



Title	PC-8801を用いたTSSグラフィック端末プログラム
Author(s)	国本, 雅夫
Citation	大阪大学大型計算機センターニュース. 1984, 52, p. 99-112
Version Type	VoR
URL	<a href="https://hdl.handle.net/11094/65600">https://hdl.handle.net/11094/65600</a>
rights	
Note	

*The University of Osaka Institutional Knowledge Archive : OUKA*

<https://ir.library.osaka-u.ac.jp/>

The University of Osaka

# PC-8801を用いたTSSグラフィック端末プログラム

大阪大学工学部 国本雅夫

## §1. はじめに

パーソナルコンピュータ PC-8801は、大型計算機センターの端末として盛んに利用されているが、さらにグラフィックディスプレイ端末としても利用できれば便利このうえない。折しも本誌 Vol.13, No.2 に小田晃一氏が FM-8 を用いたグラフィック端末プログラムを発表され<sup>1)</sup>、また、藤井博氏がパーソナルコンピュータを大型計算機センターの端末にする方法について記事を書かれた<sup>2)</sup>ので、これらの資料を参考にして PC-8801 用スクリーンエディット機能付きグラフィック端末プログラムを作製した。ただし、小田氏による記事はセンターに登録されている图形ライブラリのうち、ライブラリファイル名 LIB / GPLT を使用するという前提で書かれたものであり、統合化ライブラリ LIB / ABLIB や FORTRAN 77(V) 用ライブラリ LIB / GPLTV を用いた場合にはこの記事だけではグラフィック端末プログラムを作成することはできなかった。そこで、新たに LIB / ABLIB や LIB / GPLTV を利用した場合にセンターから送られてくる图形データの解析を試み、PC-8801 の特長の 1 つである 640 × 400 ドットの高解像度ディスプレイ機能を利用したグラフィック端末プログラムを作製した。その結果、PC-8801 をグラフィック端末として用いることに成功し、プリンター (EPSON MP-80 type II) を利用してグラフィック出力結果のハードコピーをとることもできるようになった。

そこで、グラフィックデータの処理方法について解析した結果を示し、さらに本グラフィック端末プログラムの使用法についての説明を行いたいと思う。

## §2. 機器構成と機能

本プログラムの対象システムを図 1 に示す。ディスクは工人舎 KD-276A 以外(例えば、PC-8031-2W など)

でも全く問題ないものと思わ

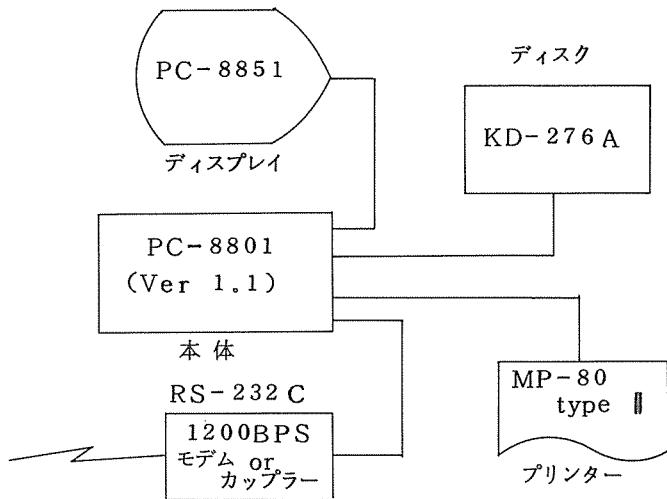


図 1. システム構成

れる。ディスプレイは  $640 \times 400$  ドットを表示できるものでなければならない。RS-232C については 300 BPS のカップラーでも使用可能であるが、グラフィック端末としては 300 BPS ではデータ転送速度が遅すぎて実用的ではない。逆に、9600 BPS では BASIC プログラムを一部追加することにより使用可能になる（§ 6-5 参照）。プリンターについては EPSON MP-80 以外でも使用可能であるが、その場合、本プログラムに組み込まれたグラフィック画面のハードコピープログラムを一部修正する必要があると思われる。なお、この MP-80 type II には 32K bytes のバッファ（MELCO 製 PB-32）を取り付けてあるので 1200 BPS でも全く問題なく交信内容のプリント出力を行うことができる。

本プログラムの開発に用いた PC-8801 は、BASIC・ROM が Ver 1.1、システムディスクが [Aug 20, 1982] である。BASIC・ROM については他に Ver 1.0 システムディスクについては他に [Feb 4, 1982], [Apr 24, 1982] などがあるが、これらのバージョンにおいても本プログラムが有効であるかどうかについては、確認できなかった。なお、電源投入時に CRT に表示されるオープニングメッセージによりバージョンの確認を行うことができる。

次に、本プログラムの特長を以下に示す。

- 1) N6922 グラフィックディスプレイターミナルとして動作することができる。
- 2) グラフィック出力結果のハードコピーをとることができる。
- 3) 複数行にわたるクリーンエディットを行うことができる。

### § 3. 図形データの解説

図形データの解説については文献 1) に詳しく述べられているが、§ 1 でも述べたように、この記事だけではグラフィック端末プログラムを作成することはできない。なぜなら LIB/GP LT では図形データは必ず 5 バイト 1 組で送られてくるが、LIB/GPLTV, LIB/ABLIB では必ずしも 5 バイト 1 組ではなく、4 バイト 1 組の図形データや 3 バイト 1 組のものなども存在するからである。そこで、これらの短縮された図形データの意味について述べる。ただし、基本的な図形データの意味については、小田氏の記事を参考にして欲しい。

#### 1) グラフィック命令

N6922 に対するグラフィック命令は ASCII "GS" コード、つまり  $1D_H$  である<sup>3)</sup>。 $1D_H$  以後、グラフィックリセット命令までのデータは図形データであり（ただし、タイムフィラー  $16H$  は除く）、さらに  $1D_H$  の直後の 5 バイトは必ず図形データとなっている。この 5 バイトの図形データを座標データに変換する方法については文献 1) に詳しく述べられているので、ここでは省略する。

