

GPNET

model10+/model20+

取扱い説明書



network supply

◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆ 目 次 ◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆

第1章 概要説明

□応用例	1-1
■直感的理解を得るために簡単なモードA応用例	1-1
■直感的理解を得るために簡単なモードB応用例	1-2
□各部の名称と意味 GPNETmodel110+	1-3
■LEDモニタの意味	1-3
■ディップスイッチの位置	1-3
□各部の名称と意味 GPNETmodel120+	1-4
■LEDモニタの意味	1-4
■ディップスイッチの位置	1-4
□準備・初期設定	1-5
■インタフェースの適合	1-5
① RS232Cインタフェースの適合	1-5
② GPIBインタフェースの適合	1-6
■モードA/Bの設定	1-6
■オンリーモードの設定	1-7
□モードについて	1-8
■モードA/モードB/オンリーモード	1-8
■モードAについて	1-8
■モードBについて	1-9

第2章 プログラミング

□プログラミングの実際	2-1
■モードA：プログラミングの実際	2-1
■モードA：バイナリデータの受信	2-4
■モードA：バイナリデータの送信	2-5
■モードB：プログラミングの実際	2-6

第3章 コマンド説明

□モードAのコマンド	3-1
■初期化コマンド	3-1
●リセットコマンド	3-1
●インタフェースクリア (IFC)	3-1
■設定コマンド	3-2
●デリミタ設定コマンド (DEL)	3-2
●X制御コマンド (XC)	3-3
●通信タイムアウト設定コマンド (TIME)	3-4
■データ通信コマンド	3-5
●データ通信出力コマンド (TLK)	3-5
●データ通信複数行出力コマンド (TML/TMS)	3-6
●データ通信入力コマンド (LSN)	3-8
●データ通信入力コマンド (LSN/2次デリミタ)	3-9
●データ通信複数行入力コマンド (LML)	3-10
●バイナリデータ受信コマンド (LB)	3-11
●バイナリデータ送信コマンド (TB)	3-12
●マルチラインメッセージ・送信 (WB①)	3-13
●マルチラインメッセージ・送信 (WB②)	3-14
●マルチラインメッセージ・受信 (RB)	3-15
■ポーリングコマンド	3-16
●シリアルポールコマンド (POL)	3-16
●オートポールコマンド (AP)	3-17
●オートポールコマンド (AP IF QS)	3-18
●サービスリクエスト受信確認 (?QS)	3-19
●サービスリクエスト発信 (QS)	3-20
●サービスリクエスト待機 (WQS)	3-21
●サービスリクエスト送信確認 (?PD)	3-22
■汎用コマンド	3-23
●デバイスクリアコマンド (DC)	3-23
●デバイストリガコマンド (DT)	3-24
●RENライン制御コマンド (REM/LOC)	3-25
●フォーマットコマンド (FMT)	3-26
●ステータス確認コマンド (?ST)	3-28
●RENライン確認コマンド (?RM)	3-29
●ステータス確認コマンド (?DC)	3-29
●ステータス確認コマンド (?DT)	3-30
●リセットコード変更コマンド (RC)	3-30
●通信同期コマンド (SYC)	3-31
□モードBのコマンド	3-32
●リセット (デバイスクリア)	3-32
●初期設定 "S n Dn1 n2 n3X n /m*"	3-32

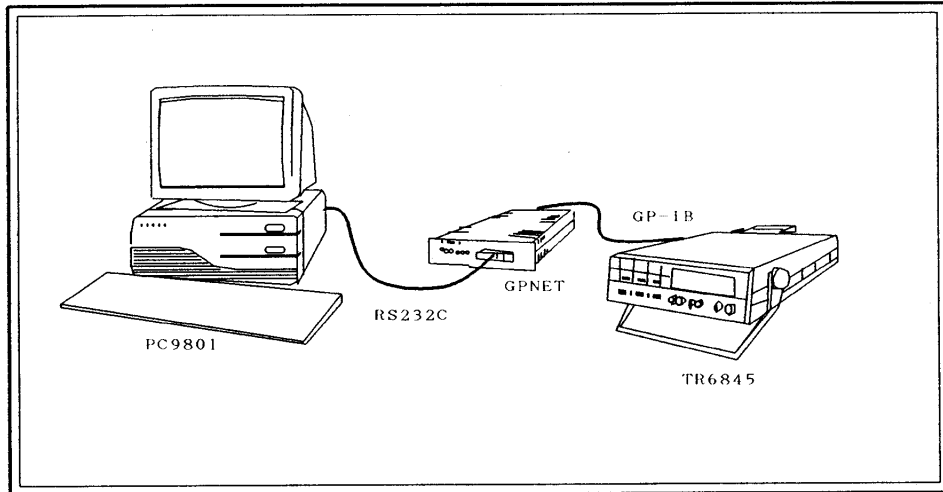
第4章 付 録

- モード A コマンド一覧表
- GPNET model110+ 製品仕様
- GPNET model20+ 製品仕様
- GPNET model110+/model20+ R S 2 3 2 C コネクタ
- GPNET model110+/model20+ G P I B コネクタ
- USASCII Character Set

第一章 概要説明

応用例

■直感的理解を得るために簡単なモードAの応用例を示します。



GP-IB 計測器としてポピュラーなアドバンテストTR-6841（デジタル・マルチメータ）とRS232C側に NEC-PC9801 とを接続しています。

model110+のディップスイッチは

DIP-SW1

DIP-SW2

プログラム例

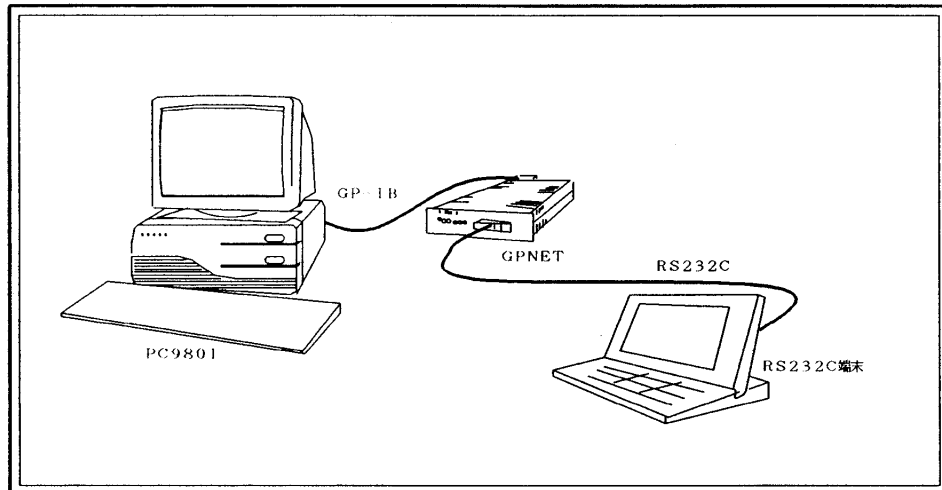
```

20 OPEN "COM:N82XN" AS #1          : RS232Cオープン
30 PRINT#1, CHR$(1); :INPUT#1, G$   : GPNET 初期化
40 PRINT#1, "IFC"                  : インタフェースクリア
50 PRINT#1, "XC 19 17"             : X制御指定
60 PRINT#1, "DC 4"                 : TR6845初期化
70 PRINT#1, "TLK TO 4#S1DLOR0"     : TR6845設定コマンド
                                     出力モード S1
                                     デリミタ CR+LF
                                     レンジ AUTO

80 PRINT#1, "DT 4"                 : デバイストリガ
90 PRINT#1, "LSN FROM 4" :INPUT#1, DAT$ : TR6845よりデータを取り込む
100 PRINT DAT$ :GOTO 80            : データを表示して戻る

```

■直感的理解を得るために簡単なモードBの応用例を示します。



GP-IB コントローラとして NEC-PC9801 を使用しています。RS232C側には、適当なターミナルやコンピュータ等を接続します。

model110+のディップスイッチは

DIP-SW1

DIP-SW2

プログラム例

```

20 ISET IFC           : インタフェークリア
30 CMD TIMEOUT = 50  : タイムアウト設定
40 CMD DELIM = 0     : デリミタCR+LF
50 `
60 WBYTE &H3F, &H21, &H04; : デバイスクリア
70 PRINT@4; "S1D000X0/0*" : Bモード初期設定
80 `
90 PRINT@4; "This is message from PC9801" : 端末へのデータ送信
100 INPUT@4; D$      : 端末からのデータ受信
110 PRINT D$ :GOTO 90 : データ表示し戻る

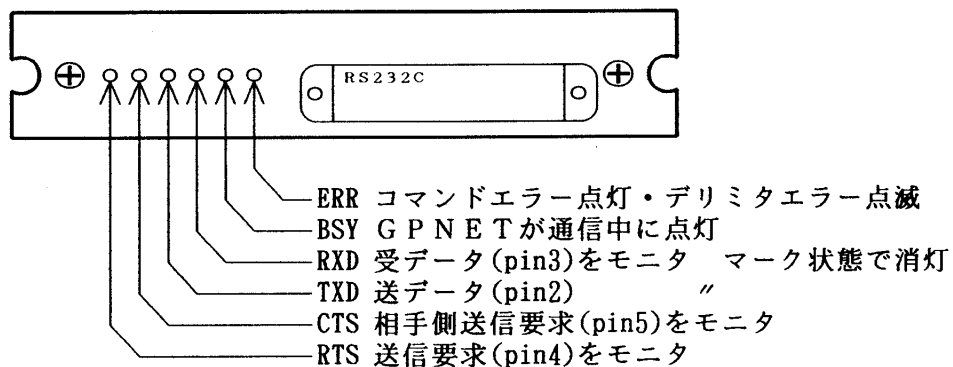
```

※ RS232C端末を、ターミナルモード(9600baud, 2stop, pn, 8bitdata)で待機させてPC9801側のプログラムを走らせませす。

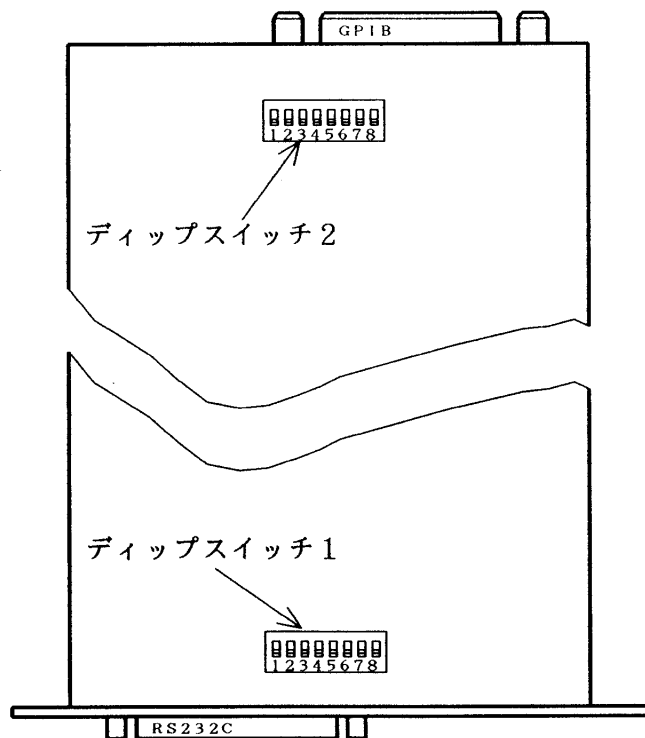
端末のディスプレイに "This is message from PC9801" が表示されます。次に端末のキーボードより適当な文字列を入力するとPC9801に取り込まれ表示されます。

