     AL,DS:[DI+20]
                CMP     AL,0                    ; 正常?
                JE      ST4_OK
                CMP     AL,1                    ; 制御ﾎﾞｰﾄﾞ異常?
                JE      ST4_BD1
                MOV     DX,OFFSET ST4_BD2_MSG
                JMP     ST4_DSP
ST4_OK:
                MOV     DX,OFFSET ST4_OK_MSG
                JMP     ST4_DSP
ST4_BD1:
                MOV     DX,OFFSET ST4_BD1_MSG
ST4_DSP:
                CALL    MESSAGE
                                                ;
                MOV     DX,OFFSET ST5_MSG       ; ﾘﾝｸﾎﾞｰﾄﾞ通信状態表示
                CALL    MESSAGE
                MOV     DX,OFFSET ST5_OK_MSG
                MOV     AL,DS:[DI+22]
                CMP     AL,0                    ; 正常?
                JE      ST5_DSP
                MOV     DX,OFFSET ST5_BD_MSG
ST5_DSP:
                CALL    MESSAGE
                                                ;
                MOV     DX,OFFSET ST6_MSG       ; PCﾘﾝｸ実施局数表示
                CALL    MESSAGE
                MOV     AL,DS:[DI+24]
                CALL    BIN_ASC
                CALL    BIN_DSP
                                                ;
                MOV     DX,OFFSET CRLF          ; 改行･復帰
                CALL    MESSAGE
                MOV     DX,OFFSET CRLF          ; 改行･復帰
                CALL    MESSAGE
                                                ;
                RET
                                                ;
                                                ;
BIN_DSP:                                        ; IN...AX=表示文字ｺｰﾄﾞ
                PUSH    AX
                MOV     DL,AL
                MOV     AH,2
                INT     21H
```

```
                POP     AX
                MOV     DL,AH
                MOV     AH,2
                INT     21H
                RET

                                                ;
                                                ;
;  ============================================
;  =            I／Oパラメータ                =
;  ============================================
                                                ; = 初期設定 =
INIT            DW      1000H                   ; 初期設定ｺﾏﾝﾄﾞｺｰﾄﾞ
                DB      0                       ; ｱﾌﾟﾘｹｰｼｮﾝ･ｿﾌﾄﾓｰﾄﾞ
                DB      1                       ; 使用ﾎﾞｰﾄﾞ枚数
                                                ;
                                                ; = ﾎﾞｰﾄﾞ登録･起動 =
BOARD           DW      1100H                   ; ﾎﾞｰﾄﾞ登録･起動ｺﾏﾝﾄﾞｺｰﾄﾞ
                DB      0                       ; ﾎﾞｰﾄﾞ登録No.
                DB      27                      ; 使用ｾｸﾞﾒﾝﾄｱﾄﾞﾚｽNo.27(D0000H)
                DB      0                       ; 割込ﾀｲﾌﾟNo.
                DB      0
                                                ;
CTWT            DW      1200H                   ; = ｺﾝﾄﾛｰﾙﾚｼﾞｽﾀWRITE =
                DB      0                       ; 登録ﾎﾞｰﾄﾞNO.
                DB      0                       ; 0固定
                DW      132                     ; WRITEﾚｼｾｽﾀNO.
                DW      1                       ; WRITEﾃﾞｰﾀ数
                DW      OFFSET WBUF             ; WRITEﾊﾞｯﾌｧOFFSET
                DW      SEG WBUF                ;          SEGMENT
                                                ;
STRD            DW      1300H                   ; = ｽﾃｰﾀｽﾚｼﾞｽﾀREAD =
                DB      0                       ; 登録ﾎﾞｰﾄﾞNO.
                DB      0                       ; 0固定
                DW      6                       ; READﾚｼﾞｽﾀNO.
                DW      13                      ; READﾃﾞｰﾀ数
                DW      OFFSET RBUF             ; READﾊﾞｯﾌｧOFFSET
                DW      SEG RBUF                ;         SEGMENT
                                                ;
                                                ;
                                                ;
;  ============================================
;  =        メッセージ ／ バッファ領域        =
;  ============================================
                                                ;
MSG1            DB      '初期設定===>$'

MSG3            DB      'ｴﾗｰ.....ｺｰﾄﾞ='
MSG3_1          DB      0,0,0,0
                DB      CR,LF,'$'

MSG4            DB      '使用ボード登録 & 起動===>$'

CTWT_MSG        DB      'ｺﾝﾄﾛｰﾙﾚｼﾞｽﾀWRITE===>$'

STRD_MSG        DB      'ｽﾃｰﾀｽﾚｼﾞｽﾀREAD===>$'

OK_MSG          DB      'ＯＫ！',CR,LF,'$'

ST1_MSG         DB      'ﾘﾝｸﾎﾞｰﾄﾞNO.       =$'
ST2_MSG         DB      'ﾈｯﾄﾜｰｸ加入局数   =$'
ST3_MSG         DB      'ﾈｯﾄﾜｰｸ加入局情報=$'
ST4_MSG         DB      'ﾘﾝｸﾎﾞｰﾄﾞ動作状態=$'
ST5_MSG         DB      'ﾘﾝｸﾎﾞｰﾄﾞ通信状態=$'
ST6_MSG         DB      'PCﾘﾝｸ実施局数    =$'

ST4_OK_MSG      DB      '動作正常',CR,LF,'$'
ST4_BD1_MSG     DB      '制御ﾎﾞｰﾄﾞ部異常',CR,LF,'$'
ST4_BD2_MSG     DB      '通信系異常',CR,LF,'$'

ST5_OK_MSG      DB      '通信正常',CR,LF,'$'
ST5_BD_MSG      DB      '通信異常',CR,LF,'$'

CRLF            DB      CR,LF,'$'

WBUF            DW      3                       ; WRITEﾊﾞｯﾌｧ
RBUF            DB      256 DUP(0)              ; READﾊﾞｯﾌｧ
```

パソコン機種別変更箇所
（左記はＰＣ９８シリーズの場合）

```
STACK       DW      256 DUP(0)          ; ｽﾀｯｸ領域
STPT        DB      0

            CODE    ENDS
            END     START
```

パソコン機種別変更箇所

ＰＣ／ＡＴ互換機

```
BOARD       DW      1100H               ; ﾎﾞｰﾄﾞ登録･起動ｺﾏﾝﾄﾞｺｰﾄﾞ
            DB      0                   ; ﾎﾞｰﾄﾞ登録No.
            DB      8                   ; 使用ｾｸﾞﾒﾝﾄｱﾄﾞﾚｽNo.
            DB      10                  ; 割り込みNo.
            DB      0
```

ＦＭＲシリーズ

```
BOARD       DW      1100H               ; ﾎﾞｰﾄﾞ登録･起動ｺﾏﾝﾄﾞｺｰﾄﾞ
            DB      0                   ; ﾎﾞｰﾄﾞ登録No.
            DB      7                   ; 基板I/OﾎﾟｰﾄｱﾄﾞﾚｽNo.
            DW      7800H               ; ｺﾝﾄﾛｰﾙ･ｽﾃｰﾀｽI/Oﾎﾟｰﾄｱﾄﾞﾚｽ
            DB      5                   ; 割り込みNo.
            DB      0
```

Ｃプログラム（ユーザー割り込みint40以降）

```
/*************************************************************/
/*          コントロール／ステータスレジスタＷＴ／ＲＤ       */
/*                        (CTST.C)                           */
/*************************************************************/
/*                    ＰＣ９８シリーズ                       */
/*                                                           */
/*     ｺﾝﾄﾛｰﾙﾚｼﾞｽﾀWRITE:                                     */
/*     ｺﾝﾋﾟｭｰﾀﾘﾝｸ用送受信ﾀｲﾑｱｳﾄ値を10秒から3秒に変更します。*/
/*                                                           */
/*     ｽﾃｰﾀｽﾚｼﾞｽﾀREAD:                                       */
/*     ﾘﾝｸﾎﾞｰﾄﾞNO. 及びﾈｯﾄﾜｰｸ加入局情報を読出します。        */
/*                                                           */
/*     使用コマンド：WRITE=1200H(ﾀｲﾑｱｳﾄ付き)                 */
/*                   READ =1300H(ﾀｲﾑｱｳﾄ付き)                 */
/*************************************************************/

#include  <dos.h>
#include  <stdio.h>
#include  <stdlib.h>


static union REGS inregs;
static union REGS outregs;
struct SREGS segregs;
static char far *p1;
static unsigned char dat[48];
unsigned int wbuf[256];                    /* WRITEﾊﾞｯﾌｧ  */
unsigned int rbuf[256];                    /* READﾊﾞｯﾌｧ   */

void main( void )
{
       int    i;
       int    err;
       long   lwork;
       char   cwork[5];
       char   c_dmy1[4];
       char   *dmy;

/* ---------------------------*/
/* ------ 初期設定処理 ------*/
/* ---------------------------*/

       dat[0] = 0x00;        /* 初期設定ｺﾏﾝﾄﾞ･ｺｰﾄﾞ=1000H  */
       dat[1] = 0x10;
       dat[2] = 0;           /* ｱﾌﾟﾘｹｰｼｮﾝ･ｿﾌﾄﾓｰﾄﾞ=0       */
       dat[3] = 1;           /* 使用ボード枚数=1           */

       printf("初期設定===>");

       if((err = mew_send()) != 0)
       {
              printf("ｴﾗｰ.....ｺｰﾄﾞ=  %x¥n",  err);
              exit(-1);
       }
       printf("ＯＫ！¥n");

/* ------------------------------------*/
/* ------ 使用ボード登録 及び起動 ------*/
/* ------------------------------------*/

       dat[0] = 0x00;        /* 使用ボード登録及び起動    */
       dat[1] = 0x11;        /* ｺﾏﾝﾄﾞｺｰﾄﾞ=1100           */
       dat[2] = 0x00;        /* 登録ボードNo.=0           */
       dat[3] = 27;          /* 使用ｾｸﾞﾒﾝﾄｱﾄﾞﾚｽNo.=27    */
       dat[4] = 0;           /* 割り込みNo.=0             */

       printf("使用ボード登録 & 起動===>");
```

パソコン機種別変更箇所
（左記はＰＣ９８シリーズの場合）

```
        if((err = mew_send()) != 0)
        {
                printf("ｴﾗｰ.....ｺｰﾄﾞ= %x¥n", err);
                exit(-1);
        }
        printf("ＯＫ！¥n");

/* ------------------------------*/
/* ------ ｺﾝﾄﾛｰﾙﾚｼﾞｽﾀWRITE ------*/
/* ------------------------------*/

        dat[0] = 0x00;
        dat[1] = 0x12;          /* ｺﾏﾝﾄﾞｺｰﾄﾞ=1200H    */
        dat[2] = 0x00;          /* 登録ﾎﾞｰﾄﾞNo.=0     */
        dat[3] = 0x00;          /* 00固定      */
        *((unsigned *)&dat[4]) = 132;       /* WRITEﾚｼﾞｽﾀNo.       */
        *((unsigned *)&dat[6]) = 1;         /* WRITEﾃﾞｰﾀ数*/
        p1 = (char far *)&wbuf[0];
        *((unsigned *)&dat[8]) = FP_OFF(p1);        /* WRITEﾊﾞｯﾌｧｵﾌｾｯﾄ    */
        *((unsigned *)&dat[10]) = FP_SEG(p1);       /* WRITEﾊﾞｯﾌｧｾｸﾞﾒﾝﾄ   */

        wbuf[0] = 3;                        /* WRITEﾃﾞｰﾀ   */

        printf("ｺﾝﾄﾛｰﾙﾚｼﾞｽﾀWRITE===>");

        if((err = mew_send()) != 0)
        {
                printf("ｴﾗｰ.....ｺｰﾄﾞ= %x¥n", err);
                exit(-1);
        }
        printf("ＯＫ！¥n");

/* ------------------------------*/
/* ------ ｽﾃｰﾀｽﾚｼﾞｽﾀWRITE ------*/
/* ------------------------------*/

LP:
        dat[0] = 0x00;
        dat[1] = 0x13;          /* ｺﾏﾝﾄﾞｺｰﾄﾞ=1200H    */
        dat[2] = 0x00;          /* 登録ﾎﾞｰﾄﾞNo.=0     */
        dat[3] = 0x00;                      /* 00固定              */
        *((unsigned *)&dat[4]) = 6;         /* READﾚｼﾞｽﾀNo.        */
        *((unsigned *)&dat[6]) = 13;        /* READﾃﾞｰﾀ数          */
        p1 = (char far *)&rbuf[0];
        *((unsigned *)&dat[8]) = FP_OFF(p1);        /* READﾊﾞｯﾌｧｵﾌｾｯﾄ    */
        *((unsigned *)&dat[10]) = FP_SEG(p1);       /* READﾊﾞｯﾌｧｾｸﾞﾒﾝﾄ   */

        printf("ｽﾃｰﾀｽﾚｼﾞｽﾀWRITE===>");

        if((err = mew_send()) != 0)
        {
                printf("ｴﾗｰ.....ｺｰﾄﾞ= %x¥n", err);
        }
        else
        {
                printf("ＯＫ！¥n");
                printf("ﾘﾝｸﾎﾞｰﾄﾞNo.       =%d¥n",rbuf[0]);
                printf("ﾈｯﾄﾜｰｸ加入局数   =%d¥n",rbuf[1]);
                printf("ﾈｯﾄﾜｰｸ加入情報   =");
                for(i = 0 ; i < 8 ; i++)
                        printf("%x ",rbuf[i+2]);
                printf("¥n");

                printf("ﾘﾝｸﾎﾞｰﾄﾞ動作状態=");
                if(rbuf[10] == 0)
                        printf("動作正常¥n");
                else
                {
                        if(rbuf[10] == 1)
                                printf("制御ﾎﾞｰﾄﾞ部異常¥n");
                        else
                                printf("通信系異常¥n");
                }

                printf("ﾘﾝｸﾎﾞｰﾄﾞ通信状態=");
```

```
                if(rbuf[11] == 0)
                        printf("通信正常¥n");
                else
                        printf("通信異常¥n");

                printf("PCリンク 実施局数    =%d¥n",rbuf[12]);
        }
        printf("¥n");

        goto LP;
}


/* ---------------------------*/
/* ------ リンクソフト呼び出し------*/
/* ---------------------------*/
int mew_send(void)
{

        p1 = (char far *)&dat[0];

        inregs.x.dx = FP_OFF(p1);
        segregs.ds = FP_SEG(p1);

        int86x( 0x60, &inregs, &outregs, &segregs);
        return(outregs.x.ax);
}
```

パソコン機種別変更箇所

ＰＣ／ＡＴ互換機

```
        dat[0] = 0x00;          /* 使用ボード登録及び起動    */
        dat[1] = 0x11;          /* ｺﾏﾝﾄﾞｺｰﾄﾞ=1100H           */
        dat[2] = 0x00;          /* 登録ボードNo.=0           */
        dat[3] = 8;             /* 使用ｾｸﾞﾒﾝﾄｱﾄﾞﾚｽNo.=8      */
        dat[4] = 10;            /* 割り込みNo.=10      */
```

ＦＭＲシリーズ