## 2) グラフィックリセット命令

グラフィックリセット命令は、ASCII "US" ( $1F_H$ ) , "CR" ( $0D_H$ ) "ESC" ( $1B_H$ ) , "FF" ( $0C_H$ )である<sup>3)</sup> これらの命令以後のデータは図形データではなく、単なる文字データであると見なす。

基本的には、図形データは5バイトで1組になっているが、前述のようにLIB/GPLTVやLIB/ABL LIBを用いた場合には、ある図形データの組に続く次の図形データの組の中にその直前の組と同じデータが存在すると、その図形データは省略される。

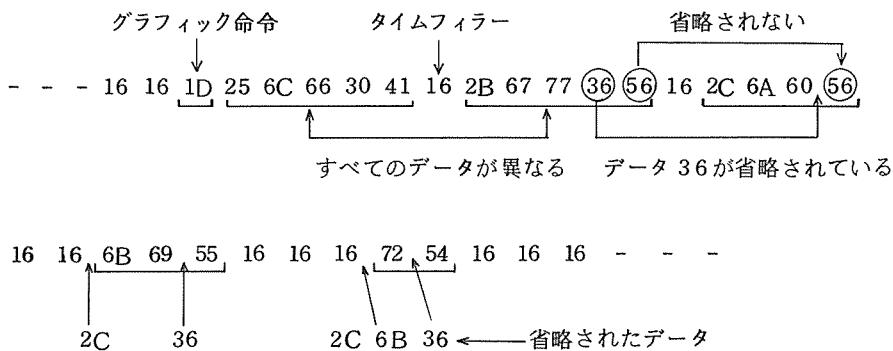


図2. LIB /GPLTVを用いた場合の図形データの例

ただし、X座標の下位データのバイトは直前の組のデータと同じであっても省略されることはない。また、2～4バイト1組のデータにおいてはY座標の下位データのバイトも省略されない。

以上のことまとめると、センターから送られてくる図形データの組は、図3のように整理することができる。この図を見てわかるように、1, 2, 5バイト長データの場合は、そのデータが基本型の何バイト目に相当するデータであるかをあらかじめ知ることができるが、3, 4バイト長データの場合には、そのデータが基本型の何バイト目のデータであるかを判別する必要が生じる。これを行うには、図形データのビットのうち、座標情報をもたないビット6を利用する。図4(a)に示すように各図形データのビット6は、Y座標上位バイト、X座標上位バイトならば必ず0、それ以外は必ず1になっている。よって、図4(b)のようにすれば、その図形データの組がどのタイプであるかを判別することができる。なお、センターから送られてくるデータは7ビット長であるので、図4(a)においてビット7は全く意味をもたない。