モードBは、RS232Cのデータを単純にGPIBに変換したい時などに使用します。

■LEDモニタの意味

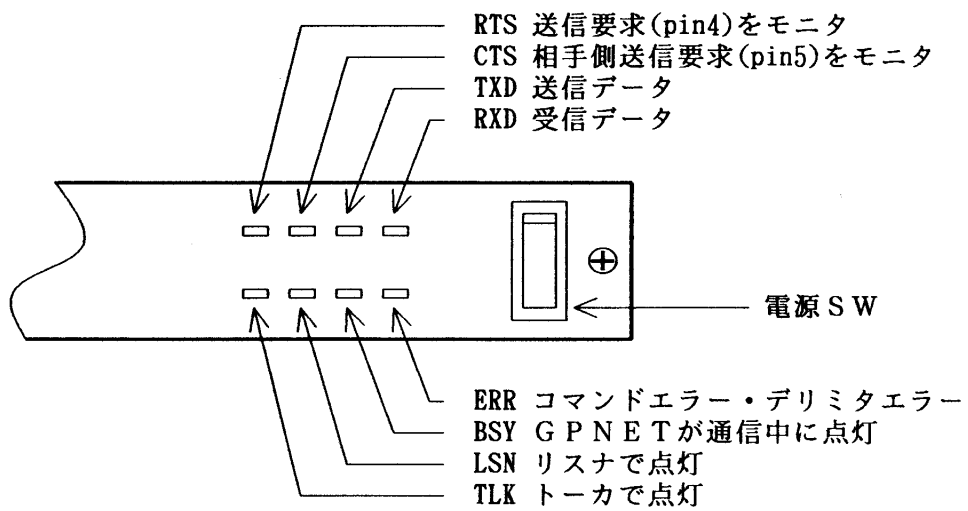


■ディップスイッチの位置(底面)

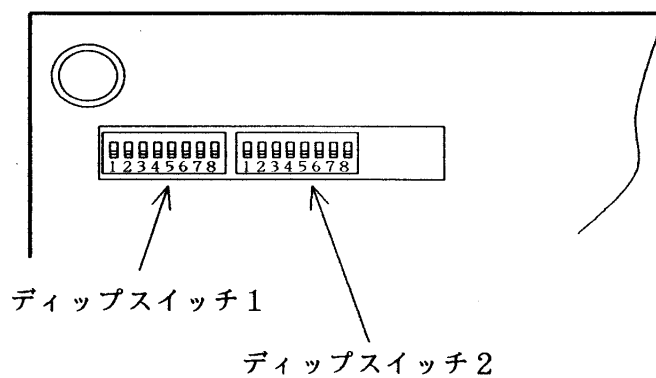


※ディップスイッチの設定方法は『準備・初期設定』を参照して下さい。

■LEDモニタの意味



■ディップスイッチの位置 (底面)



※ディップスイッチの設定方法は『準備・初期設定』を参照して下さい。

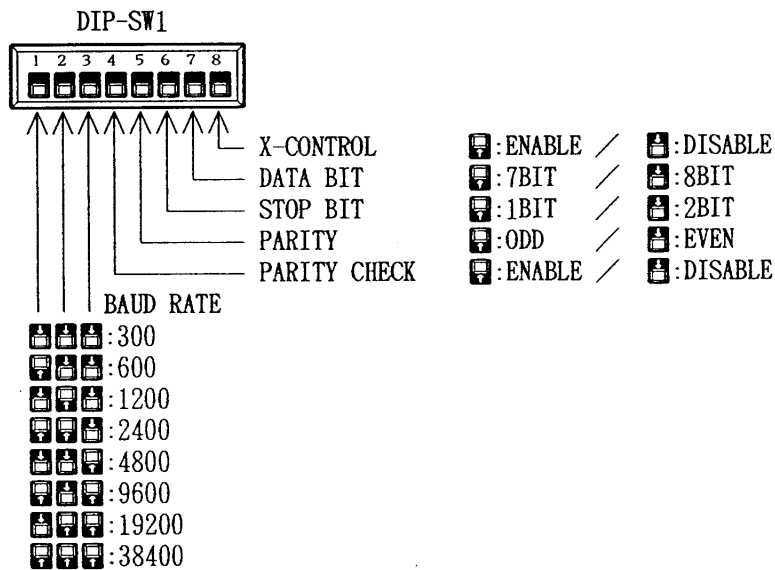
■ インタフェースの適合

GPNETを動作させるためには、GPNETとそれに接続するコンピュータもしくは機器とにおいて、ハード・ソフト面から次のような条件を満足しなければなりません。

① RS232Cインタフェースの適合

a. ボーレート及びデータフォーマットの一致

GPNET裏面のディップスイッチ1により設定します。



b. ハンドシェイク方式の一致

GPNETはCTS(clear to send)とRTS(request to send)によるハンドシェイクまたはX制御をサポートしています。ハンドシェイク方式は、各機器の仕様に従って一致させて下さい。GPNETはDCEタイプです。

c. 非同期調歩同期式 (パソコンは通常この方式が多い)

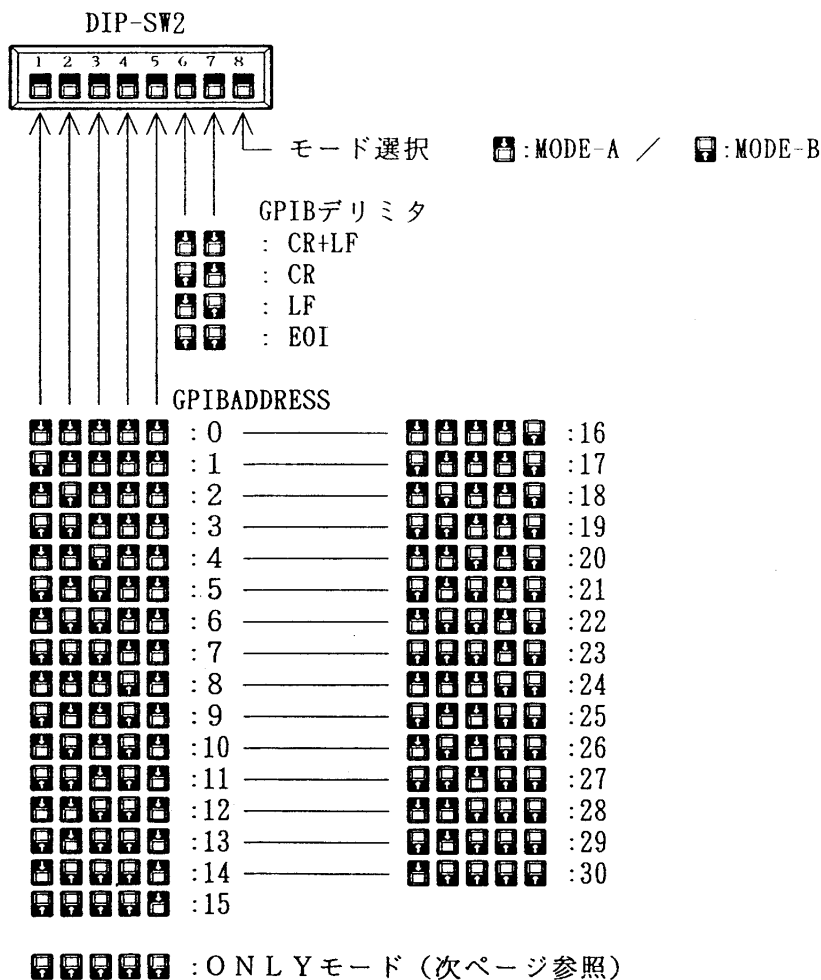
d. 全2重通信方式

② G P - I B インタフェースの適合

- a. I E E E - 4 8 8 のアドレスを正しく設定すること。
I E E E - 4 8 8 バスに接続されている機器はそれぞれ異なるアドレスを持ちます。同一バス上に同一アドレスが複数存在することは許されません。
- b. G P N E T のモード選択 (モード A / B ・オンリーモード)
R S 2 3 2 C 付コンピュータから G P I B 器機を制御する場合にはモード A
G P I B コントローラから R S 2 3 2 C 器機などと通信する場合はモード B
この他にオンリーモード (TLKONLY/LSNONLY)としても機能します。
- c. G P I B デリミタの適合
G P I B 器機によるトラブルの多くはデリミタの不一致に起因します。
必ず、接続される G P I B 器機のデリミタを確認して下さい。
G P N E T のデリミタは、CR+LF, CR, LF, EOI 及び 2 次デリミタ (カンマ) が
指定できます。

■モード A / B の設定

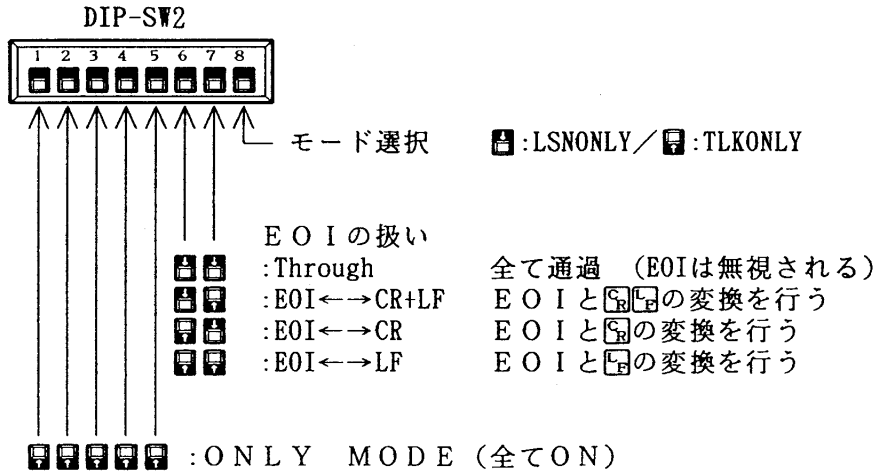
G P N E T 裏面のディップスイッチ 2 により初期状態を設定します。



■オンリーモードの設定

GPNET をオンリーモードで使用する場合のディップスイッチ2の設定方法を示します。ディップスイッチ1の設定方法はモードA/Bの場合と同じです。

GPNET裏面のディップスイッチ2の設定（ONLYモード）



モードについて

■ モードA / モードB / オンリーモード

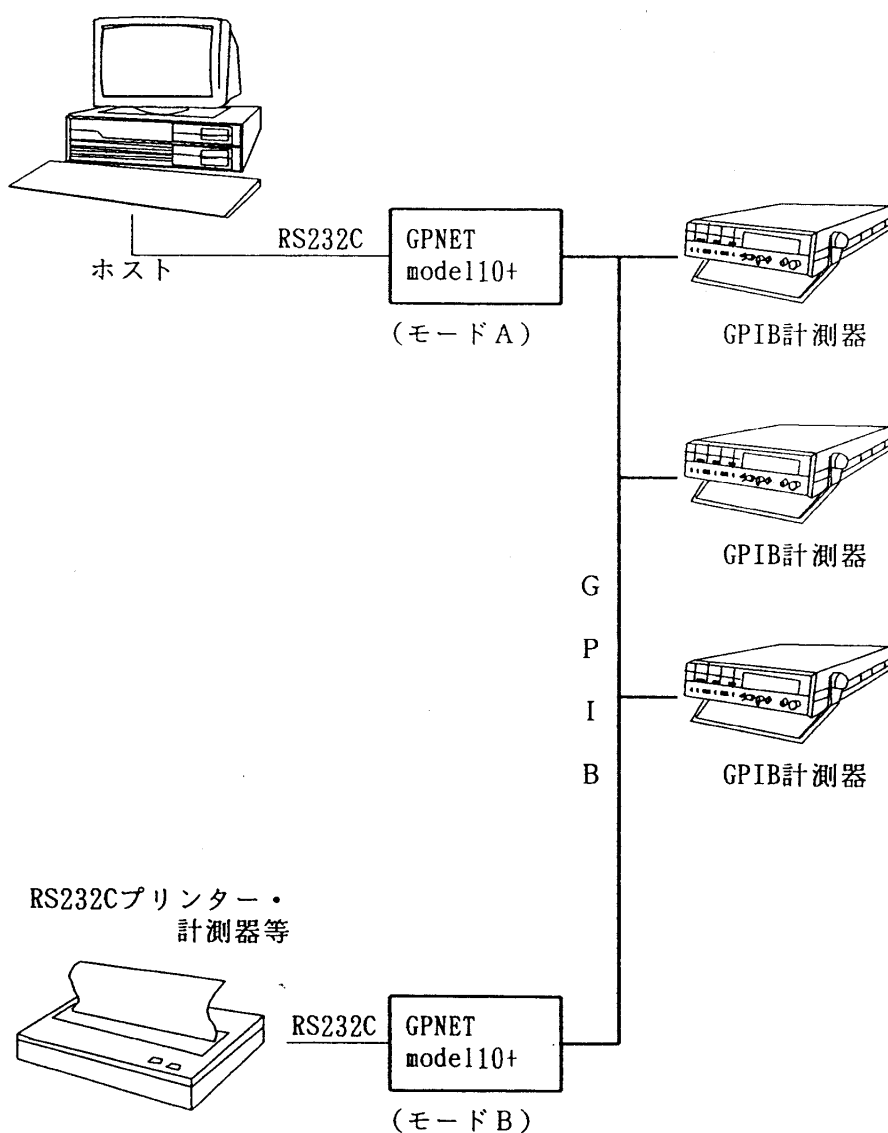
RS232C側からGPIB機能を使用する場合をモードAと呼び、GPIB側からRS232C機器等と通信する場合をモードBと呼びます。