```
        dat[0] = 0x00;          /* 使用ボード登録及び起動    */
        dat[1] = 0x11;          /* ｺﾏﾝﾄﾞｺｰﾄﾞ=1100H           */
        dat[2] = 0x00;          /* 登録ボードNo.=0           */
        dat[3] = 7;             /* 基板I/OﾎﾟｰﾄｱﾄﾞﾚｽNo.=7     */
        dat[4] = 0x00;          /* ｽﾃｰﾀｽ･ｺﾝﾄﾛｰﾙI/Oﾎﾟｰﾄｱﾄﾞﾚｽ */
        dat[5] = 0x78;          /* =7800H                    */
        dat[6] = 5;             /* 割り込みNo.=5             */
```

3－9 MEWNETーHリンクソフトのエラーコード

コード	内容	備考
101H	未登録ボードNo.指定	登録されていないボードNo.が指定された。 No.0～3以外が指定された。
102H	CH No.指定エラー	機能コマンドで使用できないCH No.が指定された。 指定CHが使用許可されていない。
103H	セグメントNo.指定エラー	範囲外のNo.が指定された。 指定されたセグメントNo.ではアクセスできない。 指定セグメント重複。
104H	割り込みNo.指定エラー	範囲外の割り込みNO.が指定された。
105H	書き込み開始レジスタNO.指定エラー	書き込み開始レジスタNO.が書き込み可能なレジスタでない。
106H	書き込みデータ数指定エラー	書き込み可能な範囲をオーバーしている。
107H	読み出し開始レジスタNO.指定エラー	読み出し開始レジスタNO.が読み出し可能なレジスタでない。
108H	読み出しデータ数指定エラー	読み出し可能な範囲をオーバーしている。
109H	割り込み登録エラー	存在していないラベル名を指定している。
10AH	割り込み許可･禁止制御コード指定エラー	0（禁止）、1（許可）以外を指定している。
10BH	アプリケーションソフトモード指定エラー	0（禁止）、1（許可）以外を指定している。
10CH	使用ボード数エラー	1～4以外を指定している。
10DH	送信先NO.指定エラー	1～64またはFFh以外を指定した。
10EH	配列変数指定エラー	存在しない配列名を指定している。 配列の要素が足りない。
10FH	割り込み不可エラー	登録されていないのに許可・禁止しようとした。
110H	初期設定未実行	初期設定処理コマンド（&H1000）が実行されていません。
111H	バッファ未定義エラー	
112H	コントロール/ステータスI/Oポート指定エラー	指定されたアドレスではアクセスできません。
113H	登録オーバー	登録タイプコマンドの使用オーバーです。
114H	データエラー	指定されたデータ内容に誤りがあります。
1FEH	コマンドＮＯＴサポート	指定されたコマンドはサポートしていません。
1FFH	コマンドエラー	
130H	すでに送信要求済み	
131H	送受信完了タイムアウトエラー	
132H	KEY BREAKによる送受信待ち中断	
133H	受信完了センスできず	
134H	送信完了センスできず	
135H	受信バッファサイズ・エラー	受信データ・サイズ>受信バッファ・サイズ
136H	リンクボード起動エラー	
137H	デリミタ受信エラー	“&”待ち時にそれ以外を受信した。 “&”を受信できなかった。
138H	受信データエラー	受信データのフォーマット異常
139H	手順乱れエラー	
13AH	センスタイプマルチフレーム受信	
13BH	送信要求のみコマンドフレームエラー	送信要求のみコマンドでマルチフレーム送信をした
13CH	送信バッファＦＵＬＬエラー	送信バッファＦＵＬＬ状態のため送信を中断した
13DH	ＰＣリンク停止中	
13EH	ＰＣリンクＣＬＯＳＥ状態	
13FH	ＰＣリンク非使用モード	ディップスイッチがＯＦＦになっている。
140H	ＰＣリンクパラメータエラー	不正なパラメータがセットされた。 パラメータなし状態。 チェック未完了。その他。

コード	内容	備考
141H	ＰＣリンク書き込みパラメータエラー	読み出し範囲に対して書き込みをしようとした。
142H	ＰＣリンクモードエラー	ランダムタイプ使用中に通常のＲ／Ｗを使用とした。
143H	ＰＣリンクＲ／Ｗ範囲リンク領域オーバー	Ｒ／Ｗ範囲がリンクボードのリンク領域を越えている。
144H	登録強制解除	
145H	登録機能使用不可	登録機能をサポートしていません。
146H	登録数＝０状態	
147H	ＰＣリンクＲ／Ｗポインタリンク範囲外	Ｒ／Ｗ開始ポインタがリンクボードのリンク範囲を越えている。
148H	同一内容登録済みエラー	登録タイプですでに同一の内容が登録されています。 送信先No.、CH等が重複している。
149H	通信局指定エラー	シリアル伝送機能多重接続マスターモードで通信局の設定がされていません。。
14AH	通信不可状態	

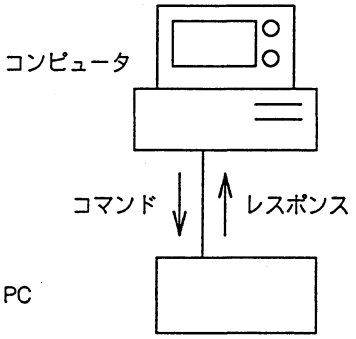
4章 MEWNETーHのプロトコル

4 － 1　MEWTOCOL-COM（コンピュータリンク）

4-1-1 MEWTOCOL-COMの概要

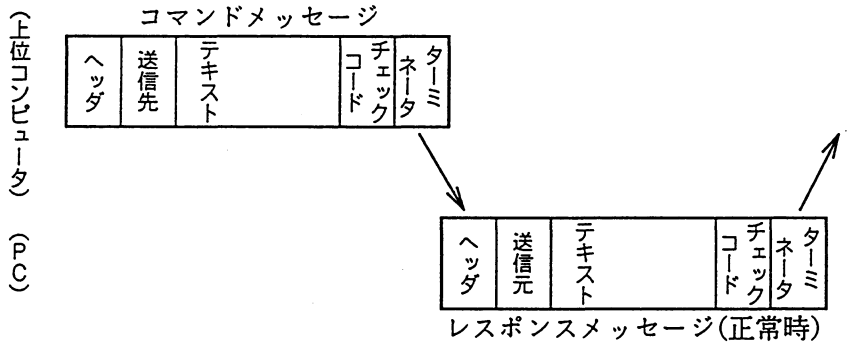
■コマンド／レスポンスの機能

コンピュータはPCにコマンド（命令）を送り、レスポンス（返事）を受け取ります。この手順によりコンピュータはPCと会話が行え、各種情報を得たり、与えたりすることができます。



注意　・コンピュータリンクを行うにはコンピュータ側のユーザープログラムのみ必要です。
PC側のプログラムは必要としません。

■コマンド／レスポンスのフォーマット



注意　・専用手順・会話形になっています。
・ASCIIコード送りです。
・最初の送信権は、コンピュータ側にあります。
・メッセージを送信するごとに送信権を移行します。

●制御コード

名称	キャラクタ	ASCIIコード	説　明
ヘッダ	%	25H	メッセージの開始を示す。
コマンド	#または<	23H／3CH	コマンド・メッセージであることを示す。
レスポンス（正常）	$	24H	正常なレスポンス・メッセージであることを示す。
レスポンス（異常）	!	21H	エラー時のレスポンス・メッセージであることを示す。
ターミネータ	C_R	0DH	メッセージの終了を示す。
デリミタ	&(+ C_R)	26H	複数フレームに分割する時の区別を示す。

●送信先、送信元 AD（H),(L)

2桁の10進数　01～32（ASCIIコード）
コマンドメッセージ内では、コマンドメッセージを受取るべきPCのUNIT No.を示します。
レスポンスメッセージ内では、レスポンスメッセージを送出したPCのUNIT No.を示します。
（H）は上の桁、（L）は下の桁を示します。
特に指定がなければ、"01"と指定してください。
ただし、FF（ASCIIコード）の時は、グローバル転送（全ユニットへの一斉転送＊）です。
注＊　グローバル転送を行った場合、そのコマンドメッセージに対するレスポンスメッセージは返送しません。

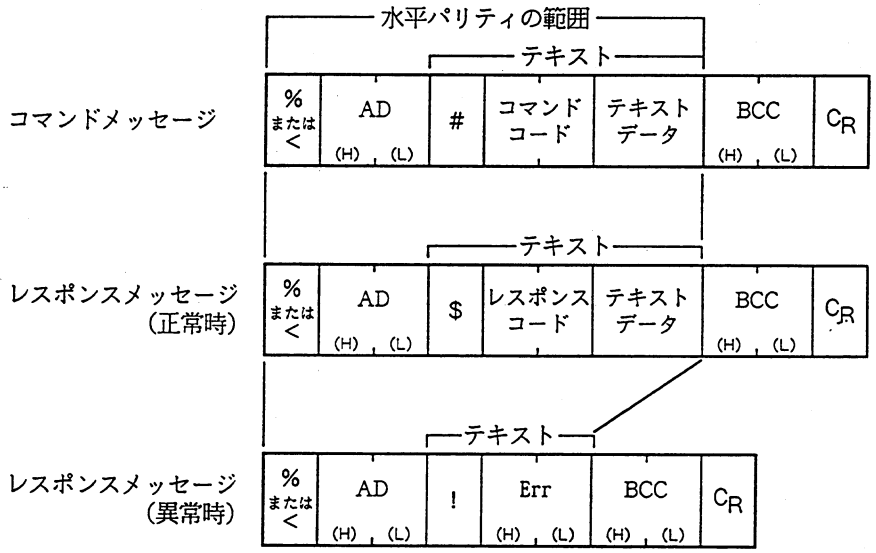
●ブロックチェックコード BCC（H),(L)

2桁の16進数　00～FF（ASCIIコード）
伝送データの誤り検出用のコード（水平パリティ）です。
ただし、BCCの代わりに＊＊を入れた場合は、BCCなしで伝送が可能です。この場合もレスポンスにはBCCが付いてきます。

●エラーコード Err（H),(L)

2桁の16進数　00～FF（ASCIIコード）
エラー発生時にその内容を示します。

■単一フレームのコマンド／レスポンス

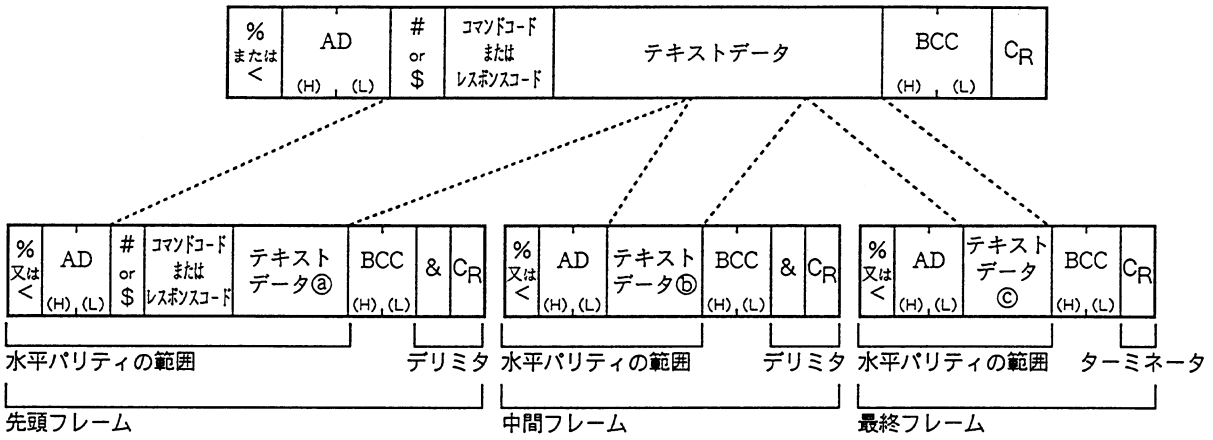


■最大メッセージ長

コマンド／レスポンスの単一フレームにおける最大メッセージ長（ヘッダからターミネータまでの文字数）を以下に示します。最大メッセージ長を越える場合は、複数フレームに分割して送信してください。（レスポンスの場合は複数フレームに分割して返信されます）。

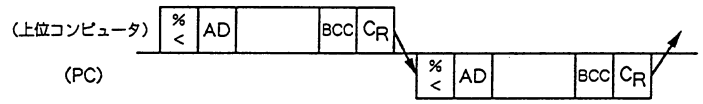
% （ヘッダ）	118文字
< （拡張ヘッダ）	2048文字 ただし、機種およびコマンドにより制約があります。 参照 機種およびコマンドによる最大メッセージ長の制約については「コマンド一覧表」（P. 252）をご参照ください。

■複数フレームのコマンド／レスポンス

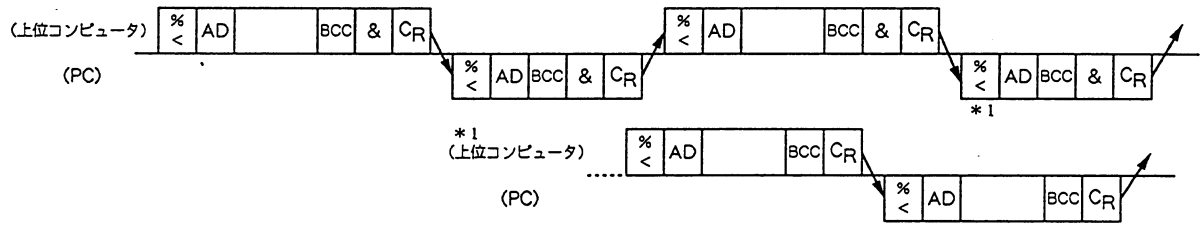


■通信タイムチャート例

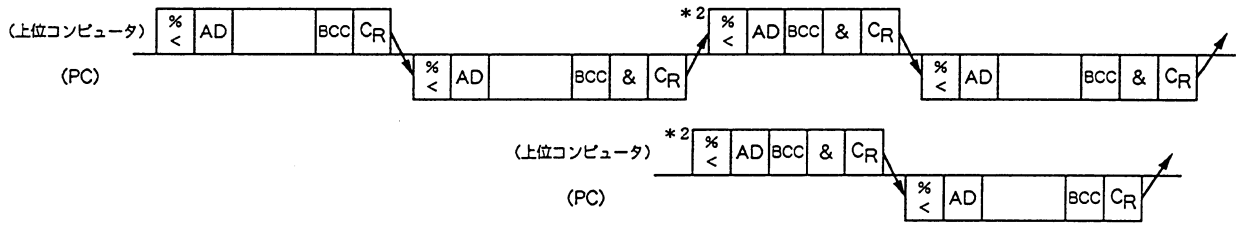
①単一フレームコマンド・単一フレームレスポンス



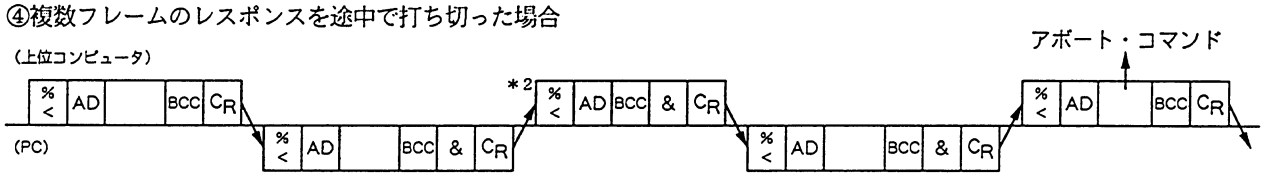
②複数フレームコマンド・単一フレームレスポンス



③単一フレームコマンド・複数フレームレスポンス



④複数フレームのレスポンスを途中で打ち切った場合

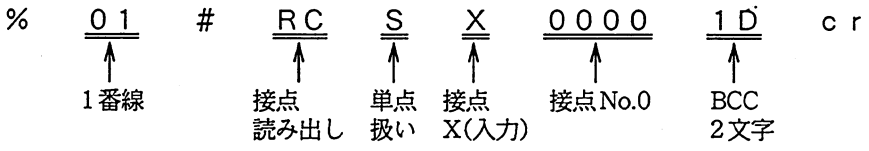


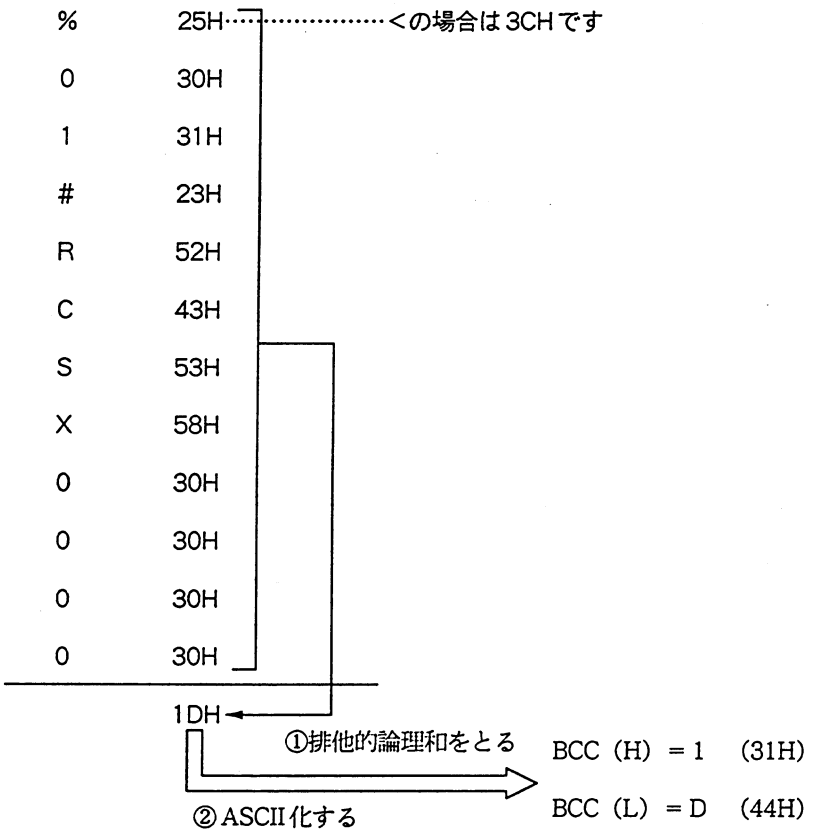
注意 複数フレームに分割して送る時は、1つのフレーム送信後、相手側の送信要求メッセージ（通信タイムチャート例の＊1）を受信するまで次のフレームの送信はできません。
複数フレームを受信する時は、次のフレームの受信を行うためには送信要求メッセージ（通信タイムチャート例の＊2）を相手側に送信してください。

■BCC（ブロックチェックコード）

BCCは、伝送データの信頼性を向上させる為に、本機の場合水平パリティを用いた誤りチェックを行うためのコードです。
BCCは、ヘッダ（%）からテキストの最終文字までの排他的論理和をとり、その８ビットデータをASCIIコードの２文字に変換して作成します。