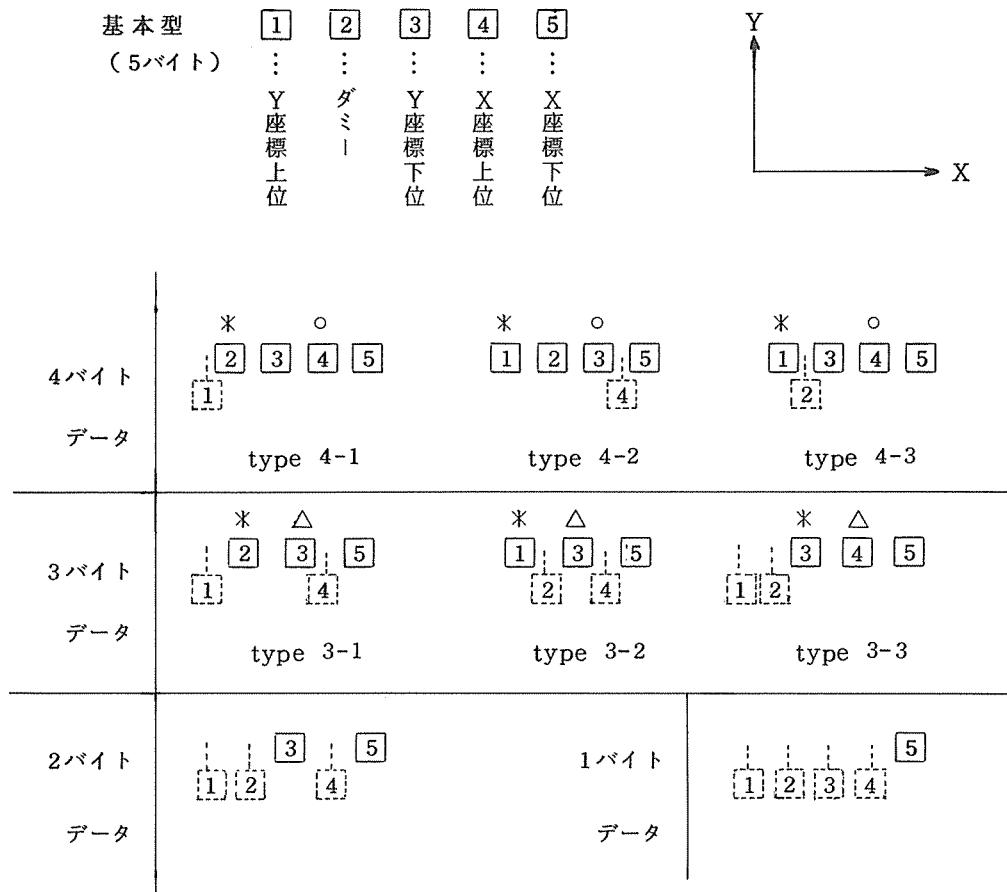


図3. 図形データの組み合せ方

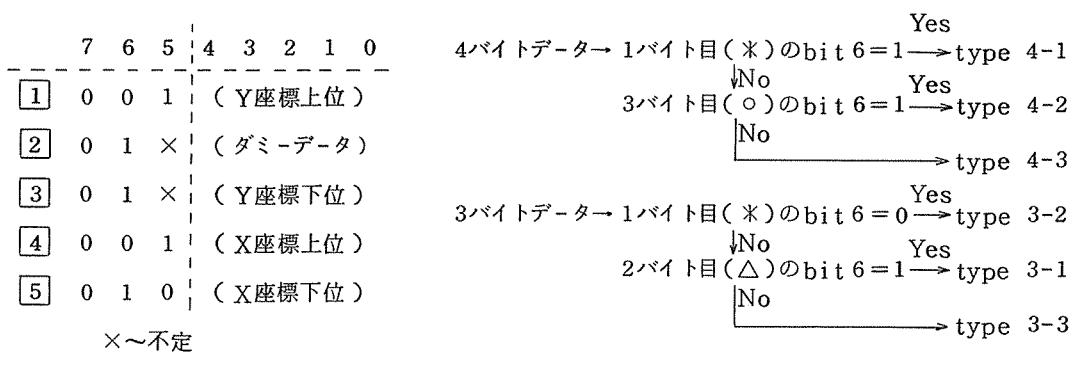


図4. データタイプの判別法

#### § 4. データ処理のための基本アルゴリズム

§ 3 および文献 1 ) をもとにすれば、センターから送られてくる図形データの意味を理解することができるが、これらの情報を基にして実際に図形データを処理する際には、その図形データが何番目のデータであるかを示すフラグを用意する必要がある。なぜなら、 $R\$ = INPUT\$$  ( $LOC(1), \#1$ ) 等によって RS-232C 用バッファーから取り出されるデータの長さは不定であり、このデータをマシン語で逐次処理する際にはデータの順位を記録しておく必要があるからである。本プログラムでは GFLG というものを用意した。この GFLG は 0 ~ 11 の値をとるが、 $GFLG = 1 \sim 10$  に対してはそれぞれの値に対応する 1 バイトのメモリを確保した。データ処理ルーチンでは、この GFLG を逐次参照して必要な処理を行うようにしている。例えば、 $GFLG = 0$  ならば、 $1D_H, 1B_H$  の検索、 $7F_H$  の除去等を行う (LEVEL 0)。また、 $GFLG = 1 \sim 5$  ならば、その時点のデータはグラフィック命令  $1D_H$  の直後の 5 バイト 1 組の図形データのうちの 1 つであることがわかるので、主にそのデータを記録することになる (LEVEL 1)。さらに、 $GFLG > 5$  ならば各図形データのバイト長に応じた処理を行い、実際に線を描く (LEVEL 2)。

これらの図形データの処理の流れを具体的に示したものが図 5 である。LEVEL 2 の処理においては、1 ~ 4 バイト長の図形データを基本型の 5 バイト長データに戻している。せっかく短縮されて節約されているデータをもう一度再現していく不経済に見えるかもしれないが、これは、N6922 用の  $1024 \times 780$  格子点の座標値を PC-8801 用の  $640 \times 390$  の座標値に変換するサブルーチンを共用化するためである。

#### § 5. プログラムについて

本プログラムは、基本的には N88-BASIC を使用している。ただし、BASIC では実現できない機能、および BASIC では処理に時間がかかりすぎる部分はマシン語を使用している。

本プログラムを PC-8801 に打ち込む際には、まず BASIC 部分を打ち込み、適当なファイルネーム ( 例えば、"TSS-G.V23" など ) でセーブする。次に、CLEAR , & HE1FF ↴ とした後、モニタに入りマシン語部分を打ち込む。蛇足ながら、マシン語部分を打ち込む際には S コマンドを用いるよりも E コマンドを用いる方が便利である。マシン語部分を打ち込み終わったら **GTRL** + **B** で BASIC に戻り、BSAVE "TSSG23.OBJ" , & HE200 , & H400 ↴ としてマシン語をセーブする。できればチェック・サムの確認をしておいたほうがよい。

次に、本プログラムの使用法について説明する。本プログラムを起動する際には、RUN" ( ファイルネーム ) " ↴ と打ち込む。すると、自動的にマシン語部分がロードされ、初期設定等が