この他、GPIB→RS232CまたはRS232C→GPIBのどちらか1方向だけのオンリーモードの機能も用意されています。

■ モードAについて

本体裏面のDIP-SW2の8番をOFFにすることによりモードAになります。

RS232C側がインテリジェントな機器（コンピュータ等）の場合に、GPIB機器をRS232C側から制御します。マスタコントローラ・トーカー・リスナとしてコンピュータ側から利用します。本機のGPIBアドレスは0～30まで設定可能です。



■モードA：プログラミングの実際

これまでに述べた準備等を基に実際のプログラミングを解説します。ここでは、『モードA応用例』のプログラムを参考にモードAの基本動作と応用について理解を深めます。現代社会を忙しく駆け抜ける諸氏のために、とりあえずこの章を読み終えたのち、巻末のコマンド一覧表を片手にプログラミングできるよう努めました。尚、解説にはBASIC言語を使用していますが言語は何であっても構いません。

① RS232Cオープン

一部の機種を除いてRS232Cポートをオープンするための手続きを必要とします。GPNETの設定(DIPSW1)に適合するように正しく行って下さい。手続き方法は機種により異なりますので、メーカーのマニュアルに従って下さい。

```
OPEN"COM:N82XN" AS #n
```

② GPNET との間で一番最初に行う通信

GPNETは、01H(BASIC表現でCHR\$(1))を受信すると無条件に初期化されます。続いてROMバージョンを送信してきます。このとき通信バッファ等をクリアすることから『ゴミ取り』と称します。ROMバージョンが受信できればRS232Cラインは一応インタフェースされています。必ず最初に行うして下さい。

```
PRINT#n, CHR$(1);  
INPUT#n, G$
```

③ 通信デリミタの指定。

d1は GPIB デリミタ、d2は GPIB 2 次デリミタ (カンマの扱い)、d3は RS232C デリミタです。計測器によってデリミタが異なる場合には、都度このコマンドを実行して下さい。

```
PRINT#n, "DEL d1 d2 d3"
```

値	d1	d2	d3
0	CR+LF	通過	CR+LF
1	CR	デリミタ扱い	CR
2	LF		LF
3	EOI		

- ③ Xコントロールの必要があればここで指定します
n1は送信停止コード、n2は送信開始コードを指定します。 Xコントロールについての詳しい説明は、第2章を参照して下さい。

```
PRINT#n, "XC n1 n2"
```



ここまでの通信の初期化手続きです。

ここからは、GPIBインタフェースとして機能します。



- ④ システムコントローラの宣言をします。
IFC を発行することにより、GPNET はシステムコントローラとなりGPIBバスの主導権が与えられます。 システムコントローラは、GPIBバス上に1台のみ存在が許されます。 GPNET がスレーブの場合には発行しません。

```
PRINT#n, "IFC"
```

- ⑤ 計測器へデータを出力する場合。
例としてアドレスmの機器に"ABCDEF"を送信します。
下例の2行は PRINT#n, "TLK TO m #ABCDEF"と記述することも可能です。

```
PRINT#n, "TLK TO m"    (コマンド)
PRINT#n, "ABCDEF"     (データ)
```

- ⑥ 計測器からデータを取り込む場合。
アドレスmの機器からデータを取り込みます。

```
PRINT#n, "LSN FROM m" (コマンド)
INPUT#n, DAT$         (データ)
```

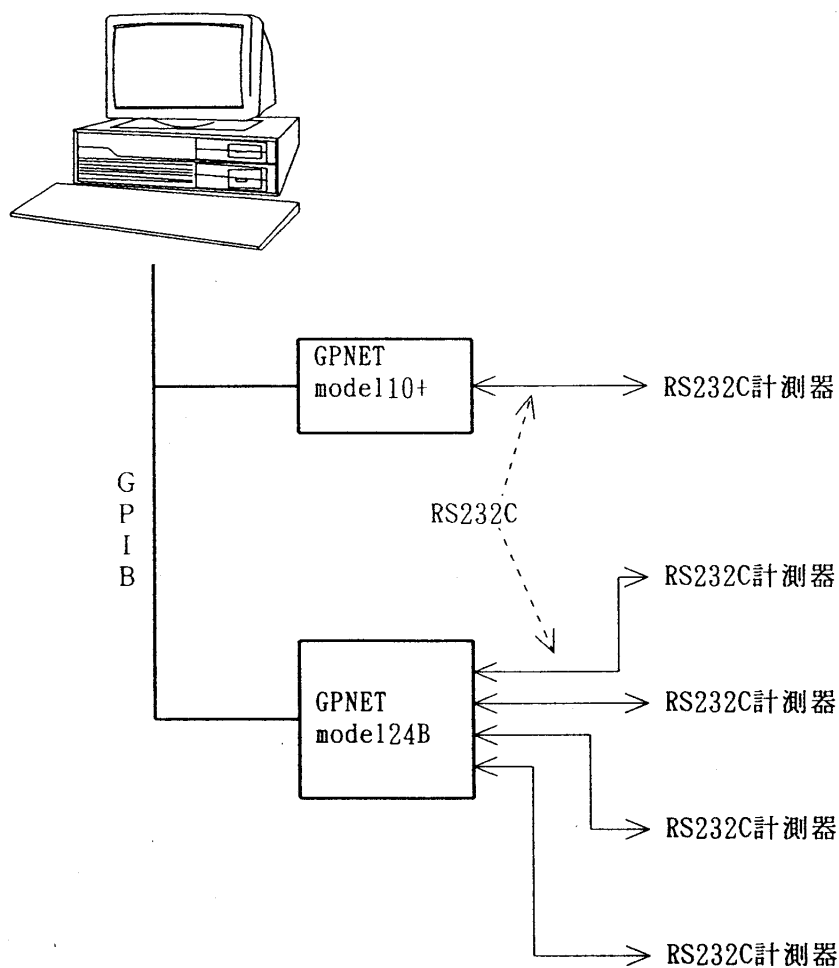
- ⑦ シリアルポールの後、計測器（SRQ発信元）からデータを取り込む。
?QSで GPIB機器からサービスリクエストが発信されているか否かを問い合わせます。下記のプログラムにおいて、70行はシリアルポールのコマンドです d1~dmの順でポーリングを実行し、最初に検出したSRQ発信元のアドレスとステータスバイトを返します。100行はサービスリクエストを発信した機器からデータを取り込みます。
80行でのDVとSBは GPNETから16進アスキー形式で返されたデータを数値変数へ直接取り込んでいますが、コンピュータによっては文字データとして取り込んだ後に関数などで数値に変換しなければならない場合がありますから注意して下さい。

```
50 PRINT#n, "?QS" : INPUT#n, ST$
60 IF ST$ <> "1" THEN 50
70 PRINT#n, "POL d1 d2 .... dm"
80 INPUT#n, DV, SB
90 IF DV=0 THEN 50
100 PRINT#n, "LSN FROM "; DV
110 INPUT#n, D$
120 PRINT "device="; DV, "Data="; D$
```


■ モードBについて

本体裏面のDIP-SW2の8番をONにすることによりモードBになります。
RS232C付プリンターや計測器などをGPIB側からコントロールします。
この機能は、RS232C側がインテリジェントである必要がなく、GPIBの出力がそのままRS232Cに置換えられます。アドレスは1~30まで設定可能です。

モードBは極めて応用範囲が広く、手軽にデータ収集システムが構築できます。
RS232C付のデジタイザー、バーコードリーダー、計測器、出力装置、その他様々の機器がGPIB端末に変身します。



※ GPNET model124Bは、4チャンネルのRS232C通信マルチプレクサです。

第二章 プログラミング

■モードA：バイナリデータの受信

IEEE-488バスから受信しバイナリ形式のまま RS-232Cへ送信する場合。

LSNコマンドによる方法

```

デリミタ設定： PRINT#1, "DEL 3 0 3"
タイムアウト： PRINT#1, "TIME n"
                : PRINT#1, "LSN FROM 5" :CALL INPSUB
                                     ↳機械語ルーチン

```

最終バイトをEOIとともに送信された、バイナリデータをRS232C側に送出します。このとき最終バイト送出後、TIMEコマンドで指定されただけダミーカウントします。コンピュータ側はタイムアウトにより、送信終了を検出します。

LB@コマンドによる方法

```

タイムアウト： PRINT#1, "TIME n"
                : PRINT#1, "LB@ FROM 5" :CALL INPSUB
                                     ↳機械語ルーチン

```

最終バイトをEOIとともに送信された、バイナリデータをRS232C側に送出します。このとき最終バイト送出後、TIMEコマンドで指定されただけダミーカウントします。コンピュータ側はタイムアウトにより、送信終了を検出します。

IEEE-488バスから受信しアスキー変換した後 RS-232Cへ送信する場合。

LB, LB*コマンドによる方法

```

100 PRINT#1, "DEL 3 0 0"           100 PRINT#1, "DEL 3 0 0"
110 PRINT#1, "LB FROM 5"           110 PRINT#1, "LB* FROM 5"
120 PRINT#1, "" :INPUT#1, D$       120 INPUT#1, D$ :PRINT D$
130 PRINT D$ :IF D$<>"*" THEN 120

```

最終バイトとともに送信された、バイナリデータをアスキー変換した後にRS232C側に出力します。25H, FFH, D3H が送られてきた場合次のようになります。

(LBコマンドの場合)

25デリミタ
FFデリミタ
D3デリミタ
*デリミタ

(LB* コマンドの場合)

25FFD3デリミタ

■モードA：バイナリデータの送信

アスキー表現によるHEXデータをGPNETに与えます。次の2例の場合、GPNETは全く同じ動作をします。

TBコマンドによる方法

```
PRINT#1, "TB TO 4#1A0DDFFC"
```

WBコマンドによる方法

```
PRINT#1, "WB 3F 24/1A0DDFFC"
```

..... プロトコルについて

GPNET model10のプロトコルは、FMTコマンドにより何種類かの選択を可能にしています。GPIB機器のなかには、UNT, UNL, EOIなどを受信するときのタイミングに制約を設けているものが存在することによるものです。

例えば、UNLをデータモードの最後につけるとバスロックしたり、UNTを最後に送信しないと停止してしまうといった不完全な機器もあり、これにEOIがからんでいっそう面倒なことになってしまうケースもよくあります。

FMTコマンドはプロトコルについて以下のような選択ができます。

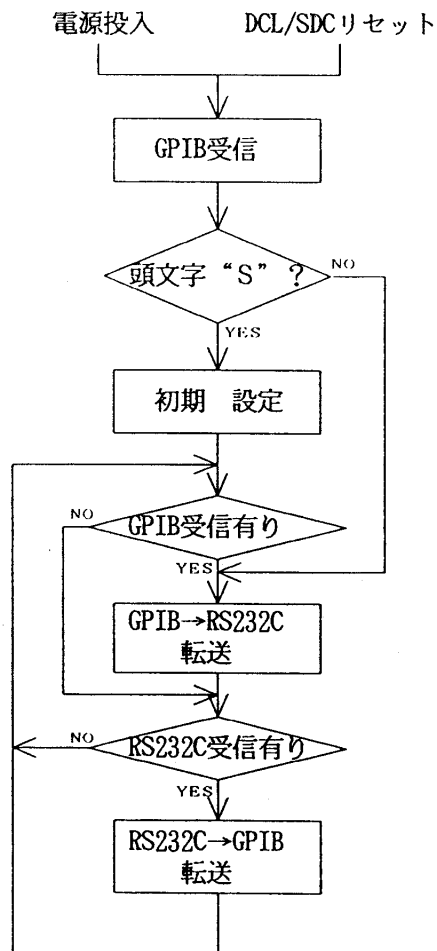
- ① UNTの有無
- ② UNLの有無
- ③ デリミタと共に、EOIを送信するか否か