例）





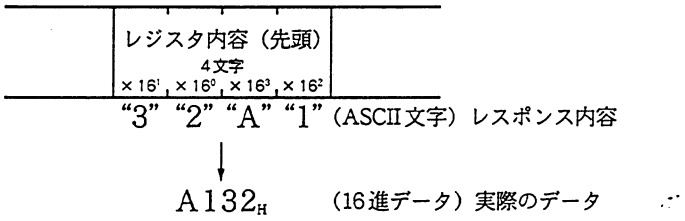
■コマンド・レスポンスの表記方法

コマンド／レスポンスで使用するデータの表記は下記の３種類があります。

・16進データ

×16^0、×16^1 ～は、16進データを示しています。

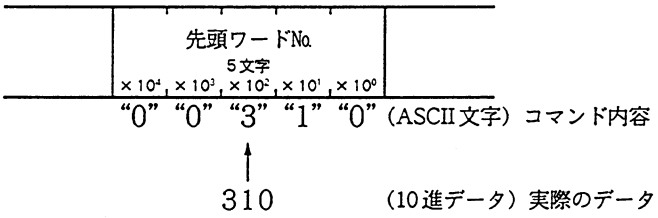
（例）データエリアリード（RD）のレスポンス部のレジスタ内容



・10進データ

×10^0、×10^1 ～は、10進データを示しています。

（例）データエリアリード（RD）のコマンド部の先頭ワード内容

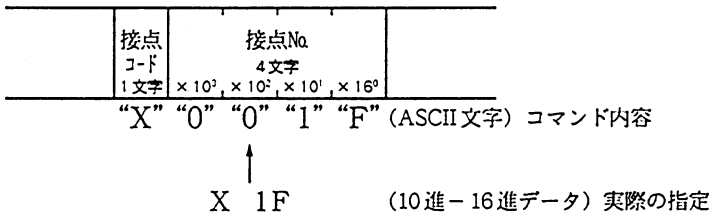


・10進－16進データ

I／O（X、Y）、内部リレー（CR）、リンクリレー（LR）の接点番号は、最下位桁は、16進数表記となっており、それ以上の上位桁は、10進表記になっています。（T／Cの接点番号は最下位桁まで10進表記です）

この場合は、×16^0、×10^1、×10^2 ～と表記しています。

（例）接点エリアリード（RCS）のコマンド部の接点指定



注意 データには、文字数の指定があります。例えば上記の「接点No.」は、４文字（４ケタ）で指定しますので、X1Fの接点エリアを読み出す場合は接点No.を「001F」と頭に０をつめて、４文字（４ケタ）にしてください。

■コマンド一覧表

コマンド名称	内容説明	コード	BASIC CPU	掲載ページ
接点エリアリード	接点のON／OFF状態を読み出す ・一点のみ指定する ・複数の接点を指定する ・ワード単位での範囲を指定する	RC (RCS) (RCP) (RCC)	○	253 253 254
接点エリアライト	接点をON／OFFします ・一点のみ指定する ・複数の接点を指定する ・ワード単位での範囲を指定する	WC (WCS) (WCP) (WCC)	○	254 255 255
接点エリアのプリセット (フィルコマンド)	指定した範囲のエリアを16点分のON／OFFパターンでうめる	SC	○	256
データエリアリード　（*）	データエリアの内容を読み出す	RD	○	257
データエリアライト	データエリアにデータを書き込む	WD	○	258
データエリアのプリセット (フィルコマンド)	指定した範囲のデータエリアに同じ内容を書き込む	SD	○	259
タイマ／カウンタ設定値エリアリード	タイマ／カウンタの設定値を読み出す	RS	×	260
タイマ／カウンタ設定値エリアライト	タイマ／カウンタの設定値を書き込む	WS	×	260
タイマ／カウンタ経過値エリアリード	タイマ／カウンタの経過値を読み出す	RK	×	261
タイマ／カウンタ経過値エリアライト	タイマ／カウンタの経過値を書き込む	WK	×	261
モニタ接点登録・登録リセット	モニタする接点を登録する	MC	○	262
モニタデータ登録・登録リセット	モニタするデータを登録する	MD	○	263
モニタ実行	登録した接点やデータをモニタする	MG	○	264
システムレジスタリード　（*）	システムレジスタ内容を読み出す	RR	×	265
システムレジスタライト	システムレジスタ内容を設定する	WR	×	265
PCステータスリード	PCの仕様、エラー発生時のエラーコードなどを読み出す	RT	○	266
プログラムブロックリード　（*）	PCに書き込まれているプログラムを読み出す	RP	×	268
プログラムブロックライト	PCにデータ化されたプログラムを書き込む	WP	×	269
リモートコントロール	PCの動作モードを切り替える	RM	○	269
アボート	通信を途中で打ち切る	AB	○	269
階層コントロール	階層間リンク使用時に別の階層のPCと通信する	LC	○	270

注意 上記（*）マークのコマンドは、FP3ラダーCPUの場合、<ヘッダによる単一フレームでのレスポンスの最大長は1953文字（486ワードの読み出しに相当）です。またこれ以上の読み出しの場合は、複数フレームレスポンスとなります。

4-1-2 MEWTOCOL-COMコマンドリファレンス

[RCS] 接点エリアリード（単点）

接点のON／OFF状態を一点のみ読み出します。

■コマンド

% または <	送り先	#	R	C	S	接点コード	接点No.	BCC	CR
	×10¹, ×10⁰					1文字	4文字 ×10³, ×10², ×10¹, (×10⁰)	×16¹, ×16⁰	

S：単点扱い

接点No.：( ) 内は、TM／CTの場合です。

■レスポンス（リードOK）

% または <	送り元	$	R	C	接点データ	BCC	CR
	×10¹, ×10⁰				1文字	×16¹, ×16⁰	

（リードエラー）

% または <	送り元	!	エラーコード	BCC	CR
	×10¹, ×10⁰		×16¹, ×16⁰	×16¹, ×16⁰	

接点コード

接点	表記
X	“X”
Y	“Y”
CR	“R”
LR	“L”
TM	“T”
CT	“C”

接点データ

接点	表記
OFF	“0”
ON	“1”

[RCP] 接点エリアリード（複数点）

複数の接点のON／OFF状態を読み出します。

■コマンド

% または <	送り先	#	R	C	P	n	接点コード	接点No.	BCC	CR
	×10¹, ×10⁰					1文字 ×10⁰	1文字	4文字 ×10³, ×10², ×10¹, (×10⁰)		
							1文字	4文字 ×10³, ×10², ×10¹, (×10⁰)	×16¹, ×16⁰	

接点の指定 ①

P：複数点扱い　n：接点数 n=1～8

( ) 内は、TM／CTの場合です。

接点の指定 ⓝ

■レスポンス（リードOK）

% または <	送り元	$	R	C	接点データ①	接点データⓝ	BCC	CR
	×10¹, ×10⁰				1文字	1文字	×16¹, ×16⁰	

（リードエラー）

% または <	送り元	!	エラーコード	BCC	CR
	×10, ×10⁰		×16¹, ×16⁰	×16¹, ×16⁰	

接点コード

接点	表記
X	“X”
Y	“Y”
CR	“R”
LR	“L”
TM	“T”
CT	“C”

接点データ

接点	表記
OFF	“0”
ON	“1”

[RCC] 接点エリアリード（ワード単位ブロック）

接点のON／OFF状態をワード単位で読み出します。

■コマンド

% または <	送り先 ×10¹, ×10⁰	#	R	C	C	接点コード 1文字	先頭ワードNo. 4文字 ×10³, ×10², ×10¹, ×10⁰	最終ワードNo. 4文字 ×10³, ×10², ×10¹, ×10⁰	BCC ×16¹, ×16⁰	CR

ワード扱い

■レスポンス（リードOK）

% または <	送り元 ×10¹, ×10⁰	$	R	C	接点情報（先頭） 4文字 ×16¹, ×16⁰, ×16³, ×16²	.....	接点情報（最終） 4文字 ×16¹, ×16⁰, ×16³, ×16²	BCC ×16¹, ×16⁰	CR

（下位）（上位）　（下位）（上位）

（リードエラー）

% または <	送り元 ×10¹, ×10⁰	!	ｴﾗｰｺｰﾄﾞ ×16¹, ×16⁰	BCC ×16¹, ×16⁰	CR

接点コード

接点	表記
X	“X”
Y	“Y”
CR	“R”
LR	“L”
TM	“T”
CT	“C”

・接点情報は、ワード単位に16進数にて読み出されます。

[WCS] 接点エリアライト（単点）

接点を一点のみON／OFFします。

■コマンド

% または <	送り先 ×10¹, ×10⁰	#	W	C	S	接点コード 1文字	接点No. 4文字 ×10³, ×10², ×10¹, ×16⁰	接点データ 1文字	BCC ×16¹, ×16⁰	CR

単点扱い

■レスポンス（ライトOK）

% または <	送り元 ×10¹, ×10⁰	$	W	C	BCC ×16¹, ×16⁰	CR

（ライトエラー）

% または <	送り元 ×10¹, ×10⁰	!	ｴﾗｰｺｰﾄﾞ ×16¹, ×16⁰	BCC ×16¹, ×16⁰	CR

接点コード

接点	表記
Y	“Y”
CR	“R”
LR	“L”

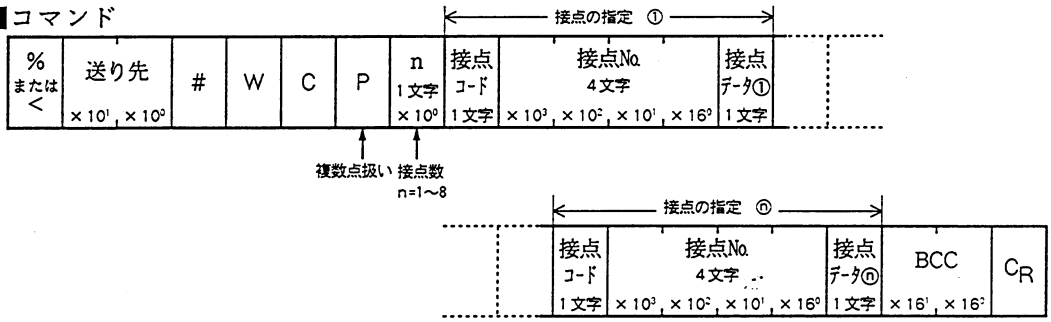
接点データ

接点	表記
OFF	“0”
ON	“1”

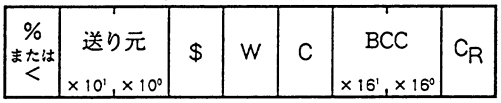
[WCP] 接点エリアライト（複数点）

複数の接点をON／OFFします。

■コマンド



■レスポンス（ライトOK）



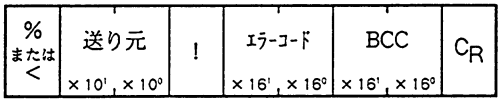
接点コード

接点	表記
Y	“Y”
CR	“R”
LR	“L”

接点データ

接点	表記
OFF	“0”
ON	“1”

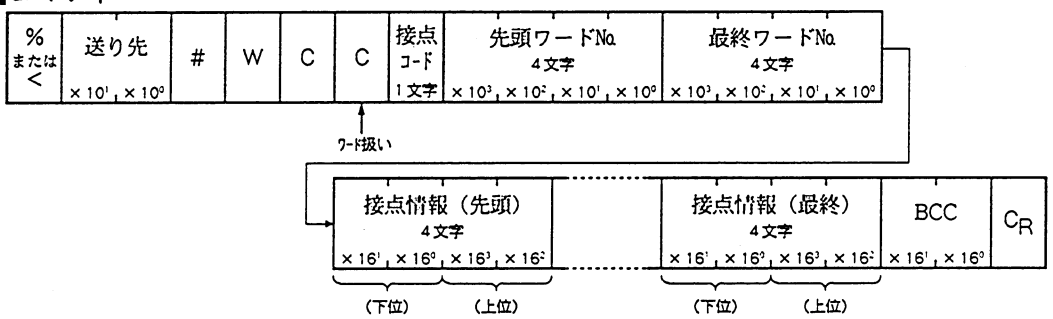
（ライトエラー）



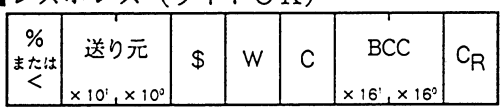
[WCC] 接点エリアライト（ワード単位ブロック）

接点をワード単位でON／OFFします。

■コマンド



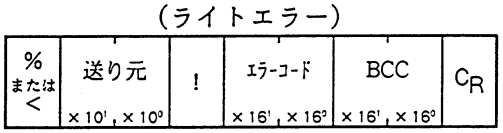
■レスポンス（ライトOK）



接点コード

接点	表記
Y	“Y”
CR	“R”
LR	“L”

（ライトエラー）

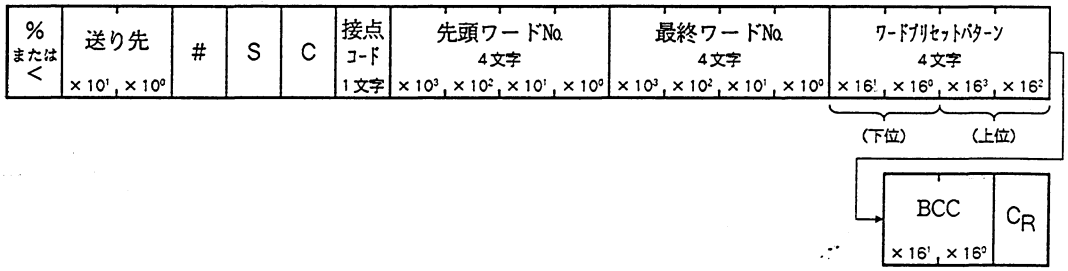


・接点情報は、ワード単位に16進数にて書かれます。

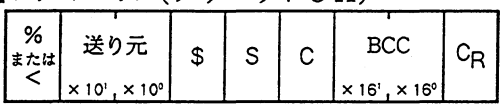
[SC] 接点エリアのプリセット（フィルコマンド）

指定した範囲のエリアを16点分のON／OFFでうめます。

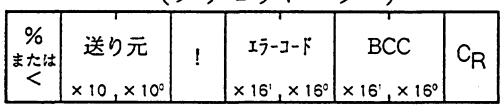
■コマンド



■レスポンス（プリセットOK）



（プリセットエラー）



接点コード

接点	表記
X	“X”
Y	“Y”
CR	“R”
LR	“L”
TM	“T”
CT	“C”

[RD] データエリアリード