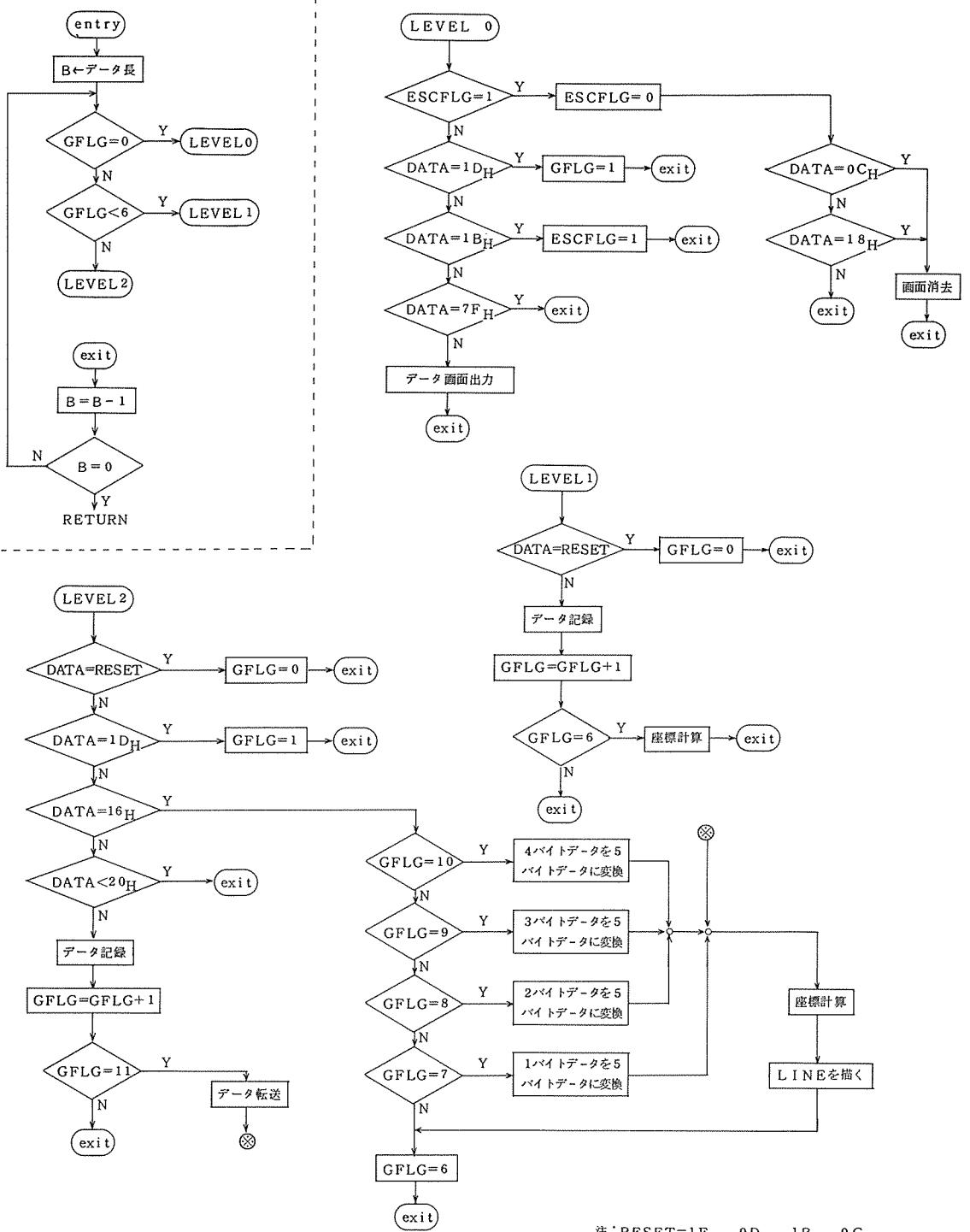


図5. 図形データ処理のためのフローチャート

行われた後、Hit [ f - 1 ] key !!! と表示されるので、[ f - 1 ] key を押してセンターと接続する。あとは、課題番号等を送信すればよい。マシン語の書き込みにDATA文を用いずにBLOAD文を使用しているのは、初期設定に必要な時間となるべく短くして、待ち時間を減らすためである。通常は単なるキャラクターディスプレイ端末として動作し、図形データがセンターから送られてきた場合には自動的にグラフィックディスプレイを開始する。一画面ごとにRETURN keyを押す必要がない分だけN6922よりも使用感は良いはずである。

### 1) スクリーンエディットの方法

ビルド・モード(つまり、\*が表示された状態)において、修正したい行を画面に表示してから(例：LIST 100-300) TAB key を押す。すると、カーソルが画面左上隅に移動するから、修正したい行にカーソルを移動させて修正を行う。一行修正し終ったら、RETURN key を押して修正した行をセンターに送信する。他にも修正したい行があれば、カーソルを移動させて同様にすればよい。修正がすべて終了したならば、STOP key を押してスクリーンエディットモードから抜け出す。STOP key を押すかわりに RETURN key を続けて2度押してもよい。スクリーンエディット中には画面最下行のfunction keyの表示を消すようにしてあるので、スクリーンエディット中であることを確認することができる。なお、このスクリーンエディターは行の先頭に行番号がついていないものは修正できないので、2行にわたる文が存在するようなプログラムの場合は、あらかじめ\$FORMコマンド等によりFORM形式に再編成しておく必要がある。

### 2) 画面消去の方法

N6922ではグラフィック出力中に BREAK key を押すと自動的に画面消去されるが、本プログラムでは画面消去されない。というのは、PC-8801を単なるキャラクターディスプレイ端末として使用している場合には、BREAK key (PC-8801では STOP key) を押すことにより画面消去されては困るからである。本プログラムを用いてグラフィック出力中に STOP key を押すと、CRTにはグラフィックスの一部が表示されたままになるが、このような場合には CLR key を押すことにより画面消去を行うことができる。つまり、CLR key が N6922 の RESET PAGE key に対応しているわけである。

### 3) function key について

各 function key には以下に示すような機能をもたせてある。

- [ f - 1 ] ..... LOG-ON , " \$\$CON , TSS , , ASC" を送信する。(グラフィックディスプレイを行う場合でも、"——, , GDP" とする必要はない。)
- [ f - 2 ] ..... LP 80 SW , プリンター出力(80桁)スイッチ。このキーを押すことにより、センターとの交信内容をプリント出力するかどうかを選択することが

できる。なお、このキーは反転キーになっており、キーを押すたびに LP  
80 OFF  $\longleftrightarrow$  LP 80 ON が表示と共に切り換わる。

[ f - 3 ] ..... LP 132 SW, プリンター出力(132行)スイッチ。132行でプリンタ  
ー出力されることを除けば、[ f - 2 ] key と同じ機能である。

[ f - 4 ] ..... LIST, キーボードから "LIST" と打ち込んだのと同じ効果がある。

[ f - 5 ] ..... RUN<sub>C</sub><sub>R</sub>, キーボードから "RUN" と打ち込んだのと同じ効果がある。

[ f - 6 ] ..... TEXT COPY, 画面コピー(ただし、キャラクターのみ)を行う。

[ f - 7 ] ..... GRAPH COPY, グラフィック画面のコピーを行う。

[ f - 8 ] ..... (ユーザー定義用)

[ f - 9 ] ..... (ユーザー定義用)