■モードB：プログラミングの実際

モードBはRS232C出力の計測器やターミナルなどを、GPIBバスに接続する事を目的に開発されたソフトウェアです。モードBは、ディップスイッチにより選択します。転送単位はデリミタで区切られた一行です。

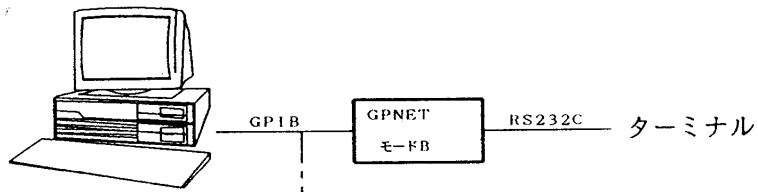
- ① ハード面の初期設定を終えたのち電源を投入します。
- ② GPNET は、まずGPIBからの入力を待ちます。
- ③ 電源投入直後およびデバイスクリアによるリセット直後、GPNET は“S”で始まる一行のコマンドを受けられます。最初の文字が“S”以外ならデータとして解釈してデフォルト値（ディップスイッチによる）のまま通信に移ります。
この“S”コマンドはモードB唯一のコマンドでありリセット後に一度だけ有効です。再設定の場合はデバイスクリアか電源の入切を行って下さい。詳しくは第2章を参照して下さい。

【モードBの基本動作】



④ 動作条件に従って通信します。

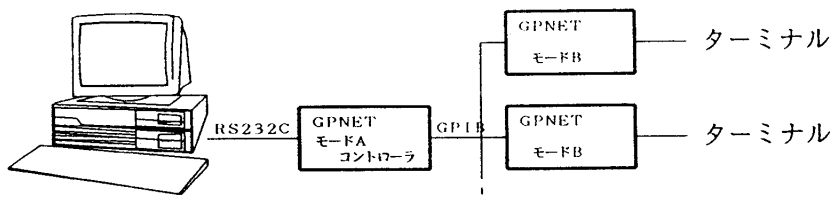
下記にモードBによるプログラム例を示します。
 デバイス4にRS232Cターミナルを接続し（デリミタ= \square 、カンマ=通過、
 X制御=無、SRQ=禁止）の内容で初期設定します。
 コントローラ(PC9801)が“A B C D E F”の文字列を送ったあとターミナル
 からの受信を待ちます。



```

10 ISET IFC
20 CMD DELIM =0
30 WBYTE &H3F, &H21, &H04;
40 PRINT@4; "S1D000X0/0*"
50 PRINT@4; "ABCDEF"
60 INPUT@4; D$
  
```

コントローラにGPNET(モードA)を使用してRS232Cより同じ通信を行
 った例を示します。



```

10 PRINT#1, "IFC"
20 PRINT#1, "DEL 0 0 0"
30 PRINT#1, "DC 4"
40 PRINT#1, "TLK TO 4 #S1D000X0/0*"
50 PRINT#1, "TLK TO 4 3ABCDEF"
60 PRINT#1, "LSN FROM 4" : INPUT#1, D$
  
```

第三章 コマンド説明

■モードAのコマンド

● リセットコマンド

```
PRINT#1, CHR$(1) ; : INPUT#1, G$
```

【機能】 GPNET内部のレジスタおよびメモリ等を初期化します。

【解説】 CHR\$(1) つまり01を受信することにより GPNETは無条件に初期化され、初期メッセージとしてROMバージョンを送信します。

GPNET がエラー状態やバスロック状態から抜ける時にも有効です。
また、システムの都合で01が使用できない場合にはRCコマンドによってリセットコードの変更または禁止ができ、電源切断まで有効です。リセットにより GPNETは電源投入直後と同じ状態に戻ります。

● インタフェースクリア

```
PRINT#1, "IFC"
```

【機能】 GPIBインタフェースを初期化し、システムコントローラを宣言します。

【解説】 IEEE-488バスのIFCラインを0.1秒間trueにした後RENラインをtrueにします。
このコマンドを実行することで、システムコントローラとなります。IEEE-488の規定ではシステムのなかに存在しうるシステムコントローラは1台に限られます。従って、このコマンドはシステムコントローラとなるべきデバイスのみが発行できます。




● デリミタ設定コマンド

```
PRINT#1, "DEL n1 n2 n3"
```

【機能】 RS232Cの側からGPNETのデリミタを設定します。

【文例】 PRINT#1, "DEL 2 0 1"

【解説】 ① n1の値により、GPIBデリミタの設定をします。
デフォルト時は (n1=0) となっています。




```
n1=0 . . . . . 
n1=1 . . . . . 
n1=2 . . . . . 
n1=3 . . . . . EOI
```

② n2の値により、カンマの扱いを指定します。
(GPIB2次デリミタ)

```
n2=1 . . . . . カンマをデリミタとして扱う
n2=0 . . . . . カンマをデータとして扱う
```

※n2=1とした場合にはLSNコマンドの手順が変わります。
詳しくはLSNコマンドの解説を参照して下さい。

③ n3の値により、RS232Cデリミタの設定を行います。
デフォルト時は (n1=0) となっています。

```
n1=0 . . . . . 
n1=1 . . . . . 
n1=2 . . . . . 
```

● X制御コマンド

```
PRINT#1, "XC c1 c2"
```

【機能】 Xコントロール・コードを指定します。

【解説】 Xコントロールとは、コントロールラインによる通信制御の代わりに伝送ライン（データライン）を利用して送信停止と許可のコードを送り、通信の制御を行う方式です。データライン2本とGNDライン1本の計3本の伝送ラインで通信制御ができるために合理的な配線が可能です。

- ① XOFF, XONの順に10進表記で記述して下さい。
一般的には、XOFF=19(13H) XON=17(11H)です。

```
PRINT#2, "XC 19 17"
```

← XON (送信開始コード)
← XOFF (送信停止コード)

- ② RTS-CTS制御を行わない場合には、GPNETのCTSとRTS(4番と5番)をショートする必要があります。
- ③ デフォルトは“XC 0 0”となっています。この場合Xコントロールは無効となります。

● 通信タイムアウト設定

```
PRINT #1, "TIME n"
```

【機能】 通信異常時のタイムアウト設定をします。

【文例】 PRINT #1, " TIME 250" (約25秒)

- 【解説】
- ① 次の状態の時タイムアウトカウンタが計時を開始します。
 1. トーカがコマンドまたはデータをバスに出力した時。
 2. リスナがデータを読み出しに行った時。
 - ② 約0.1秒毎に0～25秒位まで設定できます。nは0～255の範囲です。
 - ③ 通信が正常に行われたかどうかの確認は、“0”なら正常“1”なら異常です。このコマンドは通信の可否を確認する必要がある場合に使用すればよく、毎回使用しなくても構いません。
 - ④ 通信開始または途中でタイムアウトが発生した場合、強制的に通信状態を打ち切ります。

```
100 PRINT #1, "TIME 20"           2秒
110 PRINT #1, "TLK TO 4#S1DLOR0"
120 PRINT #1, "?TIME" : INPUT #1, TI$
130 IF TI$="1" THEN 2000
140 .....

2000 PRINT " タイムアウト エラー!!" :END
```

- ⑤ ライン120は必要に応じて実行してください。

● データ通信出力

```
PRINT #1, "TLK TO d1 d2 ..dn#" : DT$
```

【機能】 指定デバイス（1～複数）に、文字列を出力します。

【文例】 PRINT #1, " TLK TO 5#ABCDEFGF"

- 【解説】
- ① TLKコマンドは、デリミタコードとリセットコード以外はすべて通過します。
 - ② “#”はコマンドの終了をGPNETに知らせ、次にデータが続くことを意味します。
 - ③ 1ないし複数のデバイスに対してデータを出力します。指定されたあるデバイスが応答しない場合の対応は、タイムアウト指定値により異なります。タイムアウト値は“TIME”コマンドにより、指定します。（後述）

タイムアウト	0	ロックアップしたままの状態
“	1～255	指定された時間経過後に復帰

タイムアウト発生の場合他のデバイスへの出力も無効です。通信途中でのタイムアウト発生の有無は“?TIME”コマンドで確認できます。

```
100 PRINT #1, "TLME 20"
110 PRINT #1, "TLK TO 4 5 9#S1DLORO"
120 PRINT #1, "?TIME" : INPUT#1, TI$
130 IF TI$="1" THEN 2000
```

```
2000 PRINT " タイムアウト エラー!!" :END
```

- ④ 指定できるデバイス数に制限はありません。
- ⑤ UNT, UNLの出力はFMTコマンドの指定に従います。
- ⑥ 次の3つの表現は、同じ動作をします。

1. PRINT #1, "TLK TO d1 d2 ..dn"
PRINT #1, "ABCDEF"
2. PRINT #1, "TLK TO d1 d2 ..dn#ABCDEF"
3. PRINT #1, "TLK TO d1 d2 ..dn#";DAT\$
(DAT\$="ABCDEF")

● データ通信複数行出力

```

PRINT #1, "TML TO d1 d2 . . . dn"
PRINT #1, A$
PRINT #1, B$
PRINT #1, C$
-----
-----
PRINT #1, CHR$(8)      ←通信打ち切りコード

```

【機能】 指定デバイス（1～複数）に、複数の文字列を出力します。

【文例】

```

PRINT #1, " TML TO 5 6"
PRINT #1, " ABCDEFGHI"
PRINT #1, " 1234567890"
PRINT #1, CHR$(8)      ←EOI をtrueにする

```

- 【解説】
- ① TMLコマンドでは、デリミタとリセットコードおよびCHR\$(8)以外のコードは全て通過します。タイムアウトの設定はできません。正常な通信が確認されている相手とのみ交信してください。
 - ② GPIBデリミタとしてEOIの指定 (n1=3) はできません。TMLコマンドでは最終行を送信するときのみEOIを使い他のデータ行と区別しています。受信側はEOI以外のデリミタの場合、続いて入力行があるとみなして受信動作に入ります。EOIがtrueのとき最終行とみなして終了します。
 - ③ TLKコマンドは1行送信する毎にトーカ・リスナ解除をします。TLKコマンドはEOIがtrueになるまで解除しない為に効率のよい転送ができます。

同期型複数行送信

```
PRINT #1, "TMS TO 5"  
INPUT #1, S:PRINT #2, A$  
INPUT #1, S:PRINT #2, B$  
INPUT #1, S:PRINT #2, C$  
INPUT #1, S:PRINT #2, CHR$(8)
```

【機能】 GPNET ホストとの同期をとりながら複数行送信します。

【文例】 PRINT#1, "TMS TO 5"

【解説】 ① 1行のデータを送信する前にINPUT文を実行し、ダミー・データを受け取ってから送信します。Xコントロールも、コントロール・ラインも持たない場合に、このコマンドを使用することにより、バッファ・オーバーセズに通信可能です。

1. 受信側は、“LML”で受信します。

● データ通信入力

```
PRINT #1, "LSN FROM d"
INPUT #1, D$
```

【機能】 指定デバイスから、デリミタで区切られたデータを入力します。

【文例】 100 PRINT #1, "LSN FROM 5"
110 INPUT #1, D\$

【解説】 ① LSNコマンドで扱うことのできるコードは、RS232Cデリミタを除くすべてです。DELコマンドによりGP-IBデリミタをEOI (n1=3) に設定している場合バイナリーデータを受信できますが、DELコマンドによるRS232Cデリミタ設定の違いにより次のような2つの形式があります。

- a. n3=3の場合、GP-IB バスのEOIラインがtrueになったときにRS232Cの送信を打ち切ります。このときTIMEコマンドが前に実行されていなければ、設定値までカウントします。ホスト側はこのタイムアウトを検出することにより、デリミタとして扱います。
- b. n3が3 以外の場合、GP-IB バスのEOIラインがtrueになったとき指定されたデリミタコードを送信した後打ち切ります。

② 通信途中でのタイムアウト発生の有無は" ?TIME" コマンドで確認できます。

```
100 PRINT #1, "TIME 20" ; 2秒を設定
110 PRINT #1, "LSN FROM 5" : INPUT #1, DAT$
120 PRINT #1, "?TIME" : INPUT #1, TI$
130 IF TI$="1" THEN 2000
```

```
2000 PRINT " タイムアウト エラー!!" :END
```

③ UNLの出力はFMTコマンドの指定に従います。
(デフォルト0)