データエリアの内容を読み出します。

DT、LD、FLの内容を読み出す場合

■コマンド

% または <	送り先	#	R	D	データコード	先頭ワードNo.	最終ワードNo.	BCC	CR
	×10¹, ×10⁰				1文字	5文字 ×10⁴, ×10³, ×10², ×10¹, ×10⁰	5文字 ×10⁴, ×10³, ×10², ×10¹, ×10⁰	×16¹, ×16⁰	

■レスポンス（リードOK）

% または <	送り元	$	R	D	レジスタ内容（先頭）	レジスタ内容（最終）	BCC	CR
	×10¹, ×10⁰				4文字 ×16¹, ×16⁰, ×16³, ×16²	4文字 ×16¹, ×16⁰, ×16³, ×16²	×16¹, ×16⁰	
					(下位) (上位)	(下位) (上位)		

（リードエラー）

% または <	送り元	!	エラーコード	BCC	CR
	×10¹, ×10⁰		×16¹, ×16⁰	×16¹, ×16⁰	

データコード

データ	表記
DT	“D”
LD	“L”
FL	“F”

インデックスレジスタの内容を読み出す場合

■コマンド

% または <	送り先	#	R	D	データコード	0 0 0 0 0 0 0 0 0	BCC	CR
	×10¹, ×10⁰				2文字	9文字	×16¹, ×16⁰	

■レスポンス　リードOK　（ＩＸまたは、ＩＹの場合）

% または <	送り元	$	R	D	レジスタ内容	BCC	CR
	×10¹, ×10⁰				4文字 ×16¹, ×16⁰, ×16³, ×16²	×16¹, ×16⁰	
					(下位) (上位)		

リードOK　（ＩＤの場合）

% または <	送り元	$	R	D	レジスタ内容（IX）	レジスタ内容（IY）	BCC	CR
	×10¹, ×10⁰				4文字 ×16¹, ×16⁰, ×16³, ×16²	4文字 ×16¹, ×16⁰, ×16³, ×16²	×16¹, ×16⁰	
					(下位) (上位)	(下位) (上位)		

（リードエラー）

% または <	送り元	!	エラーコード	BCC	CR
	×10¹, ×10⁰		×16¹, ×16⁰	×16¹, ×16⁰	

データコード

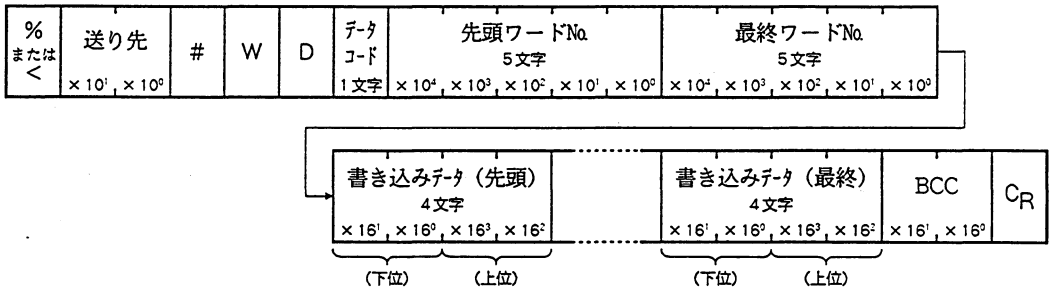
データ	表記
IX	“I” “X”
IY	“I” “Y”
IX,IY	“I” “D”

[WD] データエリアライト

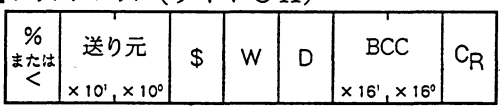
データエリアの内容を書き込みます。

DT、LD、FLの内容を書き込む場合

■コマンド



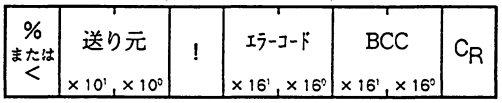
■レスポンス（ライトOK）



データコード

データ	表記
DT	“D”
LD	“L”
FL	“F”

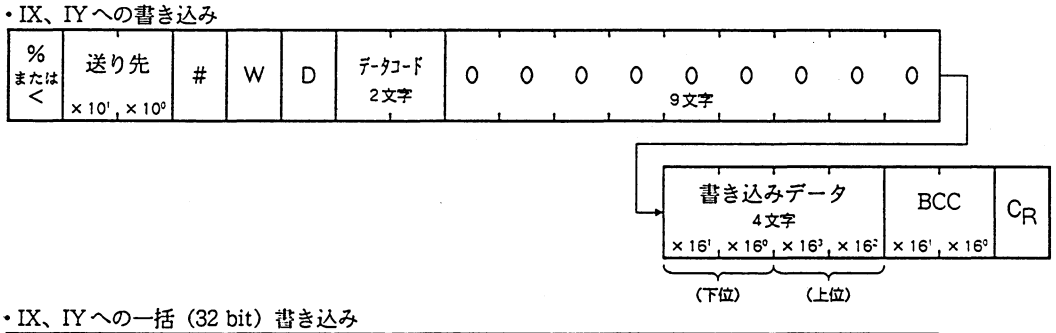
（ライトエラー）



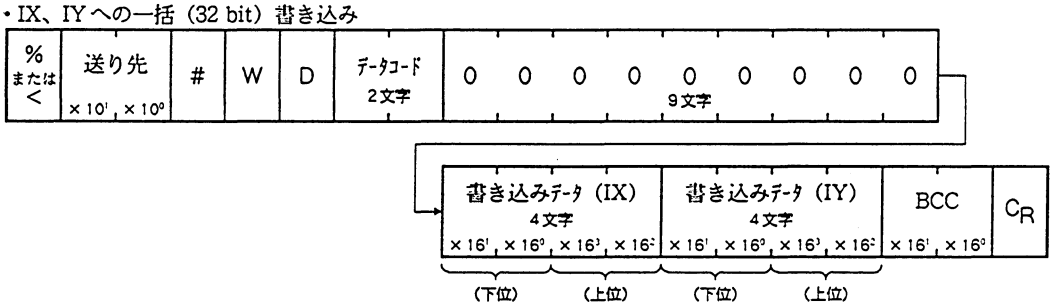
インデックスレジスタに書き込む場合

■コマンド

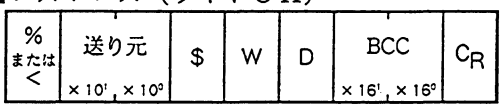
・IX、IYへの書き込み



・IX、IYへの一括（32 bit）書き込み



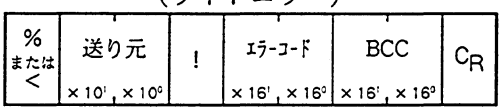
■レスポンス（ライトOK）



データコード

データ	表記
IX	“I” “X”
IY	“I” “Y”
IX,IY	“I” “D”

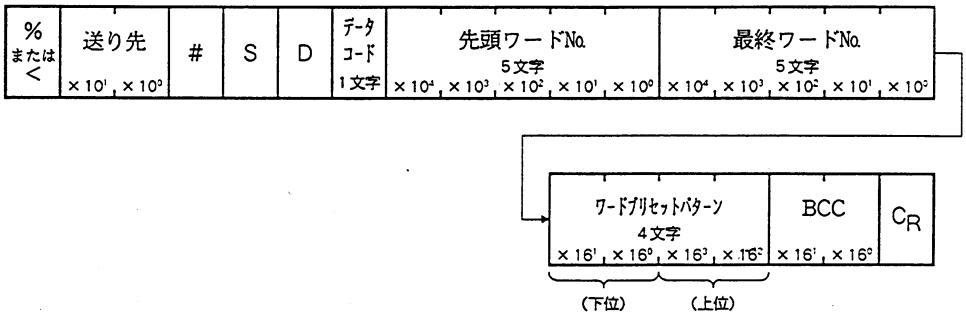
（ライトエラー）



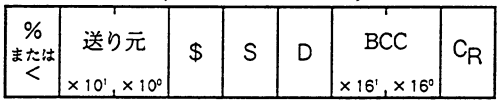
[SD] データエリアのプリセット

指定した範囲のデータエリアに同じ内容を書き込みます。

■コマンド



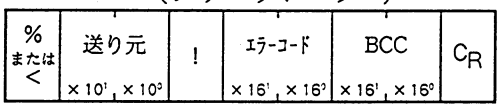
■レスポンス（プリセットOK）



データコード

データ	表記
DT	“D”
LD	“L”
FL	“F”

（プリセットエラー）



［RS］設定値エリアリード

タイマ／カウンタの設定値を読み出します。

■コマンド

% または <	送り先 ×10¹, ×10⁰	#	R	S	先頭ﾀｲﾏ/ｶｳﾝﾀNo. 4文字 ×10³, ×10², ×10¹, ×10⁰	最終ﾀｲﾏ/ｶｳﾝﾀNo. 4文字 ×10³, ×10², ×10¹, ×10⁰	BCC ×16¹, ×16⁰	CR

■レスポンス（リードOK）

% または <	送り元 ×10¹, ×10⁰	$	R	S	設定値（先頭） 4文字 ×16¹, ×16⁰, ×16³, ×16²	設定値（最終） 4文字 ×16¹, ×16⁰, ×16³, ×16²	BCC ×16¹, ×16⁰	CR

（下位）（上位）　（下位）（上位）

（リードエラー）

% または <	送り元 ×10¹, ×10⁰	!	ｴﾗｰｺｰﾄﾞ ×16¹, ×16⁰	BCC ×16¹, ×16⁰	CR

［WS］設定値エリアライト

タイマ／カウンタの設定値を書き込みます。

■コマンド

% または <	送り先 ×10¹, ×10⁰	#	W	S	先頭ﾀｲﾏ/ｶｳﾝﾀNo. 4文字 ×10³, ×10², ×10¹, ×10⁰	最終ﾀｲﾏ/ｶｳﾝﾀNo. 4文字 ×10³, ×10², ×10¹, ×10⁰

書き込みﾃﾞｰﾀ（先頭） 4文字 ×16¹, ×16⁰, ×16³, ×16²	書き込みﾃﾞｰﾀ（最終） 4文字 ×16¹, ×16⁰, ×16³, ×16²	BCC ×16¹, ×16⁰	CR

（下位）（上位）　（下位）（上位）

■レスポンス（ライトOK）

% または <	送り元 ×10¹, ×10⁰	$	W	S	BCC ×16¹, ×16⁰	CR

（ライトエラー）

% または <	送り元 ×10¹, ×10⁰	!	ｴﾗｰｺｰﾄﾞ ×16¹, ×16⁰	BCC ×16¹, ×16⁰	CR

［RK］経過値エリアリード

タイマ／カウンタの経過値を読み出します。

■コマンド

% または <	送り先	#	R	K	先頭ﾀｲﾏ/ｶｳﾝﾀNo.	最終ﾀｲﾏ/ｶｳﾝﾀNo.	BCC	CR
	×10¹, ×10⁰				4文字 ×10³, ×10², ×10¹, ×10⁰	4文字 ×10³, ×10², ×10¹, ×10⁰	×16¹, ×16⁰	

■レスポンス（リードOK）

% または <	送り元	$	R	K	経過値（先頭）	経過値（最終）	BCC	CR
	×10¹, ×10⁰				4文字 ×16¹, ×16⁰, ×16³, ×16²	4文字 ×16¹, ×16⁰, ×16³, ×16²	×16¹, ×16⁰	
					(下位) (上位)	(下位) (上位)		

（リードエラー）

% または <	送り元	!	ｴﾗｰｺｰﾄﾞ	BCC	CR
	×10¹, ×10⁰		×16¹, ×16⁰	×16¹, ×16⁰	

［WK］経過値エリアライト

タイマ／カウンタの経過値を書き込みます。

■コマンド

% または <	送り先	#	W	K	先頭ﾀｲﾏ/ｶｳﾝﾀNo.	最終ﾀｲﾏ/ｶｳﾝﾀNo.	書き込みﾃﾞｰﾀ（先頭）	書き込みﾃﾞｰﾀ（最終）	BCC	CR
	×10¹, ×10⁰				4文字 ×10³, ×10², ×10¹, ×10⁰	4文字 ×10³, ×10², ×10¹, ×10⁰	4文字 ×16¹, ×16⁰, ×16³, ×16²	4文字 ×16¹, ×16⁰, ×16³, ×16²	×16¹, ×16⁰	
							(下位) (上位)	(下位) (上位)		

■レスポンス（ライトOK）

% または <	送り元	$	W	K	BCC	CR
	×10¹, ×10⁰				×16¹, ×16⁰	

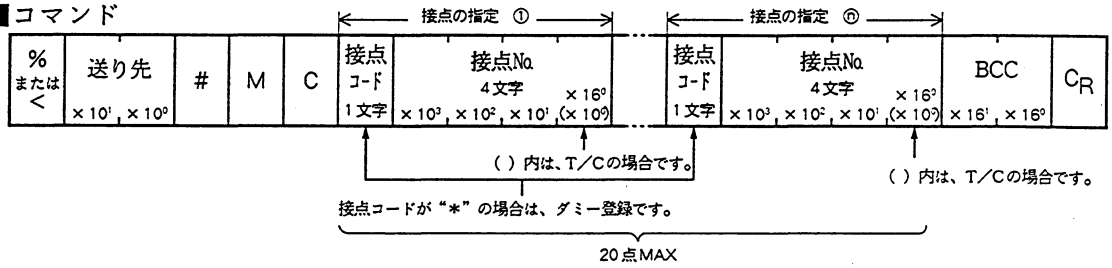
（ライトエラー）

% または <	送り元	!	ｴﾗｰｺｰﾄﾞ	BCC	CR
	×10¹, ×10⁰		×16¹, ×16⁰	×16¹, ×16⁰	

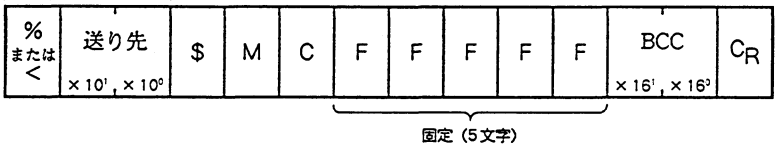
［MC］モニタ接点登録・リセット

モニタする接点を登録します。

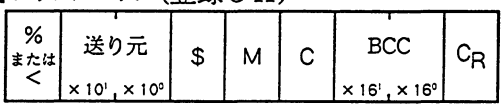
■コマンド



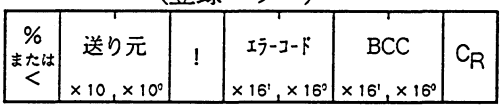
(登録リセット)



■レスポンス (登録OK)



(登録エラー)



接点コード

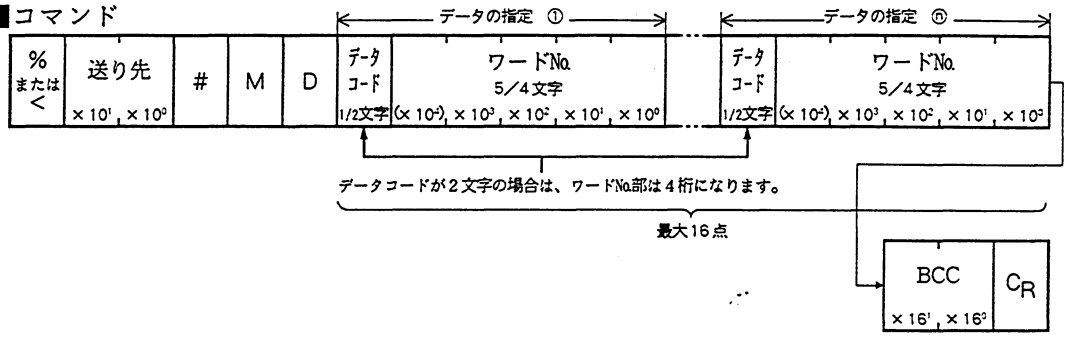
接点	表記
X	"X"
Y	"Y"
CR	"R"
LR	"L"
TM	"T"
CT	"C"

・登録個数は、1台80点までです。

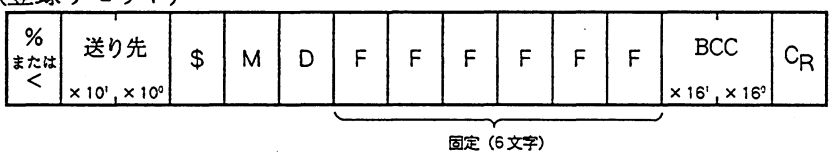
［MD］モニタデータ登録・登録リセット

モニタするデータを登録します。

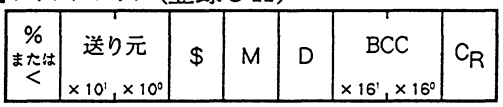
■コマンド



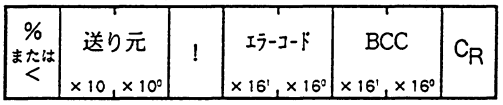
(登録リセット)



■レスポンス (登録OK)



(登録エラー)



データ種類	データコード
データレジスタ	D
リンクレジスタ	L
ファイルレジスタ	F
設定値	S
経過値	K
インデックスレジスタX	IX
インデックスレジスタY	IY
ワード外部入力	WX
ワード外部出力	WY
ワード内部リレー	WR
ワードリンクリレー	WL

データコード2文字

・登録個数は、1台あたり16点までです。

・モニターデータ登録には、ダミー登録（“＊”）はできません。

注意 ・データコードのうちIX,IY,つまり1文字目がIのものは、ワードNo.の4文字は0に設定してください。

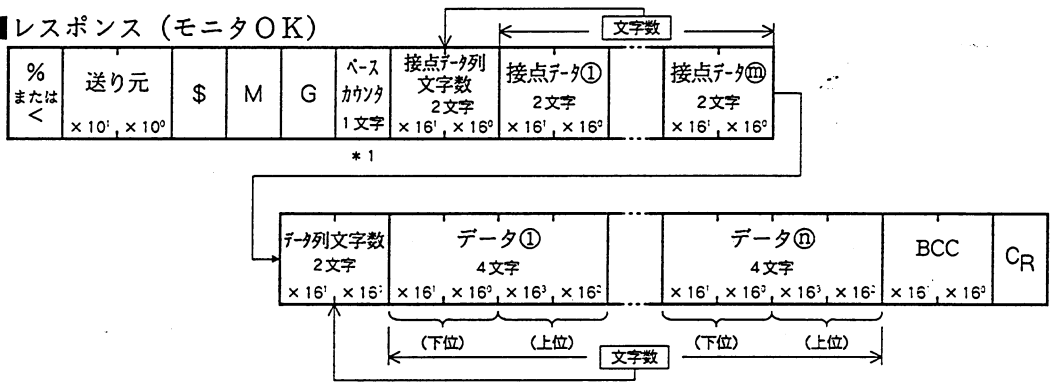
［MG］モニタ実行

登録した接点やデータをモニタします。

■コマンド

% または <	送り先	#	M	G	BCC	CR