[ f - 10 ] ..... TERMINATE, 本プログラムを終了し、BASIC のコマンドレベルに戻  
る。通常は、このキーを押す必要はない。

センターから画面消去命令が送られてくると、グラフィックディスプレイに備えて function key の表示も消去するようにしているが、この場合 [ SHIFT ] key を押すことにより function key の表示を再開する。

#### 4) Back Space, Break について

バックスペースは [ key ], [ DEL ] key , [ CTRL ] + [ H ] key いずれも可能である。  
また、ブレークは [ STOP ] key を押せばよい。

## § 6. その 他

### 1) PC-8801本体後部のディップスイッチについて

SW1-6は、必ずON(DELコードを処理する)に設定しておくこと。このスイッチが  
OFF(DELコードを無視する)になっていると、図形データとしての<sub>7FH</sub>まで無視される  
ため、グラフィック出力結果がおかしくなってしまう。

### 2) プリンター出力について(§2参照)

[ f - 7 ] key を押してもグラフィック画面のハードコピーを取ることができない場合は、  
本プログラムのBASIC部分の870行を COPY 2 あるいは COPY 3 に変更するか、  
マシン語部分を一部書き替えるかする必要がある。

### 3) ハードコピー指定について

本プログラムでは、プログラムの構成の都合上、CALL VSTERM, CALL HDCOPY 等によるハードコピー命令は無視するようにしてあるため、プログラムの途中でハードコピーを行いたい場合はREAD文, PAUSE文等によってプログラムを一時的に停止させた後、[ f ]

- 7 ] key を押す必要がある。

#### 4) 画面の縦横比率について

N 6922 の管面と、PC-8851 の CRT とでは縦横比率が異なるため、円を描くようなプログラムを本システムを用いて実行すると 縦 : 橫 = 0.8 : 1 程度の橢円になってしまう。マシン語ルーチンを一部改造することにより 1 : 1 の円を描くようにもできるが、そうすると横方向の解像度が落ちてしまうためマシン語ルーチンの改造は行っていない。どうしても真円を描かせたければ、FORTRAN プログラムのグラフィックルーチンにおいて Y 軸方向が 1.2 倍になるようとするか、あるいはマシン語部分の E58DH - E597H を 00H に変更すれば、ほぼ真円を描くようになる。ただし、一般に X 軸 - Y 軸 を描くようなグラフィックスでは 縦 : 橫 = 0.8 : 1 でも問題はないと思う。さらに、本プログラムに組み込まれたグラフィック画面のハードコピープログラムを用いると、CRT 上では 0.8 : 1 でもハードコピー出力結果は ほぼ 1 : 1 に戻るようになっている。

#### 5) 本プログラムを 9600 BPS で使用する方法

ACOS の R10.1 より使用可能になったフロー制御機能を利用することにより、本プログラムを 9600 BPS でも使用することができる。ただし、ブレーク信号送信の際に Line buffer overflow が起こるのを防ぐために、9600 BPS の場合に限り以下の BASIC プログラムを追加する必要がある。

```
505 OUT &HE6, 3  
515 OUT &HE6, 7
```

### § 7. おわりに

本プログラムでは、グラフィックディスプレイ機能に重点を置いたため、従来の PC-8801 用ターミナルプログラムにあったファイルハンドラールーチンはすべて削除してあるが、利用者の方によってはこの機能が必要な場合もあると思う。その場合は、ファンクションキーの設定等を適宜変更していただきたい。

本文を書くにあたって、いろいろ御指導頂いた大阪大学工学部産業機械工学科・嘉納秀明助教授ならびに大阪大学大学院産業機械工学専攻・小田晃一氏に厚く感謝いたします。

### 〔参考文献〕

- 1) 小田晃一：「マイコンによる TSS グラフィック端末」大阪大学大型計算機センターニュース，Vol.13, No.2, P.79, 1983
- 2) 藤井 博：「パーソナルコンピュータを大型計算機センターの端末に」大阪大学大型計算機