④ データ送信後にUNTを出力します。

- データ通信入力 (GP-IB 2次デリミタ使用)

```
10 PRINT#1, " LSN FORM d"  
20 PRINT#1, " " : INPUT#1, DAT$  
   ----- データの処理 -----  
80 PRINT#1, " " : INPUT#1, E$  
90 IF E$="0" THEN 20  ↳ EOIのとき "1"
```

【機能】 指定のデバイスから2次デリミタで区切られた複数行データを入力します。

- 【解説】
- ① LSNコマンド (GP-IB 2次デリミタ使用) で扱うことのできるコードはRS232Cデリミタと GP-IBデリミタ、GP-IB 2次デリミタを除くすべてです。タイムアウトの設定はできませんので正常な通信が確認されている相手とのみ交信してください。
 - ② GP-IB 1次デリミタはCR, LF, CR+LF, EOI が指定できます。
 - ③ 2次デリミタの場合、続いて入力行があるとみなして受信動作に入ります。1次デリミタがtrueのとき最終行とみなして終了します。

● データ通信複数行入力コマンド

```

10 PRINT#1, "LML FROM d"
20 PRINT#1, " " : INPUT#1, DAT$
   ----- データの処理 -----
80 PRINT#1, " " : INPUT#1, E$
90 IF E$="0" THEN 20  ↳ EOIのとき "1"

```

【機能】 指定デバイスから複数行のデータを入力します。

- 【解説】
- ① LMLコマンドで扱うことのできるコードは、RS232CデリミタとGP-IB デリミタを除くすべてです。タイムアウトの設定はできませんので、正常な通信が確認されている相手とのみ交信してください。
 - ② GP-IB デリミタとしてEOIの指定はできません。LMLコマンドでは最終行を送信するときのみEOIを使い、他のデータ行と区別しています。
受信側はEOI以外のデリミタの場合、続いて入力行があるとみなして受信送信側と受信側の対応を示します。

送信側 (コントローラ)	受信側 (スレーブ)
100 PRINT #1, "TML TO 3"	100 PRINT #1, "LML"
110 PRINT #1, "1234567890"	110 PRINT #1, " "
120 PRINT #1, "ABCDEFGHJKLMN"	120 PRINT #1, "DAT\$"
130 PRINT #1, S\$	130 PRINT DAT\$
140 PRINT #1, CHR\$(8)	140 PRINT #1, " " : INPUT #1, E\$
150 END	150 IF E\$="0" THEN 110
	160 END

- ③ LSNコマンドは1行送信する毎にトーカ・リスナ解除をします。LMLコマンドはEOIがtrueになるまで解除しない為に効率のよい転送ができます。

● バイナリーデータ受信

```
PRINT #1, "LB FROM d" : INPUT #1, D$
PRINT #1, "LB* FROM d" : INPUT #1, D$
PRINT #1, "LB@ FROM d" : CALL INPSUB
```

【機能】 指定デバイスより、バイナリーデータを受信します。

【文例】

```
100 PRINT#1, "LB FROM 6"
110 PRINT#1, " " : INPUT#1, D$
120 IF D$<>"*" THEN END
```

```
100 PRINT#1, "LB* FROM 6" : INPUT#1, D$
100 PRINT#1, "LB@ FROM 6" : CALL INPSUB
```

【解説】 送信されてくるデータが、01, FF, 3E, D0, 4 byteバイナリーとします。

- ① “LB FROM n” の場合バイナリーデータはアスキー表現 + デリミタで出力され、最終バイト (EOI=true) の次ぎに “*” (アスタリスク) を出力します。

```
01 [デリミタ]
FF [デリミタ]
3F [デリミタ]      ←CRT に表示した場合
D0 [デリミタ]
* [デリミタ]
```

- ② “LB* FROM n” の場合、アスキー表現が連続して出力され最終バイトの次ぎにデリミタを出力します。

```
01FF3ED0 [デリミタ]      ←CRT に表示した場合
```

- ③ “LB@ FROM n” の場合、バイナリーデータのまま出力されます。この時、DEL コマンドにより n3=3 に設定してある場合、最終バイトで打ち切り、TIME コマンドで設定されたタイムカウンタの後に終了します。その他の場合は指定デリミタが付加されますのでご注意ください。

BASIC の入力コマンドでは、バイナリーデータは受け取れませんので CALL コマンド等でプログラミングする必要があります。

● バイナリーデータ送信コマンド

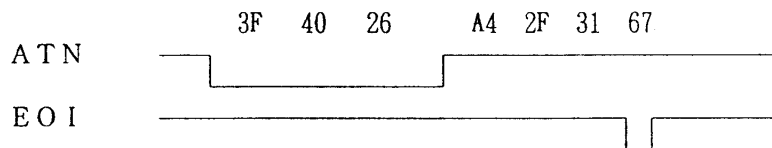
```
PRINT#1, "TB TO d#A42F3167"
```

【機能】 指定デバイスにバイナリーデータを送信します。

【文例】 PRINT #1, "TB TO 6#A42F3167"
PRINT #1, "TB TO 6#A42F3167*"

- 【解説】 ① アスキー表現のHEXデータをバイナリーコードに変換して送信します。
- ② データの最後に"*"（アスタリスク）がある場合直前のデータと同時にEOIラインをtrueにします。

```
PRINT#1, "TB TO 6#A42F3167*"
```



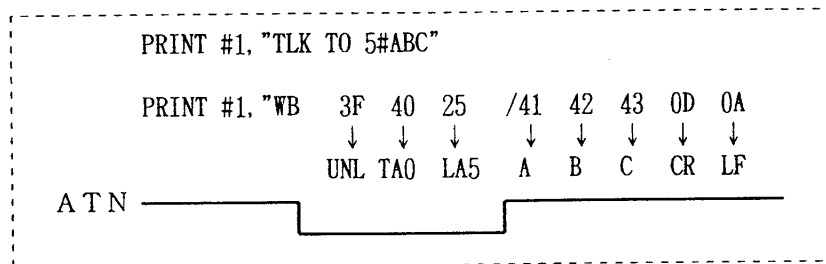
● マルチラインメッセージコマンド

```
PRINT #1, "WB Hex Hex Hex . . ."
PRINT #1, "WB Hex Hex ./Hex Hex . . ."
```

【機能】 HEX表現のコマンド・データをバスラインに出力します。

【文例】 PRINT #1, "WB 3F 40 24 08"

- 【解説】
- ① WBコマンド実行直後ATNラインはtrueになります。続いてHEX表現のデータをバスラインにコマンドとして出力します。
 - ② “/” (スラッシュ) はATNラインを反転します。デリミタは付加されません。ATNラインはfalseの状態を終了します。



- ③ “WB” に続いてスペースはあっても無くても構いません。HEXデータは必ず2文字で表現してください。

```
"AF DD 23 8"      ←×
"AF DD 23 08"     ←○
```

- ④ データの最後に “*” (アスタリスク) がある場合直前のデータの出力と同時にEOIラインをtrueにします。
- ⑤ このコマンドは、コントローラのみ発行できます。
- ⑥ UNL, UNT, デリミタ等の扱いについては、充分注意してお使いください。

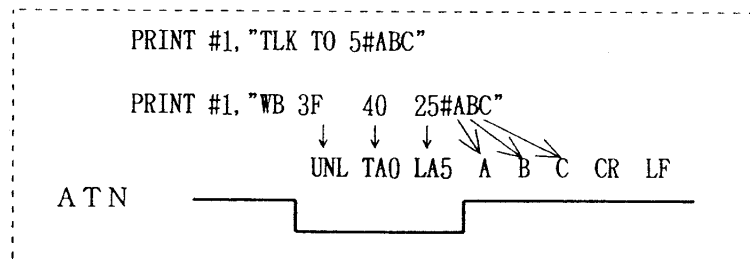
● マルチラインメッセージコマンド

```
PRINT #1, "WB Hex Hex . . . # d d d . . ."
```

【機能】 HEX表現のコマンド及び文字列データをバスラインに出力します。

【文例】 PRINT #1, "WB 3F 40 24#ABC"

- 【解説】
- ① WBコマンド実行直後ATNラインはtrueになります。続いてHEX表現のデータをバスラインにコマンドとして出力します。
 - ② “#” に続く文字列をバスラインにデータとして出力します。デリミタの扱いについては、指定されたフォーマット（DELコマンド又はデフォルト）に従います。UNT, UNLは出力されません。ANTラインはfalseの状態を終了します。



- ③ このコマンドは、コントローラのみ発行できます。
- ④ UNL, UNT, デリミタ等の扱いについては、充分注意してお使いください。

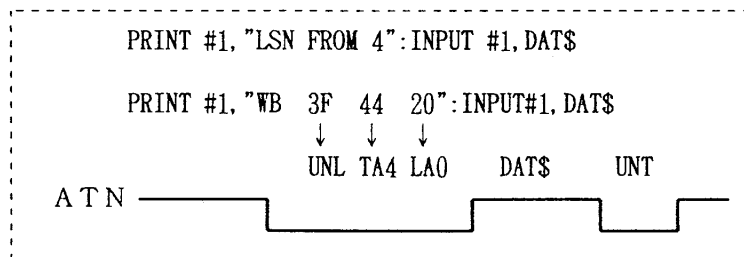
● マルチラインメッセージコマンド

```
PRINT #1, "RB Hex Hex . . ."
INPUT #1, DAT$
```

【機能】 HEX表現のコマンドをバスラインに出力した後にデータ入力します。

【文例】 PRINT #1, "RB 3F 44 20":INPUT #1, DAT\$

【解説】 ① RBコマンド実行直後ATNラインはtrueになります。続くHEX表現のデータをバスラインにコマンドとして出力します。コマンド実行後に必ずINPUTポート文によって、データ入力する必要があります。



- ② データ送信後にUNTを出力します。
- ③ UNL, UNT等の組み合わせを自在にしたい場合などに使用します。入力データのデリミタの取扱いはDELコマンドによる指定、又はデフォルトに従います。
- ④ このコマンドは、コントローラのみ発行できます。
- ⑤ UNL UNT デリミタ等の扱いについては、充分注意してお使いください。

● シリアルポールコマンド

```
PRINT #1, "POL d1 d2 . . . dn"
PRINT #1, DV, ST
```

【機能】 指定デバイスのシリアルポールを実行します。

【文例】

```
200 PRINT #1, "?QS":INPUT #1, QS$
210 IF QS$<>"1" THEN GOTO 200
220 PRINT #1, "POL 1 2 3 4 5 6"
230 INPUT #1, DV, ST
240 IF DV=0 THEN GOTO 200
250 PRINT #1, "LSN FORM";DV:INPUT #1, DAT$
```

- 【解説】
- ① 指定デバイス（1～30）に対してコマンドラインで指定されている順にシリアルポールを実行し、ステータスバイトが0でない時DVにデバイスアドレス、STにステータスバイトを返します。16進アスキー形式のデータです。言語によっては直接、数値変数に取り込めないものもありますので注意してください。
 - ② シリアルポールを実行しても、どのデバイスも応答しなかった時DVには"&H00"を返します。SRQラインが"true"であっても機器側のなんらかの理由によりすぐには応答しない場合がありますから、状況に応じPOLコマンドを繰り返すことも必要です。
 - ③ タイムアウト機能をアクティブにすればロックアップを回避できます。
 - ④ 多数のデバイスを高速にポーリングする必要がある時は、APコマンドの方が有効です。ただしAPコマンドではステータスバイトのビットは“真”でなければなりません。
 - ⑤ 残ったポーリングデバイスへのポーリングは実行されません。

- シリアルポールコマンド (オートポール)

```
PRINT #1, "AP" : INPUT #1, DV, ST
```

【機能】 PDVコマンドで登録されたデバイスのシリアルポールを実行します。

【文例】

```
200 PRINT#1, "?QS" : INPUT#1, QS$
210 IF QS$ <> "1" THEN GOTO 200
220 PRINT#1, "AP" : INPUT#1, DV, ST
230 IF DV=0 THEN GOTO 200
240 PRINT#1, "LSN FORM"; DV : INPUT#1, DAT$
```