	×10¹, ×10⁰				×16¹, ×16⁰	

■レスポンス（モニタOK）



＊1 ベースカウンタは、前回のレスポンスから今回のレスポンスまでのシーケンサのスキャン数が10以上の時は、“A”を返します。

（モニタエラー）

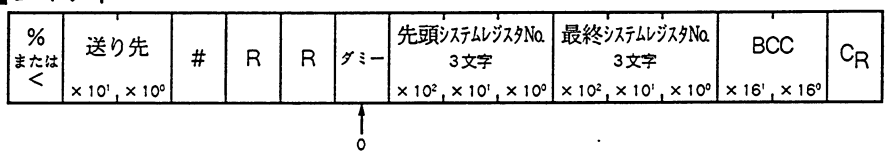
% または <	送り元	!	ｴﾗｰｺｰﾄﾞ	BCC	CR
	×10, ×10⁰		×16¹, ×16⁰	×16¹, ×16⁰	

・接点データは、接点データ①のbit 0より登録された順に入っています。
・データは、データ①より登録された順に入っています。

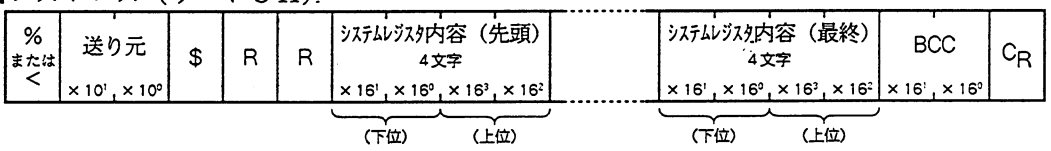
［RR］システムレジスタリード

システムレジスタ内容を読み出します。

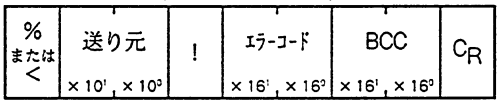
■コマンド



■レスポンス（リードOK）



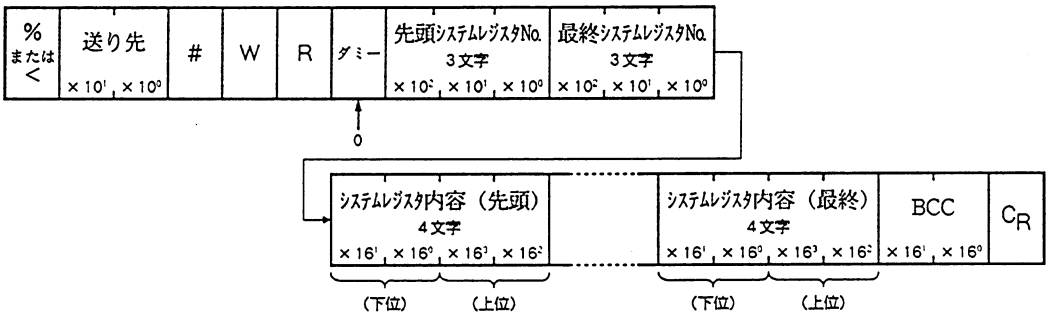
（リードエラー）



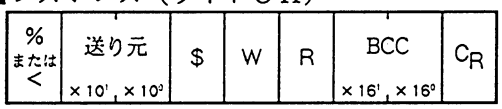
［WR］システムレジスタライト

システムレジスタを設定します。

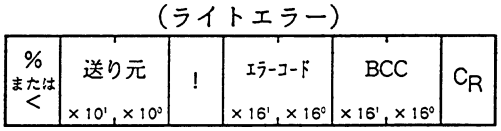
■コマンド



■レスポンス（ライトOK）



（ライトエラー）



[RT] PCステータスリード

PCの仕様、エラー発生時のエラーコードなどを読み出します。

■コマンド

% または <	送り先 ×10¹, ×10⁰	#	R	T	BCC ×16¹, ×16⁰	CR

■レスポンス（リードOK）

% または <	送り元 ×10¹, ×10⁰	$	R	T	機種ｺｰﾄﾞ 2文字 ×10¹, ×10⁰	ﾊﾞｰｼﾞｮﾝ 2文字 ×16¹, ×16⁰	ﾌﾟﾛｸﾞﾗﾑ容量 2文字 ×16¹, ×16⁰	動作ﾓｰﾄﾞ 2文字 ×16¹, ×16⁰

システム用リンク情報 2文字 ×16¹, ×16⁰	ｴﾗｰﾌﾗｸﾞ 2文字 ×16¹, ×16⁰	自己診断ｴﾗｰNo. 4文字 ×16¹, ×16⁰, ×16³, ×16²	BCC ×16¹, ×16⁰	CR

（下位）（上位）

（リードエラー）

% または <	送り元 ×10¹, ×10⁰	!	ｴﾗｰｺｰﾄﾞ ×16¹, ×16⁰	BCC ×16¹, ×16⁰	CR

■機種コード

CPUユニットの機種を10進数2文字で表わします。

コード	機　種
03	FP3 ラダータイプ （プログラム容量：10Kステップ仕様）
08	FP3H BASICタイプ （プログラム容量：128Kバイト）
09	FP3 BASICタイプ （プログラム容量：64Kバイト）
13	FP3 ラダータイプ （プログラム容量：16Kステップ仕様）
20	FP10／ FP10Sタイプ

■プログラム容量

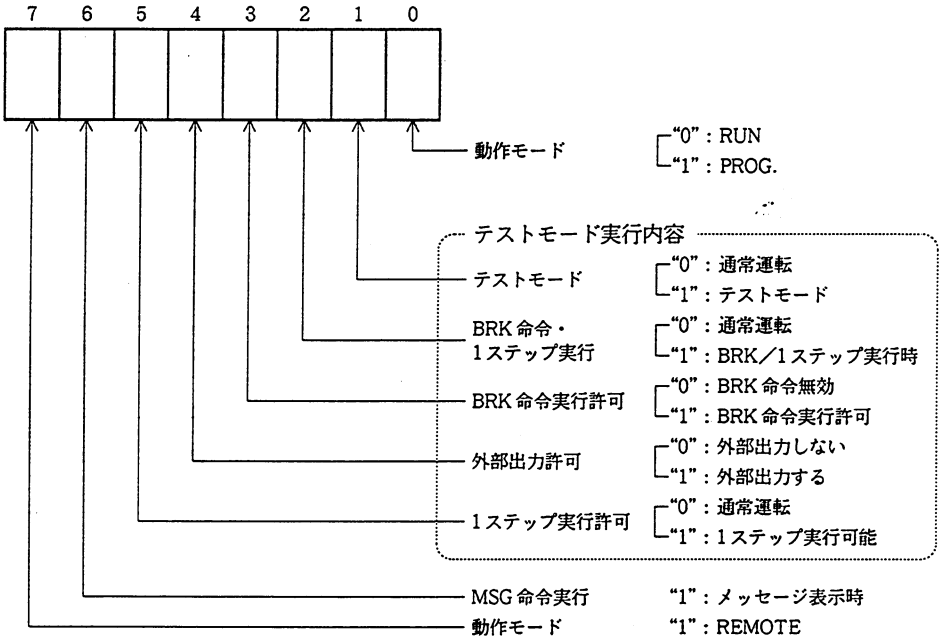
システムレジスタNo.0で設定しているプログラム容量を10進数2文字（偶数）で表わします。単位はKステップです。

コード	プログラム容量	最終ステップアドレス
02	2Kステップ	1,534
n		1,024 × n − 512 − 2 （例：n＝8の時、7678）
16	16Kステップ	15,870

注意 ・FP10、FP10Sの場合は、コードは“0”になります。

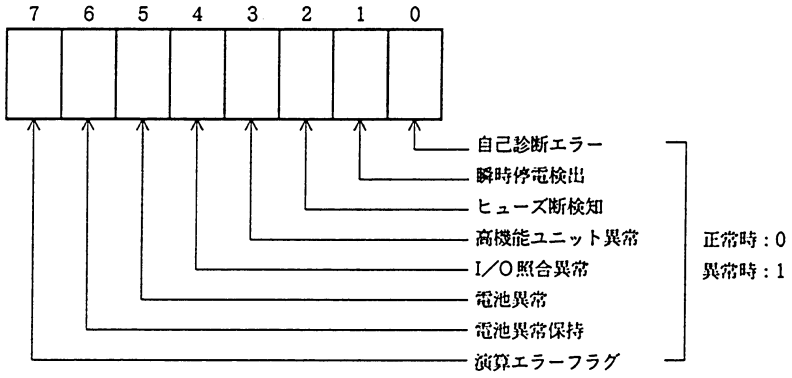
■動作モード

・特殊内部リレーR9020～R9027の内容を16進数2文字で表わします。
・CPUユニットのモード切り替えスイッチの設定（RUN／PROG.／REMOTE)、通常運転かテスト運転か等を確認することができます。
次のように2進数表記にして読みます。



■エラーフラグ

8つのエラーフラグ（特殊内部リレー)、R9000～R9007の状態を16進数2文字で表わします。
次のように2進数表記にして読みます。



■自己診断エラーコード

・エラー発生時の自己診断エラーコードを16進数4桁で表します。自己診断エラーコードは通常10進数で扱っていますので、ご注意ください。
例えば、読み出した内容が、16進数で“2B00”であれば、自己診断エラーコードは“2B”、10進数で“43”（演算渋滞）になります。
・エラーが発生していない場合は“0000”になります。

[RP] プログラムブロックリード

PCからシーケンスプログラムのデータを読み出します。読み出したデータは、フロッピーディスク等に保存したり、別のPCにWPコマンドを使用して書き込んだりすることができます。

> 読み出したプログラムデータを編集ソフトNPST－GRで呼び出すことはできませんので、ご注意ください。

■コマンド

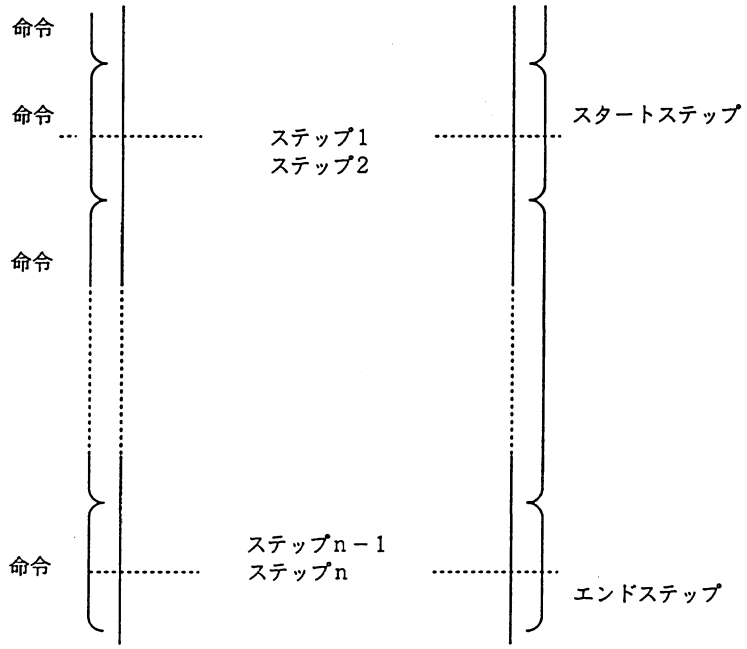
% または <	送り先	#	R	P	先頭ステップアドレス 5文字	最終ステップアドレス 5文字	BCC	CR
	×10¹, ×10⁰				×10⁴, ×10³, ×10², ×10¹, ×10⁰	×10⁴, ×10³, ×10², ×10¹, ×10⁰	×16¹, ×16⁰	

■レスポンス（リードOK）

% または <	送り元	$	R	P	先頭ステップデータ 4文字	最終ステップデータ 4文字	BCC	CR
	×10¹, ×10⁰				×16¹, ×16², ×16³, ×16⁰ (下位, 上位)	×16¹, ×16², ×16³, ×16⁰ (下位, 上位)	×16¹, ×16⁰	

（リードエラー）

% または <	送り元	!	ｴﾗｰｺｰﾄﾞ	BCC	CR
	×10¹, ×10⁰		×16¹, ×16⁰	×16¹, ×16⁰	



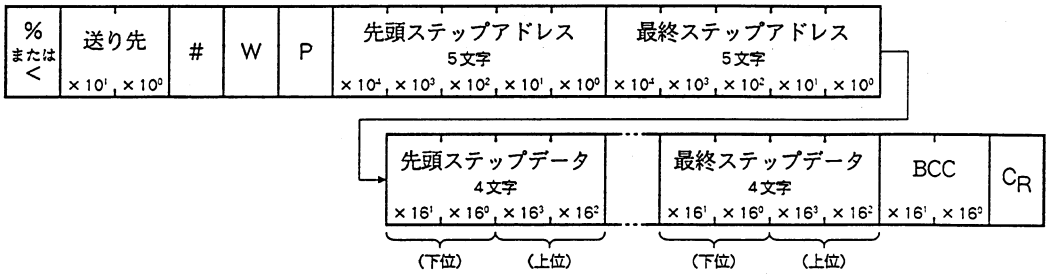
指定ステップにより命令の途中になることもあります.

[WP] プログラムブロックライト

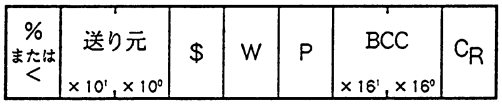
RPコマンドで読み出したシーケンスプログラムのデータを、PCに書き込みます。コンピュータ側のプログラムで自動的に書き込みますので、複数のプログラムを切り替える場合などに有効です。

＊プログラムの書き込みを行う時は、CPUユニットを「PROG.」モードにしてください。

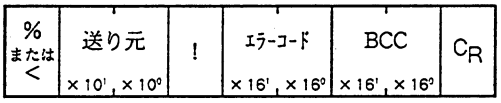
■コマンド



■レスポンス（ライトOK）



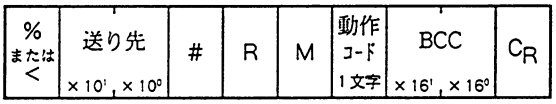
（ライトエラー）



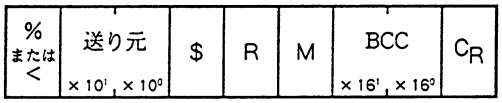
[RM] リモートコントロール

PCの動作モードを切り替えます。

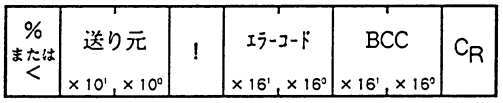
■コマンド



■レスポンス（リモコンOK）



（リモコンエラー）



■動作コード

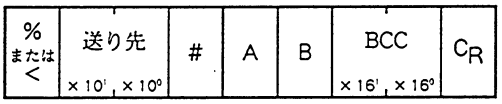
コード	動作
"R"	PROGRAMモード→RUNモード（起動）
"P"	RUNモード→PROGRAMモード（停止）

・PCのモードがREMOTEモードの場合のみ有効です

[AB] アボート

PCからの複数フレームのレスポンスの受信を途中で打ち切る時にコマンド送信側（コンピュータ）が発行します。

■コマンド



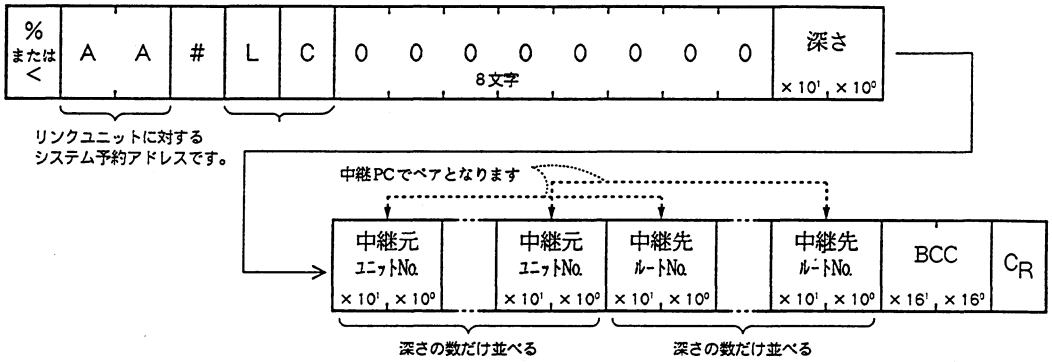
■レスポンス

無し

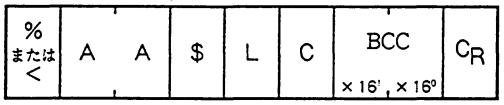
[LC] 伝送経路変更コマンド

接続する相手局が他の階層にある場合は、このコマンドで相手局までの経路を指定します。

■コマンド

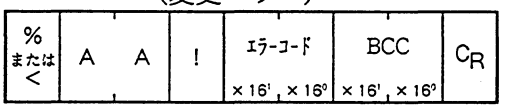


■レスポンス（変更ＯＫ）



項目名	設定値	内　容
深さ	00～03	接続するスレーブ局までの深さ
中継元ユニットNo.	01～64	中継元ユニットのユニットNo.
中継先ルートNo.	01～06	中継先ユニットのルートNo.

（変更エラー）



4－2 シリアル伝送制御コマンド

4-2-1 制御コマンドの概要

制御コマンドは、外部マスター機器からマスター局(リンクユニット)に対して動作状態の変更等を制御するためのコマンドで、スレーブ局の選択や、データの送受信要求等を行います。制御コマンドの送受信伝送手順はコンピュータリンクMEWTOCOL－COMと基本的に同じで、コマンドメッセージを外部マスター機器からマスター局に送信し、マスター局からのレスポンス(応答)メッセージを受信します。

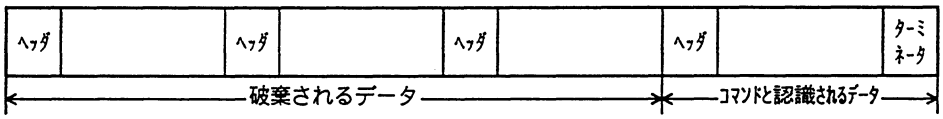
●制御コマンドは、外部マスター機器より、下記フォーマットでマスター局に送信します。

% または <	A	A	#	コマンドコード	パラメータ	BCC ブロックチェックコード ×16¹, ×16⁰	ターミネータ	ターミネータ

% または <	A	A	!	エラーコード ×16¹, ×16⁰	BCC ×16¹, ×16⁰	ターミネータ	ターミネータ

フォーマット内はすべてASCII文字で記述します。コマンドメッセージが正常にスレーブ局に送信された場合は、各コマンド特有のフォーマットのレスポンスメッセージが戻ります。なお、エラーが発生した場合は、上のようなエラーレスポンスメッセージが戻ります。

●マスター局は、常にヘッダを監視し、上記フォーマットのコマンドメッセージを認識した時点でコマンドに対応する動作を行います。ヘッダ～ターミネータの間に、下記のように複数のヘッダが存在する場合は、最後のヘッダ～ターミネータまでを制御コマンドとして処理し、それ以前のデータは破棄されます。



注意 ●ヘッダに使用するコードはヘッダ変更コマンドによって任意の文字に変更可能です。
参照「S7：制御コマンド・ヘッダコード変更」

●パラメータのサイズは、コマンドによって異なります。 参照「制御コマンドについて（下図)」