3) 「図形処理基本システム説明書〈GDSP-2/4/6 PLOT〉」

日本電気株式会社

4) 「PC-Techknow 8800 Vol.1」株式会社 アスキー

5) 「PC-8801 マシン語入門」株式会社 アスキー

6) 「PC-8801 ユーザーズ・マニュアル」 日本電気株式会社

7) 「PC-8801 リファレンス・マニュアル」 日本電気株式会社

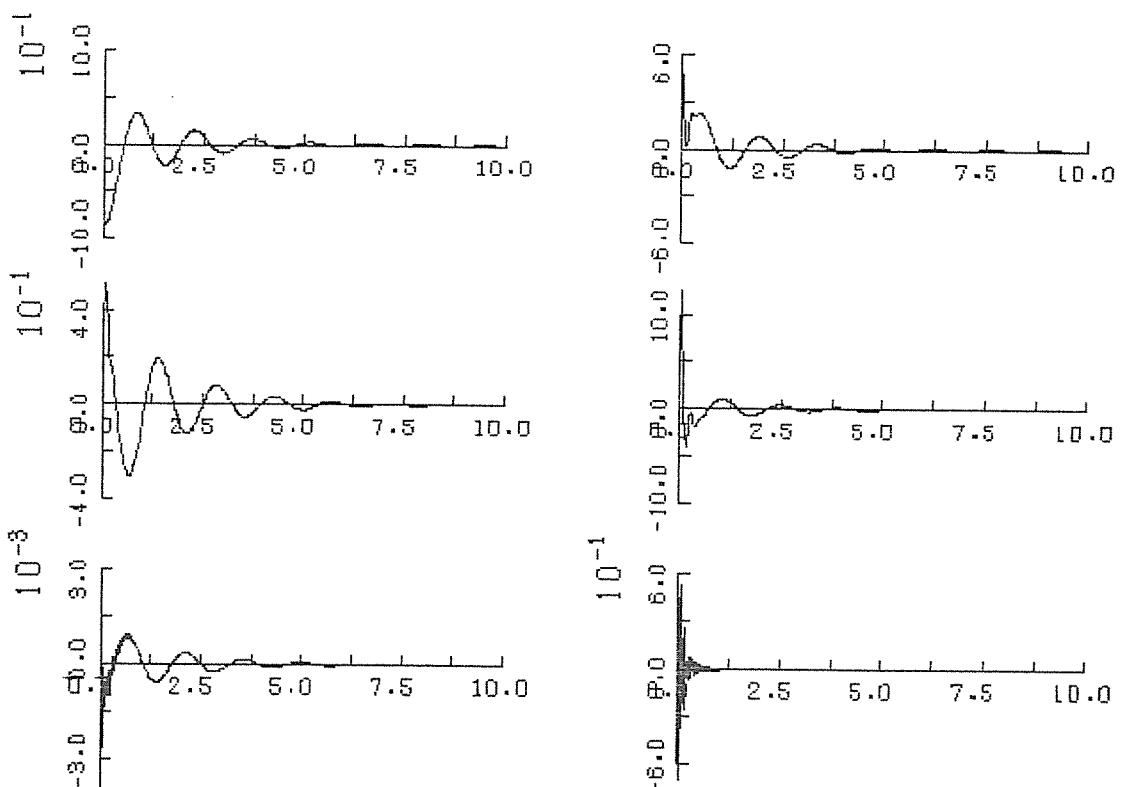


図6. 本プログラムを用いたグラフィック出力例

```

10 '***** TSS TERMINAL PROGRAM WITH SCREEN EDIT ROUTINE ****
20 '** AND WITH GRAPHIC DISPLAY ROUTINE **
30 '** BY M.KUNIMOTO 11/22/83 **
40 '** Ver 2.3 12/19/83 **
50 '**
60 '*****
70 WIDTH 80,25:PRINT "***** TSS-G Ver 2.3 Copyright (C) 1983 by Masubuchi Lab
. *****":PRINT :PRINT "[TABJ key --- SCREEN EDIT":PRINT "[CLRJ key --- RE
SET PAGE":PRINT
80 CLEAR ,&HE1FF:BLOAD "TSSG23.OBJ",R
90 CONSOLE 0,25,0,0:SCREEN 2,2
100 KEY 1,"LOG-ON":KEY 2,"LP 80 (OFF)":KEY 3,"LP 132(OFF)":KEY 4,"LIST ":KEY 5,
RUN"+CHR$(13):KEY 6,"TEXT COPY":KEY 7,"GRAPH COPY":KEY 8,"":KEY 9,"":KEY 10,"TER
MINATE"
110 DEFINT A-Z
120 CURSOR=&H4290:HSCLS2=&H866B:GRAPH.COPY=&H8687:S.E=&HE2CB:TEXT.COPY=&HE2F1:GR
APHIC=&HE31B
130 LISTING=0:LY=0:LX=0:LY1=0:LX1=0
140 BS$=CHR$(&H8):TAB$=CHR$(&H9):CL$=CHR$(&HC):CR$=CHR$(&HD):FS$=CHR$(&H1C):GS$=
CHR$(&H1D):RS$=CHR$(&H1E):US$=CHR$(&H1F):DEL$=CHR$(&H7F)
150 CALL HSCLS2:SCREEN ,0
160 CONSOLE ,1
170 OPEN "COM:E71XS" AS #1
180 ON STOP GOSUB *STOP.TRAP
190 ON KEY GOSUB *F.KEY1,*F.KEY2,*F.KEY3,*F.KEY4,*F.KEY5,*F.KEY6,*F.KEY7,*RET,*R
ET,*F.KEY10
200 ON ERROR GOTO *ERROR.TRAP
210 STOP ON:KEY ON
220 PRINT "Hit [f.1] key !!!":PRINT
230 *COM.TRAP
240 BL=LOC(1):IF BL=0 THEN *SEND
250 R$=INPUT$(BL,#1)
260 CALL GRAPHIC(R$,LISTING,LY,LX,LY1,LX1)
270 GOTO *COM.TRAP
280 *SEND
290 CALL CURSOR:K$=INKEY$:IF K$="" THEN *COM.TRAP
300 IF K$=BS$ OR K$=DEL$ OR K$=GS$ THEN GOSUB *BACK.SPACE:GOTO *COM.TRAP
310 IF K$=FS$ OR K$=RS$ OR K$=US$ THEN *COM.TRAP
320 IF K$=CL$ THEN CLS:CALL HSCLS2:GOTO *COM.TRAP
330 IF K$=TAB$ THEN *SCREEN.EDIT
340 PRINT K$;:PRINT #1,K$;
350 IF LISTING THEN LPRINT K$;
360 GOTO *COM.TRAP
370 *BACK.SPACE
380 X=POS(0)-1:Y=CSRLIN:IF X<0 THEN 420
390 PRINT #1,BS$;
400 LOCATE X,Y:PRINT " ";:LOCATE X,Y
410 IF LISTING THEN LPRINT BS$;
420 RETURN
430 *RET
440 RETURN
450 *ERROR.TRAP
460 IF ERR=23 THEN PRINT "Line buffer overflow":GOTO 480
470 ON ERROR GOTO 0
480 RESUME NEXT
490 RETURN
500 *STOP.TRAP
510 FOR W=1 TO 100:OUT &H21,&H3F:NEXT W:OUT &H21,&H37
520 RETURN
530 *SCREEN.EDIT
540 CONSOLE ,0:ON STOP GOSUB *RET
550 X=POS(0):Y=CSRLIN:LOCATE 0,0
560 FOR AD=&HEF9A TO &HEFB2:POKE AD,1:NEXT AD
570 L$="":LINE INPUT L$