- 【解説】
- ① 指定デバイス (1~30) に対してPDVコマンドで登録されている順にシリアルポールを実行しステータスバイトが真の時DVにデバイスアドレス、STにステータスバイトを返します。16進アスキー形式のデータです。言語によっては直接、数値変数に取り込めないものもありますので注意してください。
 - ② シリアルポールを実行しても、どのデバイスも応答しなかった時DVには"&H00"を返します。SRQラインが"true"であっても機器側のなんらかの理由によりすぐには応答しない場合がありますから状況に応じてPOLコマンドを繰り返すことも必要です。
 - ③ 残ったポーリングデバイスへのポーリングは実行されません。

● シリアルポールコマンド

```
PRINT #1, "AP IF QS"
INPUT #1, DV, ST
```

【機能】 SRQラインが“true”の時にAPコマンドを実行します。

【文例】 200 PRINT #1, "AP IF QS" : INPUT #1, DV, ST
210 IF DV=0 THEN GOTO 200
220 PRINT #1, "LSN FORM";DV : INPUT #1, DAT\$

【解説】 ① SRQラインが“true”の時にAPコマンドを実行する複合コマンドです。
② 次の2つのプログラムは同じ意味を持ちます。

100 PRINT#1, "AP IF QS"	100 PRINT#1, "?QS" : INPUT#1, QS\$
110 INPUT#1, DV, ST	110 IF QS\$<>"1" THEN 100
120 IF DV=0 THEN GOTO 100	120 PRINT#1, "AP" : INPUT#1, DV, ST
130 PRINT#1, "LSN FROM";DV	130 IF DV=0 THEN GOTO 100
140 INPUT#1, DAT\$	140 PRINT#1, "LSN FROM";DV
	150 INPUT#1, DAT\$

● ポーリングデバイス登録コマンド

```
PRINT #1, "PDV d1 d2 . . . .dn"
```

【機能】 指定デバイスをAPコマンド用のテーブルに登録します。

【文例】 PRINT #1, "PDV 1 2 3 4 5 6"

【解説】 ① APコマンドはPDVコマンドで登録されたデバイスに対してシリアルポールを実行します。nは1～30まで登録できます。
② POLコマンドによるポーリングデバイス指定と違い GPNET内部のレジスタに登録されるため、高速動作をします。

● サービスリクエスト確認コマンド

```
PRINT#1, "?QS" : INPUT#1, Q$
```

【機能】 コントローラが、スレーブからのサービスリクエストの有無を問い合わせます。

【解説】 ① IEEE-488バスのSRQラインを“true”にしてスレーブがSRQメッセージを発しているか否か、ステータスを返します。

```
SRQ 有り   → “1”  
SRQ 無し   → “0”
```

② コントローラは、SRQ有りの場合に通常シリアルポールを実行します。

③ デバイス5,6,7にトリガをかけ、SRQを発信した順にシリアルポールを実行した場合を仮定して、下記にサンプルプログラムを示します

```
100 PRINT#1, "DT 5 6 7"  
110 PRINT#1, "?QS" : INPUT#1, Q$  
120 IF Q$ <> "1" THEN 110  
130 PRINT#1, "POL 5 6 7" : INPUT#1, DV, ST  
140 IF DV=0 THEN 180  
150 PRINT#1, "LSN FROM"; DV : INPUT#1, DATA$  
160 PRINT DATA$  
170 GOTO 130  
180 PRINT " 測定終了"
```

④ このコマンドは、コントローラのみ発行できます。

● サービスリクエスト発信コマンド

```
PRINT #1, "QS nn"
```

【機能】 スレーブがSRQ（サービス・リクエスト）を発信します。

【文例】

```
100 PRINT #1, "QS 66"  
110 PRINT #1, "?ST" : INPUT #1, ST$  
120 IF ST$ <> "2" THEN 110  
130 PRINT #1, "TLK #"; A$
```

【解説】 文例ではスレーブがコントローラに対してSRQを発信した後、自分がトーカー指定されるのを待って、A\$を送出しています。

- ① ステータスバイトは10進のアスキー表現にしてください。
ビット7が“1”で真のステータスバイトとなるためにオフセット値は&H40=64となります。

ステータスバイト2・・64+2→66

- ② POLコマンドの場合、ステータスバイトの予約はありません。
- ③ スレーブのみ発行できます。

● サービスリクエスト待機コマンド

```
PRINT#1, "WQS"
```

【機能】 コントローラが、スレーブからのSRQ受信まで待ちます。

【文例】

```
100 ON COM GOSUB 500
110 PRINT#1, "WQS"
120 .....
.....
500 REM    ...SRQ 割込み処理
510 PRINT#1, "POL 5 6 7" :INPUT#1, DV, ST
520 IF DV=0 THEN 560
530 PRNT#1, "LSN FROM";DV :INPUT#1, DATA$
540 PRINT DATA$
550 GOTO 510
560 RETURN
```

【解説】 ① RS-232C の割込み処理用のコマンドです。

```
100 PRINT#1, "?QS" : INPUT#1, Q$  ↔  100 PRINT#1, "WQS"
110 IF Q$<>"1" THEN 100                110 INPUT#1, Q$
```

上記のプログラム2例は等価です。

- ② このコマンドは、コントローラのみ発行できます。
- ③ “*”（アスタリスク）を送信することにより、待機状態を解除できます。但し、この場合でも現在のステータスは返しますので受信してください。

● サービスリクエスト確認コマンド

```
PRINT#1, "?PD" : INPUT#1, P$
```

【機能】 スレーブの時、発信したSRQが受信されたか否かを問い合わせます。

【文例】

```
100 PRINT#1, "QS 65"  
110 PRINT#1, "?PD" : INPUT#1, Q$  
120 IF Q$<>"0" THEN 110  
130 . . . 次ぎの処理
```

【解説】 ① スレーブがSRQコマンドを発信した後、コントローラが受信したか否かの確認の結果を返します。

```
SRQ受信された      → "0"  
SRQ受信されていない → "1"
```

- ② シリアルポールによるステータスバイトの受け渡しを利用してコントローラになんらかの情報を送信し、データの送信を伴わないような使いかたをする時などに使用します。
- ④ このコマンドは、スレーブのみ発行できます。

● デバイスクリア・コマンド

```
PRINT#1, "DC"  
PRINT#1, "DC d1 d2 . . . dn"
```

【機能】 全デバイスをクリアします。(Device Clear)
指定したデバイスをクリアします。(Selected Device Clear)

【文例】 100 PRINT#1, "DC"
100 PRINT#1, "DC 3 4 5 6 25"

【解説】 セレクテッドデバイスクリアの場合、d1～dnのパラメタによって
デバイス番号を指定します。(n=最大30)
デバイスクリアの場合は、単に“DC”だけを送信します。

- ① 指定されたデバイスは同時にクリアされます。
- ② このコマンドはコントローラのみ発行できます。

デバイスクリア (DC)
バス上の全部のデバイスをクリアします。

セレクテッド・デバイスクリア (SDC)
バス上の任意のデバイスだけをクリアします。

● デバイストリガ・コマンド

```
PRINT #1, "DT d1 d2 . . . dn"
```

【機能】 指定したデバイスをトリガします。(Group Execute Trigger)

【文例】 100 PRINT #1, "DT 3 4 5 6 25"

【解説】 コントローラが指定したデバイスに対しトリガコマンドを送信します。
指定できるデバイスの数は30までです。

- ① 指定されたデバイスは同時にトリガされます。
- ② このコマンドはコントローラのみ発行できます。

● RENライン制御コマンド

```
PRINT#1, "REM"
PRINT#1, "LOC"
```

【機能】 RENラインの“true” “faulse”を選択します。

【解説】 ① REN (remote enable) ラインは、IEEE-488バスに接続されている機器をリモート動作（コントローラからの情報で動作する）かを選択をコントローラがスレーブに与える為のユニラインです。このコマンドを発行するのは、システムコントローラに限ります。

```
REM  → リモート動作
LOC  → ローカル動作
```

② Local Lock Out 及び Go To Local は、機器を指定したうえでリモート/ローカルの選択を与えます。
GPNETはこの為のコマンドは用意していませんが、WBコマンドにより、簡単に送信することができます。

```
PRINT#1, "WB 3F 23 2A 01"
           ↑   ↑   ↑   ↓
           UNL アドレス3   アドレスA   GTLメッセージ
```

```
PRINT#1, "WB 3F 23 2A 11"
           ↑           ↑
           UNL         LLOメッセージ
```


● フォーマットコマンド

PRINT#1, "FMT n" (nは0~7)

【機能】 マルチラインメッセージ最後のUNT, UNL及びEOIの指定をします

【文例】 PRINT#1, "FMT 6"

【解説】 より多くのGP-IB 機器との接続を可能にするために、このコマンドを用意しています。

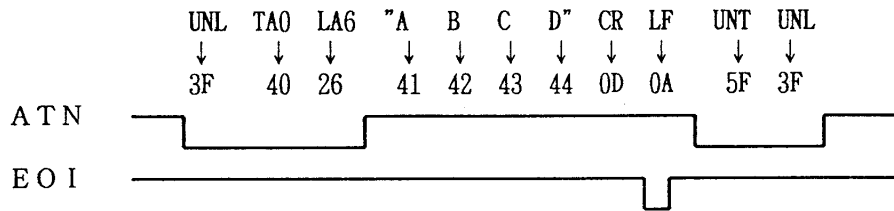
- ① 入出力の最後のUNL, UNT, EOI について制約を加えているGP-IB 機器があるため、コントローラ側がきめこまかくサポートしなければならない場合に有効です。nの意味は下記の通りです。

	UNL	UNT	EOI
FMT 0	0	0	0
1	1	0	0
2	0	1	0
3	1	1	0
4	0	0	1
5	1	0	1
6	0	1	1
7	1	1	1

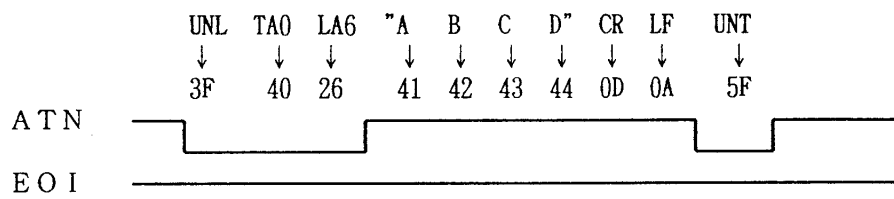
- ② EOIを"1"に設定した場合デリミタ送信と共にEOIを"true"にします。
 DELコマンドによるEOI指定は、最終バイトの送出とともにEOIを"ture"にするものでデリミタは付加しませんが、FMTコマンドはデリミタの最終バイトと共にEOIを"true"にします。

【応用】 PRINT#1, "TLK TO 6#ABCD" を例に解説

① "FMT 7" の場合



② "FMT 2" の場合



● ステータス確認コマンド

```
PRINT #1, "?ST" : INPUT #1, ST$
```

【機能】 自己の状態（ステート）を確認します。

【文例】

```
100 PRINT #1, "?ST" : INPUT #1, ST$
110 IF ST$="0" THEN PRINT " アイドリング"
120 IF ST$="1" THEN PRINT " リスナ"
130 IF ST$="2" THEN PRINT " トーカ"
140 IF ST$="3" THEN PRINT " コントローラ"
```

【解説】 文例で示したように、現在の自己の状態を確認するために使用します。IEEE-488ではバス上に存在するデバイスのステートについて、厳格に規定しています。

トーカ指定されていないデバイスがバスにデータ出力したり、コントローラでないデバイスがDTコマンドを発行したりした場合その動作は保証されません。

GPNET では、内蔵するGP-IB 専用LSI とCPU によってステート管理されていますが、ユーザの責任において、ステートの管理をしなければならぬ時があります。例えば、SRQを発信した後は、いきなりデータを送り出すのではなく、自己がコントローラにトーカ指定されるのを待ってデータ送信しなければなりません。

```
100 PRINT #1, "?ST" : INPUT #1, ST$      ; この時 ST$="0"
110 PRINT #1, "QS 66"
120 PRINT #1, "?ST" : INPUT #1, ST$
130 IF ST$ <> "1" THEN 120                ; トーカ指定を待つ
140 PRINT #1, "TLK #ABCDEFG"             ; データ送出
150 . . . . .
```

● RENライン確認コマンド

```
PRINT#1, "?RM" : INPUT#1, RM$
```

【機能】 RENラインの状態を問い合わせます。

【解説】 RENラインの状態を次のようなステータスで返します。

```
1 : RENラインが "true"  
0 :      "      " "false"
```