●BCC（ブロックチェックコード）はヘッダ～パラメータ（の最後の文字）までのキャラクタコードの排他的論理和（16進計算）をASCIIコードの2文字に変換したものです。
参照「BCC(ブロックチェックコード)」(P.250)

●ターミネータは多重接続マスター局で設定しているコードを使用します。ターミネータに使用するコードは制御コマンドによって変更可能です。ただし、メッセージフォーマット内で既に使用した文字と重複しないようにしてください。 参照「S6：マスター局ターミネータ変更」

■制御コマンドについて

シリアル伝送機能（多重接続モード）において、使用できるコマンドは、以下のとおりです。

●制御コマンド一覧

コマンドコード	内容	パラメータ数	パラメータサイズ
S1	スレーブ局接続（指定）コマンド	1	2バイト
S2	接続スレーブ局への1ブロック・データ受信要求コマンド	0	0
S3	内部情報（送／受信バッファ、ステータス領域）クリア要求コマンド	2	2
S4	スレーブ局シリアルポートRTS信号制御コマンド	2	2
S5	RS232Cインターフェイス動作状態ステータス読み出しコマンド	2	3

●その他関連コマンド一覧

コマンドコード	内容	パラメータ数	パラメータサイズ
LC	階層コントロールコマンド（相手局の階層指定用）	5～11	10～22

注意 制御コマンドは、多重接続モードのマスター局にのみ有効です。

4-2-2 制御コマンドのフォーマット

[S1] スレーブ局接続（指定）コマンド

送受信の対象となるスレーブ局を指定（変更）するコマンドです。このコマンドによりスレーブ局へのアクセスが可能になります。

■コマンド

ヘッダ	A	A	#	S	1	接続スレーブ局 ×10¹, ×10⁰	BCC ×16¹, ×16⁰	ターミネータ	ターミネータ

■レスポンス 正常時

ヘッダ	A	A	$	S	1	受信データ数 ×16¹, ×16⁰	RTS信号状態 ×16¹, ×16⁰	CTS信号状態 ×16¹, ×16⁰	BCC ×16¹, ×16⁰	ターミネータ	ターミネータ

異常時

ヘッダ	A	A	!	エラーコード ×16¹, ×16⁰	BCC ×16¹, ×16⁰	ターミネータ	ターミネータ

■説明

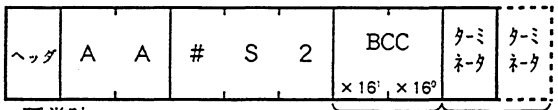
データ名称	データ内容	表記
接続スレーブ局	01～64 シリアル多重接続スレーブ局指定を行っている局の局番を設定します。	BCD
受信データ数	指定スレーブ局のRS232C受信用バッファ内データブロック数。	HEX
RTS信号状態	01＝外部機器送信可能　／ 01≠送信不可能 （指定スレーブ局のRTS信号状態）	
CTS信号状態	00＝送信不可能　／ 01＝送信可能 （外部機器への送信可能状態）	

注意
- 指定局がシリアル多重スレーブモードで動作していない場合、およびスレーブ局で動作中でもそのマスター局が異なる場合は、接続できません。
- 現在接続中のスレーブ局に対して、送信データがユニット内に残っている場合、他のスレーブ局への接続変更はできません。強制的に変更する場合は、一旦マスター局の送受信バッファをクリア（参照「S3」コマンド）してから、接続変更してください。

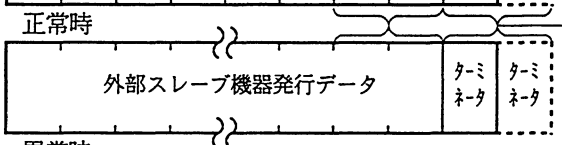
[S2] 接続スレーブ局への1ブロック・データ受信要求

スレーブ局が外部機器から受信したデータを、マスター局に送信するよう要求します。

■コマンド



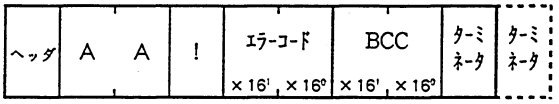
■レスポンス 正常時



このターミネータは、外部マスター機器に対応したコードです。

← 他のレスポンスとフォーマットが異なりますのでご注意ください。

異常時



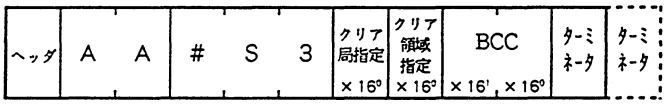
注意 ・このコマンドを実行する前にS1コマンドで、スレーブ局接続処理を行ってください。

・このコマンドを発行した接続スレーブ局側に受信データがある場合のレスポンスは、上記正常レスポンスのように、外部スレーブ機器が発行したデータが返ってきます。

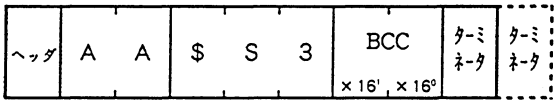
[S3] 領域クリア要求コマンド

指定局の送受信バッファをクリアします。

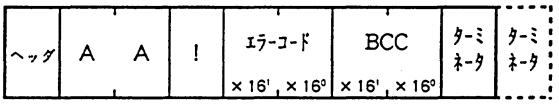
■コマンド



■レスポンス 正常時



異常時



■説明

データ名称	データ内容	表記
クリア局指定	0＝マスター局＋そのマスター局に従属する全スレーブ局 1＝マスター局 2＝接続スレーブ局（S1コマンドによって指定された局のみ） 3＝そのマスター局に従属する全てのスレーブ局	HEX
クリア領域指定	0＝RS232C用送受信バッファ＋エラーステータス領域 1＝RS232C用受信バッファ 2＝RS232C用送信バッファ 3＝エラーステータス領域（S5ｺﾏﾝﾄﾞの「読み出しﾃﾞｰﾀ指定3（ｼﾘｱﾙﾎﾟｰﾄ異常状態）」の領域）	

注意 ・接続スレーブ局指定時は、接続スレーブ局からの実行結果がレスポンスに反映されます。

・マスター局に従属する全スレーブ局を設定した場合は、各スレーブ局の実行結果は反映されません。

[S4] スレーブ局シリアルポートRTS信号制御コマンド

接続スレーブ局またはマスター局に従属する全スレーブ局の外部機器との通信制御信号RTSを制御します。

■コマンド

ヘッダ	A	A	#	S	4	対象スレーブ局指定 ×16⁰	RTS信号制御指定 ×16⁰	BCC ×16¹, ×16⁰	ターミネータ	ターミネータ

■レスポンス 正常時

ヘッダ	A	A	$	S	4	BCC ×16¹, ×16⁰	ターミネータ	ターミネータ

異常時

ヘッダ	A	A	!	エラーコード ×16¹, ×16⁰	BCC ×16¹, ×16⁰	ターミネータ	ターミネータ

■説明

データ名称	データ内容	表記
対象スレーブ局指定	0＝接続スレーブ局（S1コマンドによって指定された局のみ） 1＝マスター局に従属する全スレーブ局	HEX
RTS信号制御指定	0＝外部機器送信禁止 1＝外部機器送信許可（ただし外部機器送信許可の場合でも、スレーブの受信バッファがフルの状態では、RTS信号は禁止状態になります）	

注意
- 接続スレーブ局指定時は、接続スレーブ局からの実行結果がレスポンスに反映されます。
- マスター局に従属する全スレーブ局を設定した場合は、各スレーブ局の実行結果は反映されません
- このコマンドは、RTS制御を有効に設定した場合（MEWNET－Hシステム設定ソフトで設定します）にのみ使用できます。

[S5] シリアルポート動作状態ステータス読み出し

相手局のモードに関係なく（コンピュータリンクモードも含む）任意のリンクユニットのシリアルポート動作状態ステータスを読み出します。

■コマンド

ヘッダ	A	A	#	S	5	読出局No. ×10¹, ×10⁰	読出データ指定 ×16⁰	BCC ×16¹, ×16⁰	ターミネータ	ターミネータ

■レスポンス 正常時

ヘッダ	A	A	$	S	5	読出ステータス・データ	BCC ×16¹, ×16⁰	ターミネータ	ターミネータ

「読出ステータス・データ」の「読出データ指定」の値により異なります。「□レスポンス詳細」を参照してください。

異常時

ヘッダ	A	A	!	エラーコード ×16¹, ×16⁰	BCC ×16¹, ×16⁰	ターミネータ	ターミネータ

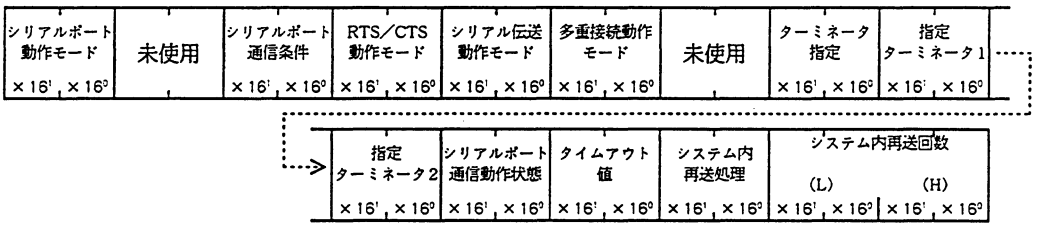
■説明

また、本コマンドを実行する前に、S1コマンドによって指定した接続スレーブ局の指定は、本コマンドを実行後も変更されません。

データ名称	データ内容	表記
読出局No.	00＝マスター局（マスター局のユニットNo.を指定しても読み出せます） nn＝各局のNo.(01～64)	BCD
読出データ指定	0＝シリアルポート動作モード設定状態 1＝データ送信先、送信元設定状態 2＝シリアルポート受信状態 3＝シリアルポート異常状態	HEX

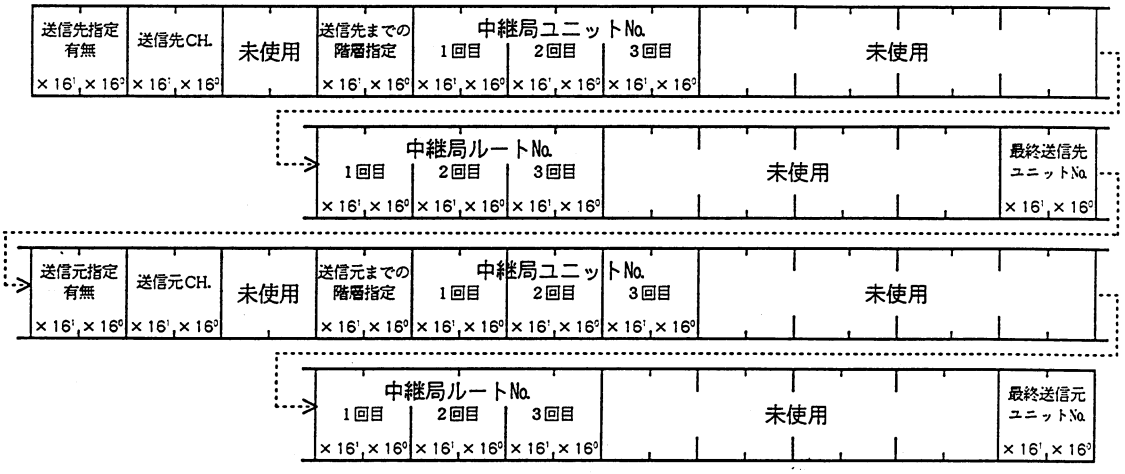
□レスポンス詳細

読み出しデータ指定「0（シリアルポート動作モード設定状態）」の場合の読み出しステータスデータ



データ名称	データ内容	表記
ｼﾘｱﾙﾎﾟｰﾄ動作ﾓｰﾄﾞ	00：コンピュータリンクモード　／01：シリアル伝送モード	HEX
シリアルポート通信条件	7 6 5 4 3 2 1 0 001＝未使用／100＝4,800／111＝600 010＝19,200／101＝2,400　（単位bps） 011＝ 9,600／110＝1,200 伝送速度 データ長：　0＝7bit　／　1＝8bit ﾊﾟﾘﾃｨﾁｪｯｸ：　0＝無し　／　1＝有り パリティ：　0＝奇数　／　1＝偶数 ｽﾄｯﾌﾟﾋﾞｯﾄ長：　0＝1bit　／　1＝2bit	HEX ↑ 左記のビット構成をHEX換算する
RTS/CTS制御指定	00＝制御しない　／01＝制御する	HEX
シリアル伝送動作モード	00＝単一接続モード　／01=多重接続モード (ｼﾘｱﾙ伝送ﾓｰﾄﾞでのみ有効)	
多重接続動作モード	00＝スレーブ局　／01=マスター局 (ｼﾘｱﾙ伝送ﾓｰﾄﾞでのみ有効)　注 単一モード時は00固定	
ﾀｰﾐﾈｰﾀ指定	00＝$^{C}_{R}$　／01＝$^{L}_{F}$　／02＝$^{C}_{R}+^{L}_{F}$ 03＝$^{L}_{F}+^{C}_{R}$　／04＝1文字任意指定／05＝2文字任意指定	
指定ﾀｰﾐﾈｰﾀ1	00～FFH（データ長が7の場合は00～7FH） （ターミネータ指定が04・05の場合に有効）	
指定ﾀｰﾐﾈｰﾀ2	00～FFH（データ長が7の場合は00～7FH） （ターミネータ指定が05の場合に有効）	
ｼﾘｱﾙﾎﾟｰﾄ通信動作状態	00=停止中　／01=起動中	
タイムアウト値	01～FFH (0.5s単位 1＝0.5s、255＝127.5s)	
システム内再送処理	00=再送しない　／01=再送する （シリアル伝送モードでのみ有効）	