```

```

580 IF L$="" THEN 620
590 YY=(CSRLIN-1)*120+&HF3C8:CALL S.E(L$,YY)
600 PRINT #1,L$+CR$;
610 GOTO 560
620 CONSOLE ,1:LOCATE X,Y
630 BL=LOC(1):IF BL>0 THEN R$=INPUT$(BL,#1)
640 ON STOP GOSUB *STOP.TRAP
650 GOTO *COM.TRAP
660 *F.KEY1
670 PRINT "$$$CON,TSS,,ASC":PRINT #1,"$$$CON,TSS,,ASC"+CR$;
680 RETURN
690 *F.KEY2
700 IF LISTING=0 THEN LISTING=1:KEY 2,"LP 80 (ON)" ELSE IF LISTING=1 THEN LISTI
NG=0:LPRINT :KEY 2,"LP 80 (OFF)" ELSE BEEP
710 RETURN
720 *F.KEY3
730 IF LISTING=0 THEN LISTING=2:KEY 3,"LP 132 (ON)":LPRINT CHR$(15); ELSE IF LIS
TING=2 THEN LISTING=0:LPRINT CHR$(18);:KEY 3,"LP 132(OFF)" ELSE BEEP
740 RETURN
750 *F.KEY4
760 PRINT "LIST ";:PRINT #1,"LIST ";
770 IF LISTING THEN LPRINT "LIST ";
780 RETURN
790 *F.KEY5
800 PRINT "RUN"+CR$;:PRINT #1,"RUN"+CR$;
810 IF LISTING THEN LPRINT "RUN"+CR$;
820 RETURN
830 *F.KEY6
840 CALL TEXT.COPY
850 RETURN
860 *F.KEY7
870 CALL GRAPH.COPY
880 RETURN
890 *F.KEY10
900 STOP OFF:KEY OFF
910 ON ERROR GOTO 0
920 END

```

Add	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	Sum
E200	21	0C	E2	11	6B	86	01	BE	00	ED	B0	C9	F3	3E	5C	4F	:12
E210	ED	79	21	00	C0	11	01	C0	01	7F	3E	36	00	ED	B0	3C	:E6
E220	FE	5E	20	EB	D3	5F	FB	C9	CD	8B	42	3E	1B	CD	D4	3E	:2F
E230	3E	41	CD	D4	3E	3E	08	CD	D4	3E	06	50	21	30	FE	C5	:E9
E240	0E	5D	CD	14	87	20	07	0E	5C	CD	14	87	28	31	CD	F7	:E9
E250	86	06	C8	0E	5D	C5	CD	0C	87	CD	D4	3E	11	B0	FF	19	:9C
E260	C1	10	F2	06	C8	0E	5C	11	80	3E	19	C5	CD	0C	87	CD	:D5
E270	D4	3E	11	B0	FF	19	C1	10	F2	11	81	3E	19	18	01	23	:D3
E280	3E	0A	CD	D4	3E	C1	10	B7	3E	1B	CD	D4	3E	3E	32	CD	:24
E290	D4	3E	3E	0C	CD	D4	3E	C9	3E	1B	CD	D4	3E	3E	4B	CD	:92
E2A0	D4	3E	3E	90	CD	D4	3E	3E	01	CD	D4	3E	C9	F3	ED	79	:FF
E2B0	7E	D3	5F	FB	C9	C5	D5	E5	11	B0	FF	06	C8	CD	0C	87	:E1
E2C0	B7	20	03	19	10	F7	E1	D1	C1	C9	00	E5	D5	23	1A	13	:40
E2D0	77	23	1A	77	E1	7E	23	66	6F	11	4F	00	19	06	50	7E	:CF
E2E0	B7	28	04	FE	20	20	06	05	28	03	2B	18	F2	E1	70	C9	:A6
E2F0	00	CD	8B	42	21	C8	F3	06	18	C5	06	50	C5	7E	CD	D4	:93
<hr/>																	
Sum:	BC	66	DC	E3	BA	CB	54	34	F5	73	A5	8E	00	F1	4F	56	:1F

Add	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	Sum
E300	3E	23	C1	10	F7	11	28	00	19	3E	0D	CD	D4	3E	3E	0A	:ED
E310	CD	D4	3E	C1	10	E3	C9	00	22	BA	E5	ED	53	B8	E5	ED	:E7
E320	43	B6	E5	CD	BB	42	3A	4C	E6	32	BC	E5	AF	32	4C	E6	:CA
E330	2A	B8	E5	7E	32	B8	E5	2A	BA	E5	46	23	7E	23	66	6F	:BC
E340	C5	3A	BD	E5	B7	CA	5A	E3	FE	06	DA	AD	E3	C3	D1	E3	:44
E350	C1	10	ED	3A	BC	E5	32	4C	E6	C9	3A	B5	E5	B7	28	1D	:96
E360	AF	32	B5	E5	7E	23	FF	0C	28	05	FE	18	C2	50	E3	3E	:9C
E370	0C	DF	E5	CD	21	40	CD	6B	86	E1	C3	50	E3	7E	23	FE	:32
E380	1D	20	08	3E	01	32	BD	E5	C3	50	E3	FE	1B	20	06	32	:BF
E390	B5	E5	C3	50	E3	FE	7F	CA	50	E3	DF	F5	3A	B8	E5	B7	:6C
E3A0	20	04	F1	C3	50	E3	F1	CD	D4	3E	C3	50	E3	7E	23	CD	:3F
E3B0	13	E5	21	BE	E5	ED	5B	BD	E5	19	77	E1	3A	BD	E5	:D8	
E3C0	3C	32	BD	E5	FE	06	C2	50	E3	E5	CD	3A	E5	E1	C3	50	:CE
E3D0	E3	7E	23	CD	13	E5	FE	16	28	1F	FE	20	DA	50	E3	E5	:B4
E3E0	21	BE	E5	ED	5B	BD	E5	19	77	E1	3A	BD	E5	3C	FE	0B	:40
E3F0	CA	DE	E4	32	BD	E5	C3	50	E3	E5	3A	BD	E5	FE	0A	CA	:E9