● ステータス確認コマンド

```
PRINT#1, "?DC" : INPUT#1, DC$
```

【機能】 DC, SDCを受信したか否かを問い合わせます。

【解説】 DC (デバイスクリア)、SDC (セレクトッド・デバイスクリア)を受信したか否かを問い合わせます。

```
1 : DC, SDCを受信している。  
0 :      "      "      していない。
```

● ステータス確認コマンド

```
PRINT#1, "?DT" : INPUT#1, DT$
```

【機能】 RENラインの状態を問い合わせます。

【解説】 GETを受信したか否かを問い合わせます。

1 : GETを受信している。
0 : " " していない。

● リセットコード変更コマンド

```
PRINT#1, "RC n"
```

【機能】 リセットコード（デフォルト値01）を変更します。

- 【解説】
- ① リセットコードはパワーオン時には01 (BASICH表現でCHR\$(1))が有効ですが、01以外のコードに変更する場合に使用します。
 - ② nの値は10進アスキー表現で、0～255までの範囲です。
 - ③ 変更後は電源切断まで有効です。
 - ④ n="0" のときリセットコードは無効になります。

● 通信同期コマンド

```
PRINT #2, "SYC" : INPUT #1, S
```

【機能】 単にGPNETとホストとの同期をとります。

【文例】

```
100 PRINT #1, "TLK TO 3#";A$  
110 PRINT #1, "DT 3"  
120 PRINT #1, "SYC";INPUT #1, S  
140 A1=INP(23)
```

【解説】 文例の場合、GPNETは120ラインの“SYC”まで一度にバッファし徐々に実行します。コンピュータは120ラインでINPUTの状態、GPNETが処理を終えダミー出力するまで待たされます。140ラインのINP(23)は110ラインのトリガの結果を入力するものと仮定します。

120ラインがない場合、非常に不合理なことが発生します。GPNETがバッファされているコマンドを徐々に実行している間にコンピュータは“DT 3”が当然実行されたとして、140ラインを実行してしまいます。その結果110ラインのトリガが、140ラインのINP(23)の後に出てしまうことになるのです。

このようなことはかなり特殊なのですが、GPNETのユーザで実際発生したケースです。

■モードBのコマンド

モードBは、底面のDIPSW2の8番をONにすることで起動されます。
このモードでは、GPNETはGPIBスレーブとして動作し単純にGPIB \leftrightarrow RS232C間での通信変換器として働きます。

● リセット

モードBのGPNETをリセットするには、GPIB側よりデバイスクリア(DCL)はセレクトッド・デバイスクリア(SDC)を施します。

但し、リセット後20mSEC間は時間をおいて下さい。

● 初期化コマンド

モードBでは、起動直後からリセット直後に1度だけGPIB側より“S”で始まる初期化コマンドを受けることができます。コマンドのフォーマットは次の通りです。

“S n Dd1 d2 d3 X n / m *” + デリミタ

【機能】 GPIB側からGPNETの通信仕様を設定します。

【解説】① “S n”は、サービスリクエストの指定をします。
RS232C機器よりGPNETへデータが転送された時、SRQをマスタコントローラに送信するモードとしないモードが選択できます。

“S 0” SRQを送信します。

“S 1” SRQを送信しません。(デフォルト)

② “Dd1 d2 d3”は、通信デリミタの指定を行います。
d1=GPIB側のデリミタ指定を行います。
d2=GPIB2次デリミタ(モードBではサポートされていません)
d3=RS232Cデリミタ指定を行います。

値	d1	d2	d3
0	CR+LF	常に0	CR+LF
1	CR		CR
2	LF		LF
3	EOI		(注1)

注1: RS232Cデリミタ(d3=3)とした場合、送信データにはデリミタを付加しません。また、受信データの区切りは、タイムアウト(200mSEC固定)となります。

③ “X n / m”はXコントロールの指定をします。

“X 0 / 0” Xコントロール禁止

“X 1 9 / 1 7” &H11/&H13でX制御を行う

↑ほとんどの場合このコードが使われています。

- ④ “*” は必ず最後に必要です。
- ⑤ GPIB側からRS232C側へ出力される場合、GPIBより1バイト入力毎にRS232Cへ出力されます。
- ⑥ RS232C側からGPIB側へ出力される場合、初期設定値により次のようになります。
- a. “S 1” に設定されている場合
RS232C側からのデータをデリミタがくるまでバッファにため込みデリミタがきた後に自分がトーカされれば出力します。
- b. “S 0” に設定されている場合
RS232C側からのデータをデリミタがくるまでバッファにため込みデリミタがきた時点でSRQをアクティブにし、トーカ指定されるのを待って送信します。(ステータスバイトは42H)

【文例】

PC9801シリーズによるモードBプログラム例

```

10 ISET IFC
20 CMD DELIM=0
30 WBYTE &H3F, &H21, &H04;
40 FOR W=1 TO 300:NEXT W
60 PRINT@4;"S1D000X0/0*"
70 ^
80 PRINT@4;"This is message."
90 INPUT@4;DAT$
100 PRINT DAT$ :GOTO 80
110 END

```

HP9000シリーズによるモードBプログラム例

```

10 CLEAR 704
11 WAIT 0.2
13 OUTPUT 704;"S1D000X0/0*"
14 ^
15 OUTPUT 704;"This is message."
16 ENTER 704;Dat$
17 PRINT Dat$
18 GOTO 15
19 END

```


第4章 付録

■モードA：コマンド一覧表

初期化コマンド			
CHR\$(1) IFC		system reset InterFace Clear	GPNET リセット インタフェースクリア
通信制御・設定コマンド			
DEL n1 n2 n3 XC n1 n2 SYC TIME n ?TIME	I	set DELimiter set X-Control SYnChronous set TIMEout TIME out?	デリミタ設定 X制御の指定 同期用コマンド タイムアウトセット タイムアウト発生か?
データ通信コマンド			
TLK TO d1 d2... TML TO d1 d2... TMS TO d1 d2... LSN FROM d LML FROM d LB FROM d LB* FROM d LB@ FROM d TB TO d WB hex hex/hex hex WB hex hex#ABC RB hex hex	O O O I I I I I O I	TaLK TO device Talk Multi Line " (Sync) LiStEN FROM dev Listen Multi Line Listen Binary " " Talk Binary Write Byte " Read Byte	単行データの送信 複行データの送信 " (同期型) 単行データの受信 複行データの受信 バイナリデータの受信 " " バイナリデータ送信 マルチラインメッセージ " "
ポーリングコマンド			
PDV d1 d2...dn POL d1 d2...dn AP AP IF QS ?QS QS n WQS ?PD	I I I I I I I	Polling DeVice table Polling Auto Polling AP IF SRQ line true SRQ line true ? Service Request Wait until SRQ true Pend SRQ	デバイステーブル セット シリアルポール オートシリアルポール " (SRQが真の時) サービスリクエスト有り? " 発信 " あるまで待つ " 受信されたか
汎用コマンド			
DC d1 d2...dn DT d1 d2...dn REM LOC FMT n ?ST ?RM ?DC ?DT RC n	I I I I I I I I I	Device Clear Device Trigger REMOte LOCAl ForMaT command get SStatus number ReMOte enable? Dvice Cleared? Dvice Triggered? Reset Code set	デバイスクリア デバイストリガ RENラインtrue RENラインfalse UNL, UNT, EOT の指定 現在のステータスを聞く RENラインtrueか? DCL, SDC 受信したか? GET を受信したか? RESET コードの変更

⓪は送信、Iは受信を伴います。

GPNET model10+ 製品仕様

■ハードウェア構成

- CPU HD647180X-6 (f = 6.144MHz)
- ROM/RAM ROM 16Kbyte (CPU内臓)
 ROM 8Kbyte
 RAM 512byte (CPU内臓)
 RAM 32Kbyte
- SIO ASCII 1CH (CPU内臓)
- GPIB μ PD7210

■通信機能

- シリアル RS232C
 調歩同期式全二重通信
 速度 (300, 600, 1200, 2400, 4800, 9600, 19200, 38000bps)
 DCE
- GPIB IEEE-488準拠
 速度 (最大1Mbyte/SEC)

■電源仕様

- 入力電源 DC + 6V mA

■動作環境

- 動作温度 0°C ~ 50°C

GPNET model20+ 製品仕様

■ハードウェア構成

- CPU HD64180S (f = 8MHz)
- ROM/RAM ROM 32Kbyte
RAM 32Kbyte
- SIO ASCII 1CH (CPU内臓)
- GPIB μPD7210

■通信機能

- シリアル RS232C
調歩同期式全二重通信
速度 (300, 600, 1200, 2400, 4800, 9600, 19200, 38000bps)
DCE
- GPIB IEEE-488準拠
速度 (最大1Mbyte/SEC)

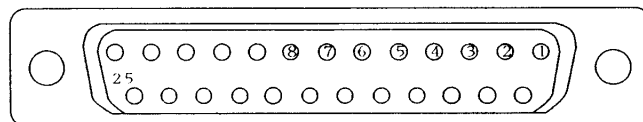
■電源仕様

- 入力電源 AC100V

■動作環境

- 動作温度 0℃~50℃

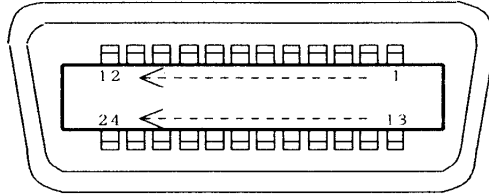
GPNET model110+/model120+ RS232Cコネクタ



GPNETのRS232Cコネクタを正面から見た図

端子	名称	説明
1	FG	フレーム・グラウンド
2	RXD	受信データ (GPNET が受信するデータ)
3	TXD	送信データ (GPNET から送信するデータ)
4	CTS	Clear To Send (GPNET が入力する送信許可信号)
5	RTS	Request To Send (GPNET が出力する送信許可信号)
6	DR	常時 +12Vを出力
7	SG	シグナル・グラウンド
8	CD	常時 +12Vを出力
25		ロジック電源 (光バージョン用DC5V出力)

GPNET model110+/model120+ GPIBコネクタ



GPNET GP-IBコネクタを正面から見た図

端子	名称	端子	名称
1	DIO1	13	DIO5
2	DIO2	14	DIO6
3	DIO3	15	DIO7
4	DIO4	16	DIO8
5	EOI	17	REN
6	DAV	18	GND
7	NRFD	19	GND
8	NDAC	20	GND
9	IFC	21	GND
10	SRQ	22	GND
11	ATN	23	GND
12	シールド	24	ロジックGND

USASCII Character Set

as modified for printing
and the inclusion of Meta characters

Decimal	Octal	Hex	Graphic	Name (Meaning) <English Text Reference>
0.	000	00	^@	NUL (used for padding) <NULL>
1.	001	01	^A	SOH (start of header)
2.	002	02	^B	STX (start of text)
3.	003	03	^C	ETX (end of text)
4.	004	04	^D	EOT (end of transmission)
5.	005	05	^E	ENQ (enquiry)
6.	006	06	^F	ACK (acknowledge)
7.	007	07	^G	BEL (bell or alarm) <BELL>
8.	010	08	^H	BS (backspace) <BS>
9.	011	09	^I	HT (horizontal tab) <TAB>
10.	012	0A	^J	LF (line feed) <LF>
11.	013	0B	^K	VT (vertical tab)
12.	014	0C	^L	FF (form feed, new page) <FF>
13.	015	0D	^M	CR (carriage return) <CR>
14.	016	0E	^N	SO (shift out)
15.	017	0F	^O	SI (shift in)
16.	020	10	^P	DLE (data link escape)
17.	021	11	^Q	DC1 (device control 1, XON)
18.	022	12	^R	DC2 (device control 2)
19.	023	13	^S	DC3 (device control 3, XOFF)
20.	024	14	^T	DC4 (device control 4)
21.	025	15	^U	NAK (negative acknowledge)
22.	026	16	^V	SYN (synchronous idle)
23.	027	17	^W	ETB (end transmission block)
24.	030	18	^X	CAN (cancel)
25.	031	19	^Y	EM (end of medium)
26.	032	1A	^Z	SUB (substitute)
27.	033	1B	^[ESC (escape, alter mode, SEL) <ESC>
28.	034	1C	^\ ^_	FS (file separator)
29.	035	1D	^]	GS (group separator)
30.	036	1E	^^	RS (record separator)
31.	037	1F	^-	US (unit separator)
32.	040	20		space or blank <SP>
33.	041	21	!	exclamation mark
34.	042	22	"	double quote
35.	043	23	#	number sign (hash mark)
36.	044	24	\$	dollar sign
37.	045	25	%	percent sign
38.	046	26	&	ampersand sign
39.	047	27	'	single quote (apostrophe)
40.	050	28	(left parenthesis
41.	051	29)	right parenthesis
42.	052	2A	*	asterisk (star)
43.	053	2B	+	plus sign
44.	054	2C	,	comma
45.	055	2D	-	minus sign (dash)
46.	056	2E	.	period (decimal point, dot)
47.	057	2F	/	(right) slash