システム内再送回数	0001～FFFFH＝再送回数　／0000H=無限 (ｼﾘｱﾙ伝送ﾓｰﾄﾞで、しかも再送処理が 1:再送するの場合に有効)　初期値=3	

読み出しデータ指定「1（データ送信先、送信元設定状態）」の場合の読み出しステータスデータ



データ名称	データ内容		表記
送信先指定有／無	00H＝指定無し　／01H＝指定有り		HEX
送信先CH.	07H＝シリアルポート　／0FH＝パソコン		
送信先までの階層指定	00H＝指定無し　／01H～03H：指定有り*1		
中継局ユニットNo.	01H～40H（64）：中継局のユニットNo.	送信先までの階層指定値分有効	
中継局ルートNo.	01H～06H：中継リンクユニットの装着位置		
最終送信先ユニットNo.	01H～40H（64）：最終の送信先ユニットNo.		
送信元指定有／無	00H＝指定無し　／01H＝指定有り		
送信元CH.	07H＝シリアルポート　／0FH＝パソコン		
送信元までの階層指定	00H＝指定無し　／01H～03H：指定有り*1		
中継局ユニットNo.	01H～40H（64）：中継局のユニットNo.	送信元までの階層指定数分有効	
中継局ルートNo.	01H～06H：中継リンクユニットの装着位置		
最終送信元ユニットNo.	01H～40H（64）：最終の送信元ユニットNo.		

送信先指定有／無　が「指定有り」の場合に有効

送信元指定有／無　が「指定有り」の場合に有効

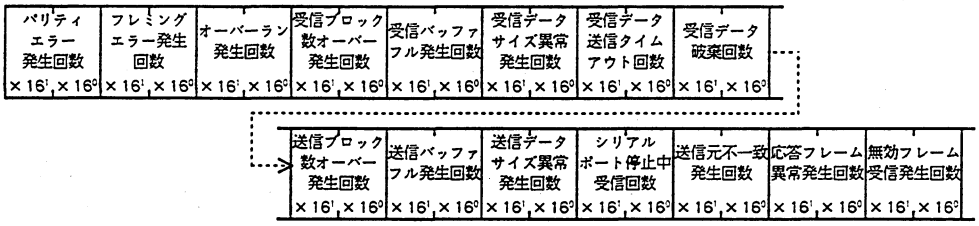
＊1 数値は中継局数を表します。

読み出しデータ指定「2（シリアルポート受信状態）」の場合の読み出しステータスデータ

データ名称	データ内容	表記
受信データ数	00～FFH 現在受信バッファに格納されている受信ブロック数	HEX
RTS信号状態	7 6 5 4 3 2 1 0 0 0 0 0 0 0＝　外部機器送信不可状態 1＝　外部機器送信可能状態 外部機器送信状態＊1 外部機器送信不可要因（「0」の場合はその要因に拠らない） 1＝ RS232C受信バッファFULLにより送信禁止 1＝ マスター局からのS4コマンドにより送信禁止（スレーブ局のみ） 1＝ シリアル伝送モードにおけるパラメータ未設定状態	HEX ↑ 左記のビット構成をHEX換算する
CTS信号状態	0＝　外部接続機器への送信禁止状態 1＝　外部接続機器への送信許可状態	HEX

＊1 RTS／CTS制御をしない場合は常時「1」になります。

読み出しデータ指定「3（シリアルポート異常状態）」の場合の読み出しステータスデータ

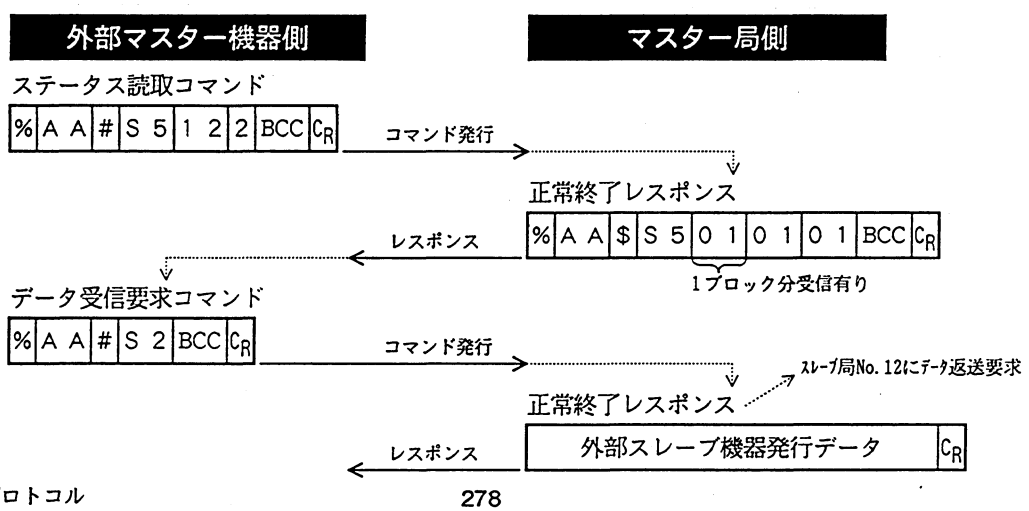


データ名称	データ内容	表記
パリティｴﾗｰ発生回数	00～FFH パリティ不一致発生回数	HEX
フレーミングｴﾗｰ発生回数	00～FFH ストップビット検出不可回数	
オーバーランｴﾗｰ発生回数	00～FFH 受信データオーバーラン回数	
受信ﾌﾞﾛｯｸ数オーバー回数	00～FFH RS232C受信用バッファのブロック数がオーバーした回数	
受信バッファフル発生回数	00～FFH RS232C受信用バッファがフル状態になった回数	
受信ﾃﾞｰﾀｻｲｽﾞ異常回数	00～FFH 受信データサイズが2Kバイトを越えるデータを外部機器より受信した回数	
受信ﾃﾞｰﾀ送信ﾀｲﾑｱｳﾄ回数	00～FFH RS232C受信用バッファ内データを他局へ送信したが規定時間内に応答がなかった回数	
受信データ破棄回数	00～FFH 指定回数の再送を行っても他局からの応答がなかったため、データを破棄した回数	
送信ﾌﾞﾛｯｸ数オーバー回数	00～FFH RS232C送信用バッファのブロック数がオーバーした回数	
送信バッファフル発生回数	00～FFH RS232C送信用バッファがフル状態になった回数	
送信ﾃﾞｰﾀｻｲｽﾞ異常回数	00～FFH 他局から受信したデータサイズに自局外部機器用ターミネータ数を加算したところ、2Kバイトを越えた回数	
ｼﾘｱﾙﾎﾟｰﾄ停止中受信回数	00～FFH シリアルポート動作停止中に他局からのシリアルポート向けのデータを受信した回数（受信データは破棄）	
送信元不一致発生回数	00～FFH 送信元が指定局でない。または、モードの異なるデータを受信した回数*1	
応答フレーム異常発生回数	00～FFH ユニット間伝送に対する応答フレーム異常の発生回数	
無効フレーム受信発生回数	00～FFH 対応していないフレームを他局から受信した回数	

＊1 例）シリアルモード時にコンピュータリンクモードのデータを他局から受信した等

■使用例

スレーブ局No.12の受信バッファ状態を読み出し、受信データがあれば、データ受信要求を行う。

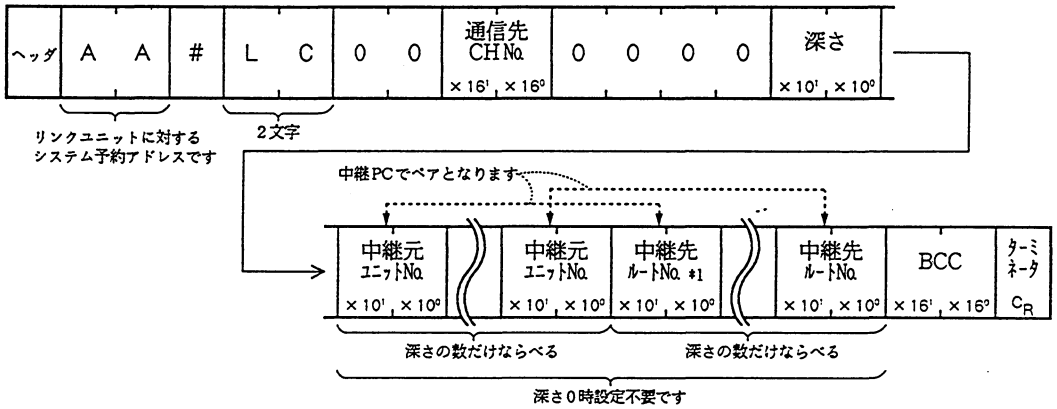


その他の関連コマンドのフォーマット

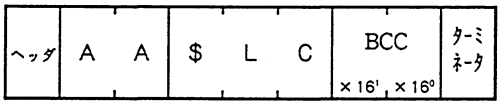
[LC] 階層コントロールコマンド

接続する相手局が他の階層にある場合は、このコマンドで、相手局までの経路を指定します。

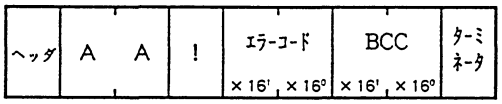
■コマンド



■レスポンス　正常時



異常時



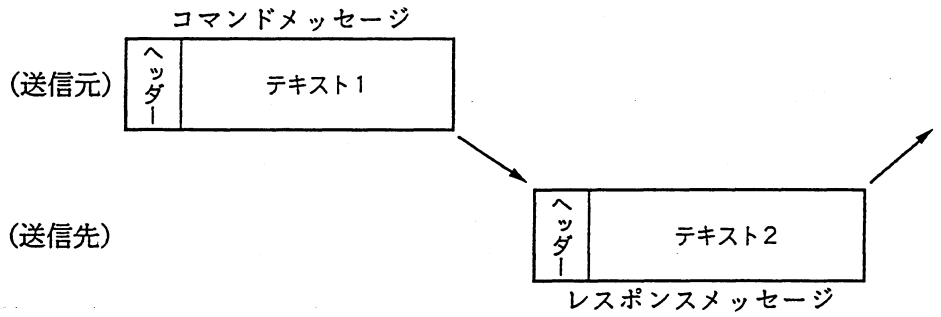
■説明

データ名称	データ内容	表記
送信先CH No.	7：相手局のRS232Cインターフェース：シリアル伝送機能 F： その他	HEX
深さ	00～03：接続するスレーブ局までの深さ	BCD
中継元ユニットNo.	01～64：中継元のユニットNo.	
中継先ルートNo.	01～06：中継先のルートNo.	

4 － 3 MEWTOCOL-DAT(データ転送)

4-3-1 MEWTOCOL-DATの概要

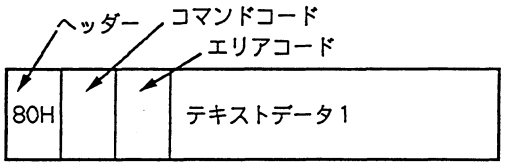
■コマンド／レスポンスの概要



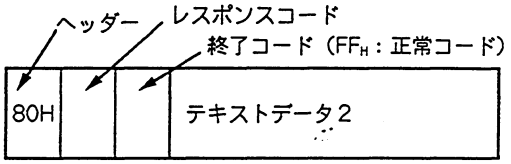
注意
- 専用手順、会話形になっています。
- バイナリコード送りです。
- コマンドメッセージを送信するごとに送信権を移行します。
- MEWNET－Hでは、テキストデータの長さは最大1020ワードです。
- 送信元がPCの場合、SEND（送信）命令、RECV（受信）命令の実行によりコマンド・メッセージが送信されます。

■コマンドコードとレスポンスコード

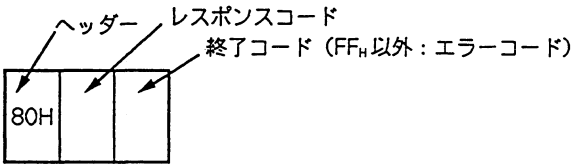
コマンドメッセージ



レスポンスメッセージ（正常時）



レスポンスメッセージ（異常時）



コマンドコード	内容説明	対応レスポンスコード
50H	データエリアの書き込み	D0H
51H	データエリアの読み出し	D1H
52H	接点情報の書き込み	D2H
53H	接点情報の読み出し	D3H

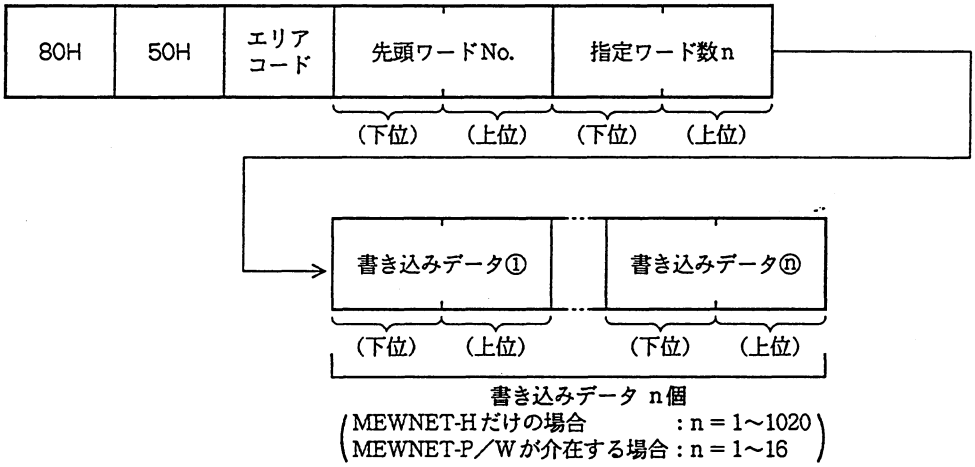
注意
- 対応レスポンスコードは、コマンドコード(1バイトバイナリ)の先頭ビットを反転(0→1)した値です。
- エリアコードについては、次ページ以降の「MEWTOCOL-DATコマンドリファレンス」を参照してください。
- 正常レスポンス時の終了コードはFFHです。異常時の終了コードはエラーコードです(「4-4 プロトコル・エラーコード」参照)。

4-3-2 MEWTOCOL-DATコマンドリファレンス

[50H] データエリアライト

データエリアの指定先頭ワードNo.より指定ワード数分のデータ書き込みを行ないます。

■コマンド



■レスポンス

正常時（ライトOK）

80H	D0H	FFH

異常時（ライトエラー）

80H	D0H	エラーコード

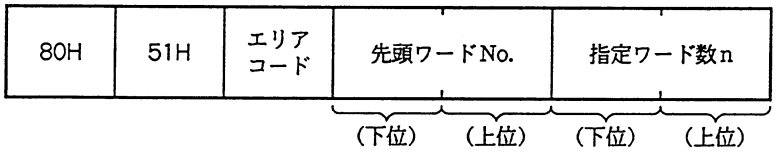
エリアコード

エリアの種類	エリアコード
リンクリレー（WL）	00
内部リレー（WR）	01
外部出力リレー（WY）	02
外部入力リレー（WX）	03
タイマ／カウンタ設定値（SV）	04
タイマ／カウンタ経過値（EV）	05
リンクレジスタ（LD）	06
特殊リレー（WR）	07
特殊データレジスタ（DT）	08
データレジスタ（DT）	09
ファイルレジスタ（FL）	0A

[51H] データエリアリード

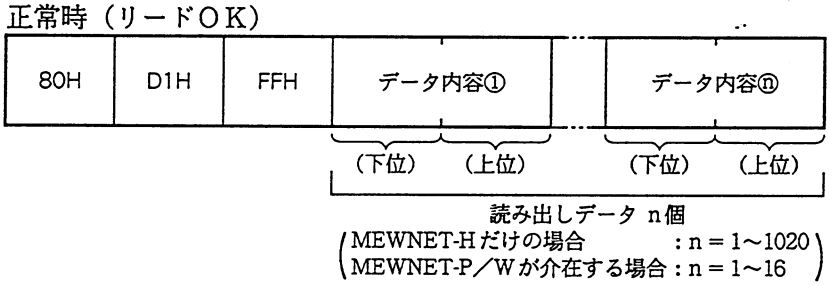
データエリアの指定先頭ワードNo.より指定ワード数分のデータ読み出しを行ないます。

■コマンド

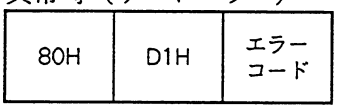


■レスポンス

正常時（リードOK）



異常時（リードエラー）



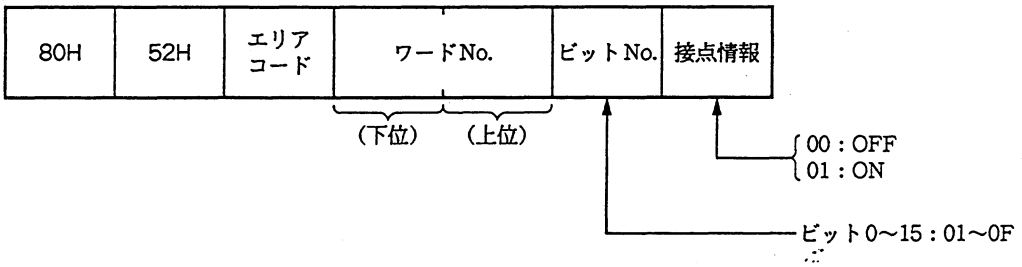
エリアコード

エリアの種類	エリアコード
リンクリレー（WL）	00
内部リレー（WR）	01
外部出力リレー（WY）	02
外部入力リレー（WX）	03
タイマ／カウンタ設定値（SV）	04
タイマ／カウンタ経過値（EV）	05
リンクレジスタ（LD）	06
特殊リレー（WR）	07
特殊データレジスタ（DT）	08
データレジスタ（DT）	09
ファイルレジスタ（FL）	0A

[52H] 接点情報のライト

接点エリアの指定接点に対する書き込みを行ないます。

■コマンド



■レスポンス

正常時（ライトOK）

80H	D2H	FFH

異常時（ライトエラー）

80H	D2H	エラーコード

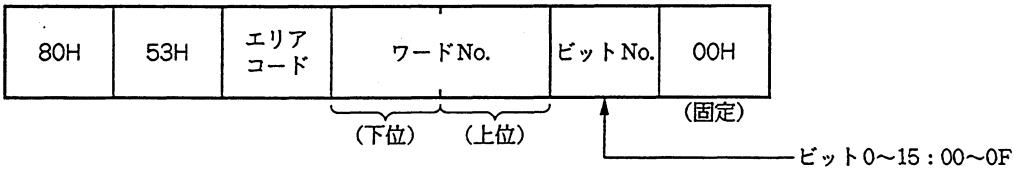
エリアコード

エリアの種類	エリアコード
リンクリレー（WL）	00
内部リレー（WR）	01
外部出力リレー（WY）	02
外部入力リレー（WX）	03
タイマ／カウンタ設定値（SV）	04
タイマ／カウンタ経過値（EV）	05
リンクレジスタ（LD）	06
特殊リレー（WR）	07
特殊データレジスタ（DT）	08
データレジスタ（DT）	09
ファイルレジスタ（FL）	0A

[53H] 接点情報のリード

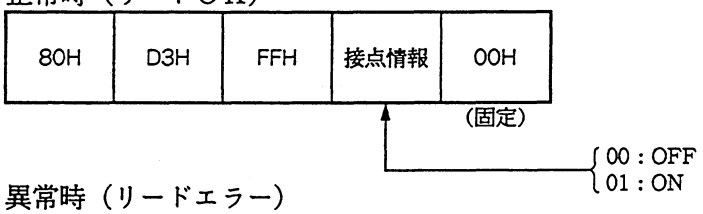