Sum: C8 FA 57 30 F1 85 E9 C2 76 E4 A6 1A 63 8E 4D 2D :EF

Add	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	Sum
E400	1A	E4	FE	09	CA	79	E4	FE	08	CA	C6	E4	FE	07	CA	D5	:4A
E410	E4	E1	3E	06	32	BD	E5	C3	50	E3	3A	C4	E5	CB	77	28	:20
E420	1B	3A	C4	E5	32	C0	E5	3A	C5	E5	32	C1	E5	3A	C6	E5	:76
E430	32	C2	E5	3A	C7	E5	32	C3	E5	C3	EA	E4	3A	C6	E5	CB	:DA
E440	77	28	1B	3A	C4	E5	32	BF	E5	3A	C5	E5	32	C0	E5	3A	:68
E450	C6	E5	32	C1	E5	3A	C7	E5	32	C3	E5	C3	EA	E4	3A	C4	:D2
E460	E5	32	BF	E5	3A	C5	E5	32	C1	E5	3A	C6	E5	32	C2	E5	:35
E470	3A	C7	E5	32	C3	E5	C3	EA	E4	3A	C4	E5	CB	77	20	15	:AB
E480	3A	C4	E5	32	BF	E5	3A	C5	E5	32	C1	E5	3A	C6	E5	32	:8C
E490	C3	E5	C3	EA	E4	3A	C5	E5	CB	77	28	15	3A	C4	E5	32	:B1
E4A0	C0	E5	3A	C5	E5	32	C1	E5	3A	C6	E5	32	C3	E5	C3	EA	:CD
E4B0	E4	3A	C4	E5	32	C1	E5	3A	C5	E5	32	C2	E5	3A	C6	E5	:41
E4C0	32	C3	E5	C3	EA	E4	3A	C4	E5	32	C1	E5	3A	C5	E5	32	:3C
E4D0	C3	E5	C3	EA	E4	3A	C4	E5	32	C3	E5	C3	EA	E4	E5	21	:8D
E4E0	C4	E5	11	BF	E5	01	05	00	ED	B0	CD	3A	E5	ED	4B	B6	:DB
E4F0	E5	11	C9	E5	3E	04	F5	60	69	03	03	7E	23	66	6F	1A	:3A

Sum: E6 2D FE 57 46 D9 1E 50 DA 6D 3A EE 16 C4 C4 FB :FD

Add	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	Sum
E500	77	23	13	1A	77	13	F1	3D	20	EC	21	A1	E5	CD	AA	0F	:B8
E510	C3	11	E4	FE	1F	28	14	FE	0D	28	10	FE	1B	28	09	FE	:9C
E520	0C	28	08	FE	1D	28	0A	C9	32	B5	E5	AF	32	BD	E5	18	:B9
E530	05	3E	01	32	BD	E5	C1	C3	50	E3	2A	C9	E5	22	CD	E5	:7B
E540	2A	CB	E5	22	CF	E5	26	00	3A	BF	E5	6F	CB	25	CB	25	:03
E550	CB	25	CB	25	CB	14	CB	25	CB	14	3A	C1	E5	E6	1F	B5	:28
E560	6F	54	5D	21	16	03	AF	ED	52	CB	3C	CB	1D	22	C9	E5	:07
E570	16	00	3A	C2	E5	5F	CB	23	CB	23	CB	23	CB	23	CB	12	:EB
E580	CB	23	CB	12	3A	C3	E5	E6	1F	B3	5F	62	6B	29	29	19	:FC
E590	CB	3C	CB	1D	CB	3C	CB	1D	CB	3C	CB	1D	22	CB	E5	C9	:68
E5A0	00	28	4C	58	31	2C	4C	59	31	29	F4	28	4C	58	2C	4C	:60
E5B0	59	29	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:82
E5C0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
E5D0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
E5E0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
E5F0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00

Sum: B4 8E 29 F9 3B CE 37 58 EC 85 84 DC 88 70 1D 09 :EB

## (付録) プログラム解説

本プログラムのBASIC部分は、基本的には藤井氏によるものを採用しているので、BASIC部分の詳しい解説は藤井氏の記事を参考にして欲しい。

### ○ BASIC部分

- 80行 BASICの使用するメモリの上限を指定し、マシン語をロードする。Rオプションを指定することにより、マシン語プログラムの $E20CH \sim E2C9H$ を $866BH \sim 8728H$ に転送する。
- 170行 RS-232Cオープン。
- 210行 ストップキー、ファンクションキーによる割り込みを許可。なお、RS-232Cによる割り込み(COM ON)を使うと、RS-232Cの割り込みがストップキーの割り込みより優先し、ブレーク信号を送信しにくくなるので、COM ON文は使用していない。
- 260行 受信データの処理ルーチンをコール。
- 320行 **CLR** key が押されたならば、画面消去を行う。
- 330行 **TAB** key が押されたならば、スクリーンエディットルーチンへ。
- 560行 テキスト画面行継続コードを1(継続していない)にする。
- 590行 N88-BASICのLINE INPUT文は1行すべてを読み込むわけではなく、カーソル位置が空白であると、同じ行の後の文は無視されてしまうため、修正した行の内容を格納する文字変数L\$の再設定を行う。

### ○ マシン語部分

$E200H \sim E20BH$	初期設定ルーチン、Rオプションの指定によりこの部分が実行される。
$E20CH \sim E227H$	グラフィック画面消去ルーチン
$E228H \sim E2C9H$	グラフィック画面ハードコピールーチン
$E2CBH \sim E2EFH$	スクリーンエディットルーチン
$E2F1H \sim E316H$	テキスト画面ハードコピールーチン
$E318H \sim E59FH$	受信データ処理ルーチン
$E5A1H \sim E5B4H$	データエリア
$E5B5H \sim E5D0H$	ワークエリア