48.	060	30	0	numeral zero
49.	061	31	1	numeral one
50.	062	32	2	numeral two
51.	063	33	3	numeral three
52.	064	34	4	numeral four
53.	065	35	5	numeral five
54.	066	36	6	numeral six
55.	067	37	7	numeral seven
56.	070	38	8	numeral eight
57.	071	39	9	numeral nine
58.	072	3A	:	colon
59.	073	3B	;	semi-colon
60.	074	3C	<	less-than sign
61.	075	3D	=	equal sign
62.	076	3E	>	greater-than sign
63.	077	3F	?	question mark
64.	100	40	@	atsign
65.	101	41	A	upper-case letter ALPHA
66.	102	42	B	upper-case letter BRAVO
67.	103	43	C	upper-case letter CHARLIE
68.	104	44	D	upper-case letter DELTA
69.	105	45	E	upper-case letter ECHO
70.	106	46	F	upper-case letter FOXTROT
71.	107	47	G	upper-case letter GOLF
72.	110	48	H	upper-case letter HOTEL
73.	111	49	I	upper-case letter INDIA
74.	112	4A	J	upper-case letter JERICH0
75.	113	4B	K	upper-case letter KAPPA
76.	114	4C	L	upper-case letter LIMA
77.	115	4D	M	upper-case letter MIKE
78.	116	4E	N	upper-case letter NOVEMBER
79.	117	4F	O	upper-case letter OSCAR
80.	120	50	P	upper-case letter PAPPA
81.	121	51	Q	upper-case letter QUEBEC
82.	122	52	R	upper-case letter ROMEO
83.	123	53	S	upper-case letter SIERRA
84.	124	54	T	upper-case letter TANGO
85.	125	55	U	upper-case letter UNICORN
86.	126	56	V	upper-case letter VICTOR
87.	127	57	W	upper-case letter WHISKEY
88.	130	58	X	upper-case letter XRAY
89.	131	59	Y	upper-case letter YANKEE
90.	132	5A	Z	upper-case letter ZEBRA
91.	133	5B	{	left square bracket
92.	134	5C	\	left slash (backslash)
93.	135	5D	}	right square bracket
94.	136	5E	^	uparrow (carat)
95.	137	5F	_	underscore

96.	140	60	`	(single) back quote (grave accent)
97.	141	61	a	lower-case letter alpha
98.	142	62	b	lower-case letter bravo
99.	143	63	c	lower-case letter charlie
100.	144	64	d	lower-case letter delta
101.	145	65	e	lower-case letter echo
102.	146	66	f	lower-case letter foxtrot
103.	147	67	g	lower-case letter golf
104.	150	68	h	lower-case letter hotel
105.	151	69	i	lower-case letter india
106.	152	6A	j	lower-case letter jericho
107.	153	6B	k	lower-case letter kappa
108.	154	6C	l	lower-case letter lima
109.	155	6D	m	lower-case letter mike
110.	156	6E	n	lower-case letter november
111.	157	6F	o	lower-case letter oscar

112.	160	70	p	lower-case letter pappa
113.	161	71	q	lower-case letter quebec
114.	162	72	r	lower-case letter romeo
115.	163	73	s	lower-case letter sierra
116.	164	74	t	lower-case letter tango
117.	165	75	u	lower-case letter unicorn
118.	166	76	v	lower-case letter victor
119.	167	77	w	lower-case letter whiskey
120.	170	78	x	lower-case letter xray
121.	171	79	y	lower-case letter yankee
122.	172	7A	z	lower-case letter zebra
123.	173	7B	{	left curly brace
124.	174	7C		vertical bar
125.	175	7D	}	right curly brace
126.	176	7E	~	tilde
127.	177	7F	^?	DEL (delete, rub out)

128.	200	80	~^@	Meta NUL
129.	201	81	~^A	Meta SOH
130.	202	82	~^B	Meta STX
131.	203	83	~^C	Meta ETX
132.	204	84	~^D	Meta EOT
133.	205	85	~^E	Meta ENQ
134.	206	86	~^F	Meta ACK
135.	207	87	~^G	Meta BEL

...

159.	237	9F	~^_	Meta US
160.	240	A0	~ -	Meta space
161.	241	A1	~!	Meta exclamation mark

...

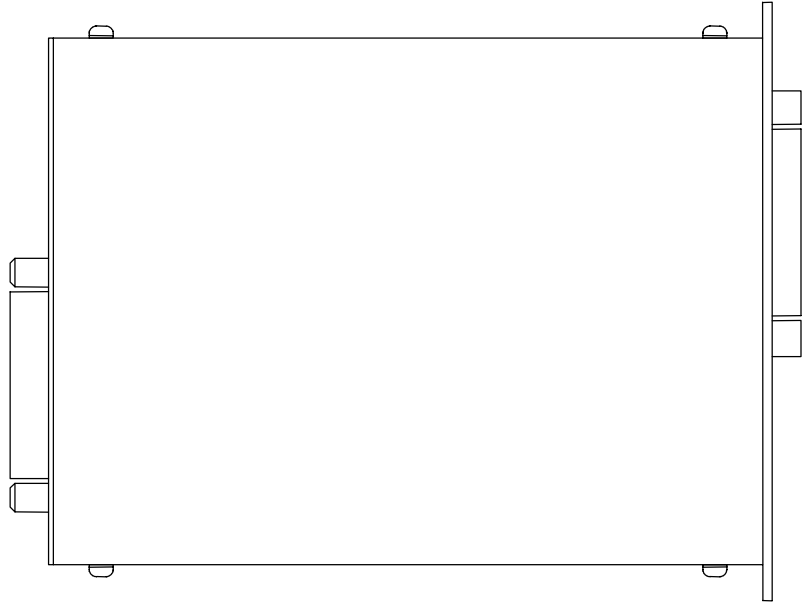
253.	375	FD	~}	Meta right curly brace
254.	376	FE	~~	Meta tilde
255.	377	FF	~^?	Meta DEL

Notes:

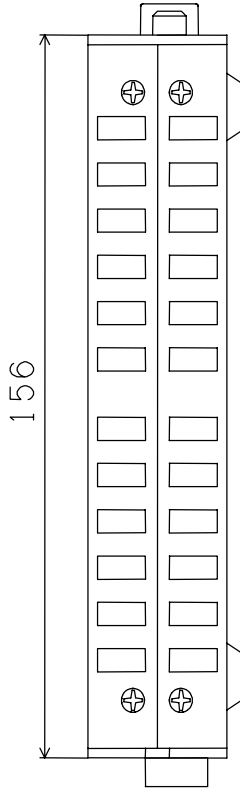
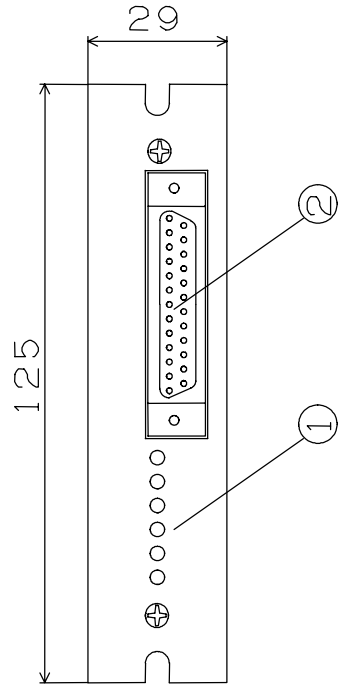
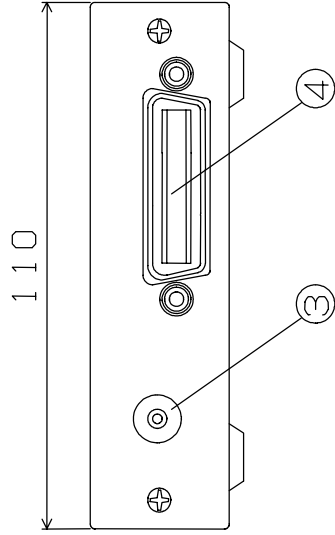
The "Meta" form of each character is created by adding 128 (decimal) to that character's ASCII value.

To prevent ambiguity, the following alternate forms can be used for printing:

94.	136	5E	^	can be printed as	^=
126.	176	7E	~	can be printed as	^~
222.	336	DE	^^	can be printed as	^^=
254.	376	FE	^^	can be printed as	^^~

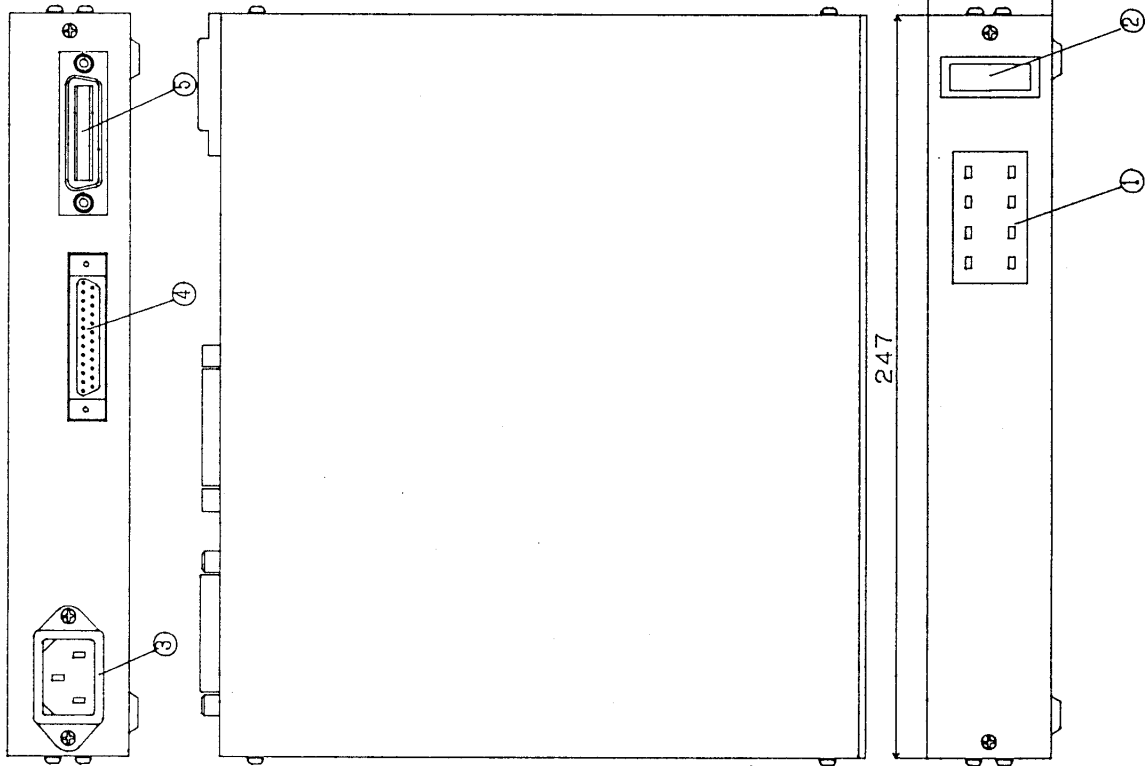


- | | |
|---|------------|
| 1 | LEDモニタ |
| 2 | RS232Cコネクタ |
| 3 | DCジャック |
| 4 | GB-1Bコネクタ |



品名 GPNET model 10+R 外装図 1 1 平成5年 1月20日 付

ネットワークサブライ



1	LEDモニタ
2	メインスイッチ
3	ACインレット
4	RS232Cコネクタ
5	GP-1Bコネクタ

品名	GPNET model 20+	R	1	1	1	平成5年
	外筐図	度				3月1日

ネットワークサーバ