接点エリアの指定接点の読み出しを行ないます。

■コマンド



■レスポンス

正常時（リードOK）



異常時（リードエラー）

エリアコード

エリアの種類	エリアコード
リンクリレー（WL）	00
内部リレー（WR）	01
外部出力リレー（WY）	02
外部入力リレー（WX）	03
タイマ／カウンタ設定値（SV）	04
タイマ／カウンタ経過値（EV）	05
リンクレジスタ（LD）	06
特殊リレー（WR）	07
特殊データレジスタ（DT）	08
データレジスタ（DT）	09
ファイルレジスタ（FL）	0A

4－4 プロトコル・エラーコード

コンピュータリンク機能／データ転送機能／シリアル伝送機能を実行した場合に発生するエラーコードの内容と処置について説明しています。

■エラーコードの出力形態

各機能実行時のエラーコードは、各機能において下記の様に出力されます。

●コンピュータリンク機能／シリアル伝送機能（多重接続モード）の場合

MEWTOCOL－COMの通信手順でのエラーレスポンスメッセージのエラーコード部に出力されます。

コマンドメッセージ

% または <	送り先 ×10¹, ×10⁰	#	

エラー
レスポンスメッセージ

% または <	送り元 ×10¹, ×10⁰	!	ｴﾗｰｺｰﾄﾞ ×16¹, ×16⁰	BCC ×16¹, ×16⁰	CR

エラーコード
（エラーコードを2文字のASCIIコードに変換して出力します。）

●データ転送機能の場合

データ転送命令の命令完了コード格納用の特殊データレジスタに出力されます。

FP3ラダーCPU／FP3 BASIC CPUの場合

特殊データレジスタ DT9039

FP10／FP10Sラダー CPUの場合

特殊データレジスタ DT90039

■エラーコードと内容

エラーコードの内容について説明します。
エラーコードは、コンピュータリンク機能／データ転送機能／シリアル伝送機能（多重接続モード）で共通のコードになっています。
エラーコードの内容に応じた処理を行って下さい。

エラーコード一覧表

リンク系エラー

エラーコード	エラー名	エラー内容と処置
22	WACKエラー	相手先ユニットの受信バッファがオーバーフローしました。 <処置> 「MEWNET-Hリンクボード導入マニュアル」の「1-4 各種機能の使用上の制約」の範囲内にて御使用ください。
23	ユニットNo.重複	自局のユニットNo.が、他局と重複設定の状態のため、送信停止中。 <処置> 「MEWNET-Hリンクボード導入マニュアル」の「4-1 ネットワークへの接続」に従い、ユニットNo.の設定変更を実施してください。
25	リンクユニットのハードエラー	通信制御部のハードウェア異常 <処置> 電源をOFFにして再度電源ONにて確認してください。 復帰しない場合　不良ユニットを交換してください。 復帰する場合　ノイズによる誤動作が考えられます。 同軸ケーブルの布設状況、使用環境を御確認してください。
26	ユニットNo.設定異常	自局のユニットNo.が01～64以外のNo.に設定にされています。 <処置> 「MEWNET-Hリンクボード導入マニュアル」の「4-1 ネットワークへの接続」に従い、ユニットNo.を1～64の範囲で設定してください。
27	NOTサポートエラー	システムでサポートしていないパケットタイプを伝送しようとしました。 <処置> 弊社にお問い合わせください。
28	無応答エラー	相手局からの応答待ちタイムアウトエラー <処置> アプリケーションプログラムにて再送処理を実施してください
29	バッファクローズエラー	<MEWNET－Hリンクボード使用時>バッファクローズ状態で送受信を実行しました。 <処置> 相手側（MEWNET－Hリンクボード）の通信チャンネルに対してオープン処理を実施してください。
30	タイムアウト	送信不可能な状態が続いています。 <処置> アプリケーションプログラムにて再送処理を実施してください

エラーコード	エラー名	エラー内容と処置
32	転送不可エラー	自局のバッファがオーバーフロー状態のため、送信を中断しました。 <処置> 「MEWNET-Hリンクボード導入マニュアル」の「1-4 各種機能の使用上の制約」の範囲内にて使用してください。
33	通信停止	ネットワーク加入スイッチがOFFのため、送信を中断しました。 <処置> ネットワーク加入スイッチをONにしてください。
36	送信先存在せず	・ネットワーク上に相手局が存在していません。 ・ネットワーク加入より離脱しました <処置> ・相手局が存在しているかどうか確認してください。 ・アプリケーションプログラムにて再送等を実施してください
38	その他の通信異常	上記以外の伝送系異常が考えられます。 <処置> アプリケーションプログラムにて再送等を実施してください。

注意 ①多階層リンクを使用して2階層以上で発生した場合は、レスポンスは戻ってきません。
②基本手順エラー、処理エラー、PCアプリケーションエラーは、ネットワーク内で（階層含む）リンク系エラーが発生した場合は、レスポンスは戻ってきません。

基本手順エラー

エラーコード	エラー名	エラー内容
40	BCCエラー	<コンピュータリンク機能／シリアル伝送機能> コマンドのデータにBCCエラーが発生しました。
41	フォーマットエラー	<コンピュータリンク機能／シリアル伝送機能> 伝送フォーマットに合わないコマンドメッセージを送っています。 ・コマンドデータに過不足がある。 ・“#”・“送り先”がない。等 <データ転送機能> ・伝送可能な容量以上を送信しようとしました。
42	NOTサポートエラー	<コンピュータリンク機能／シリアル伝送機能> サポートされていないコマンドを送っいます。 サポートされていない送り先へコマンドを送っています。等
43	手順エラー	<コンピュータリンク機能／シリアル伝送機能> 送信要求メッセージ待ち状態（PC）のときにそれ以外のコマンドを送っています。

処理系エラー

エラーコード	エラー名	エラー内容と処置
50	リンク設定エラー	<コンピュータリンク機能> 存在しないルートNo.を指定しています。
51	同時操作エラー	<コンピュータリンク機能> 他機へのコマンドを転送（送信）時に、自機の送信バッファがオーバーフローしました。
52	送信不能エラー	<コンピュータリンク機能> 他機に対する送信ができない（リンクボードの異常等）。
53	ビジーエラー	<コンピュータリンク機能> 複数フレーム処理中にコマンドを受信しました。

PCアプリケーションエラー

エラーコード	エラー名	エラー内容と処置
60	パラメータエラー	<コンピュータリンク機能／シリアル伝送機能> エリア指定パラメータが存在しないコードまたは、そのコマンドでは使用できないコードになっています。(X,Y,D ••• etc.) 機能指定パラメータ (0,1,2 etc.) が不適当なコードになっています。
61	データエラー	<コンピュータリンク機能／シリアル伝送機能> 接点No.、エリアNo.、取り扱いデータのコード形式 (BCD,HEX etc.) 超過、不足、範囲指定エラーが発生しています。 <データ転送機能使用時> 自局、他局の領域指定誤りが発生しています。
62	登録エラー	<コンピュータリンク機能／シリアル伝送機能> 登録数をオーバーしているか、未登録状態で操作している。 (モニタ登録、トレース登録等) ••• 登録オーバー時は、登録リセットをしてください。
63	モードエラー	<コンピュータリンク機能／シリアル伝送機能> コマンドを送信したときの動作モードが、そのコマンドを処理できないモードになっています。
65	プロテクトエラー	<コンピュータリンク機能／シリアル伝送機能> メモリプロテクト状態でプログラムエリアまたは、システムレジスタに書き込み動作を実行しました。
66	アドレスエラー	<コンピュータリンク機能／シリアル伝送機能> アドレス（プログラムアドレス、絶対アドレス etc.）データのコード形式 (BCD,HEX etc.) 超過、不足、範囲指定エラーです。
67	データ無しエラー	<コンピュータリンク機能／シリアル伝送機能> 読み出しデータが存在しない。 (コメントの登録等が書き込まれていないものを読み出した場合等)
72	タイムアウトエラー	<データ転送機能> 送信アンサー待ちタイムアウトエラー
73	タイムアウトエラー	<データ転送機能> 送信バッファ空待ちタイムアウトエラー
74	タイムアウトエラー	<データ転送機能> レスポンス待ちタイムアウトエラー
92	送信先切替えエラー	<シリアル伝送機能（マスター局のみ）> 前回指定した局に対するデータの送信中です。 送信先の切替えはできません

改訂履歴

マニュアル番号は、表紙下に記載されています。

発行日付	マニュアル番号	改訂内容
1994年 1月	FAF−167	初版
1995年 2月	FAF−167①	2版　誤記訂正 ・富士通FMRシリーズ用リンクボード(AFP6762)のロータリスイッチSW1, SW2の設定方法(P.40) ・コマンドのI/Oパラメータ(P.89, P.93)

●このマニュアルに使われている用紙は古紙配合率100%の再生紙を使用しております。
●この印刷物は環境にやさしい植物性大豆油インキを使用しています。

古紙配合率100%再生紙を使用しています　大豆油を主成分としたインキで印刷しています

●在庫・納期・価格など販売に関するお問い合わせは

●技術に関するお問い合わせは

制御機器コールセンター
0120-101-550
※お問い合せ商品 / リレー・機器用センサ・スイッチ・コネクタ・
プログラマブルコントローラ・プログラマブル表示器・
画像処理装置・タイマ・カウンタ・温度調節器
※サービス時間／9:00ー17:00（11:30ー13:00、当社休業日除く）
●FAX …… 06-6904-1573（24時間受付）

松下電工株式会社　制御機器本部
制御デバイス事業部
〒571-8686 大阪府門真市門真1048
TEL.(06)6908-1131〈大代表〉

このマニュアルの記載内容は平成7年2月現在のものです。

■FAF-167①　Printed in Japan