



|            |   |
|------------|---|
| Title      | PC-9800TSSインテリジェントターミナル用プログラム(MS-DOS版)への移植と2400bpsモデムの使用感                         |
| Author(s)  | 金丸, 邦康; 杉本, 直; 内田, 勝徳   |
| Citation   | センターレポート, 6, pp.58-70; 1985   |
| Issue Date | 1985-12   |
| URL        | <a href="http://hdl.handle.net/10069/25301">http://hdl.handle.net/10069/25301</a> |
| Right      |   |

This document is downloaded at: 2020-10-28T18:26:20Z

## 4. 技術解説

PC-9800TSSインテリジェントターミナル用プログラム  
(MS-DOS版)への移植と2400bpsモデムの使用感

工学部

金丸邦康, 杉本直, 内田勝徳

### 1. はじめに

インテルの8086 (または、8088) をCPUとする16ビット・パーソナルコンピュータ (以下、パソコンという) は、そのディスク・オペレーティング・システム (DOS) に、CP/M-86、MS-DOS、UNIXなどの複数の汎用OSが使用できる環境にあるが、米国市場におけるIBMパソコンのDOSがPC-DOS (MS-DOS) であることや、国内の9800シリーズ (NEC) のシェアの優位性から、MS-DOSの普及のリードは明白である。発売当初、CP/M-86のみであった富士通の16ビット機FM16βも、MS-DOSの採用を決定した。

このように16ビット・パソコンのDOSの業界標準のひとつであるMS-DOS上には、BASIC、C、PASCALなど種々の開発言語が供給されている。FORTRAN-77に関しては、そのサブセットであるMS-FORTRAN (MICRO SOFT 社) のほか数種のFORTRANコンパイラが動く。また、PC9800上では、スタンドアロンBASIC (N88-BASIC) とほぼコンパチブルなインタプリタとコンパイラの両形式のMS-DOS版BASICや、グラフィクスや通信機能を拡張させたPC-FORTRANが、メーカー (NEC) より供給されている。

したがって、FORTRANユーザーは、パソコンのMS-DOS上で、FORTRANプログラムの編集、ディバグまたは実行をおこない、大きなプログラムの場合は、同じDOS上で動くターミナルプログラムにより転送して大型計算機上で処理という分散化が可能となる。

とは言っても、パソコン上のFORTRANの仕様が、余りに大型機と比べ小さいと使いづらい。最低256KBのRAMを必要とするMS-FORTRANの仕様は、ソースおよびデータ容量として、それぞれ64KBまで使用でき、これは、約1,000行のFORTRANソースと、倍精度で100元の連立方程式を解ける程度の能力であることを意味している。

先に、著者らは、スタンドアロンのN88-BASICで記述した、1200bpsでファイル転送可能なターミナルプログラム (文献1) を発表した。今回これを一部変更して、MS-DOS上のコンパイラ版で使用し良好な結果を得た。また、本学情報処理センターより2400bpsでデータ送受信可能なモデムを借りだし、ファイル (データセット) の送受信試験を行ったので、その感想を報告する。

## 2. 移植の実際

今回使用するDOSはMS-DOS Ver 2. 11であり、次のプログラム開発言語及びユーティリティプログラムを使用し移植を行い、インタプリタとコンパイラの両方でファイルの送受信を試験した。その結果、データ転送速度2400 bpsの場合、インタプリタ上では、TSSプログラムはファイルの送受信時に画面が乱れたり、データの送り落としが生じる場合があったが、コンパイルされたTSSプログラムでは、このような転送上のトラブルを克服できることがわかった。

|               |                         |
|---------------|-------------------------|
| N88BASIC. EXE | ベーシックインタプリタ             |
| N88BASIC. LIB | ベーシックライブラリ              |
| BASICC . COM  | ベーシックコンパイラ              |
| FILECONV. EXE | MS-DOS-N88 相互のファイルコンバータ |
| SPEED . COM   | RS-232Cボーレート設定コマンド      |

N88-BASIC (86) からMS-DOSの移植に先立ち、プログラムソースの再入力为了避免のために、MS-DOSとN88-BASICとのファイルコンバータを用いた。すなわち、文献(1)の" TSS-98 "ターミナル・プログラムをアスキーセーブし、MS-DOSを立ち上げ、FILECONV. EXEを実行しファイルを転送する。MS-DOSファイルの拡張子は、. BASにする。

### 2. 1 プログラムリスト

表1に、スタンドアロンのN88-BASIC (86) 版からN88-日本語BASIC (86) (MS-DOS版) へ移植されたインテリジェントTSSターミナル・プログラムプログラムリストの主要部を示す。網掛けした部分が文献(1)に掲載のインテリジェントTSSターミナル・プログラムの変更点である。また行番号1660-1980及び2230-2437行は、スタンドアロン版と同じであるので省略している。以下修正された行をあげて、逐次簡単に説明する。

### 2. 2 ファイルの設定 (関連行 1028)

このプログラムでは、同時に3つのファイルを使用しているので、インタプリタの場合は起動時にパラメータの指定が、コンパイラの場合は注釈行が必要である。

即ち、インタプリタの場合、MS-DOSコマンドモードより

A>N88BASIC/F:3

で起動する。コンパイラの場合、注釈行

REM \$FILE, 3

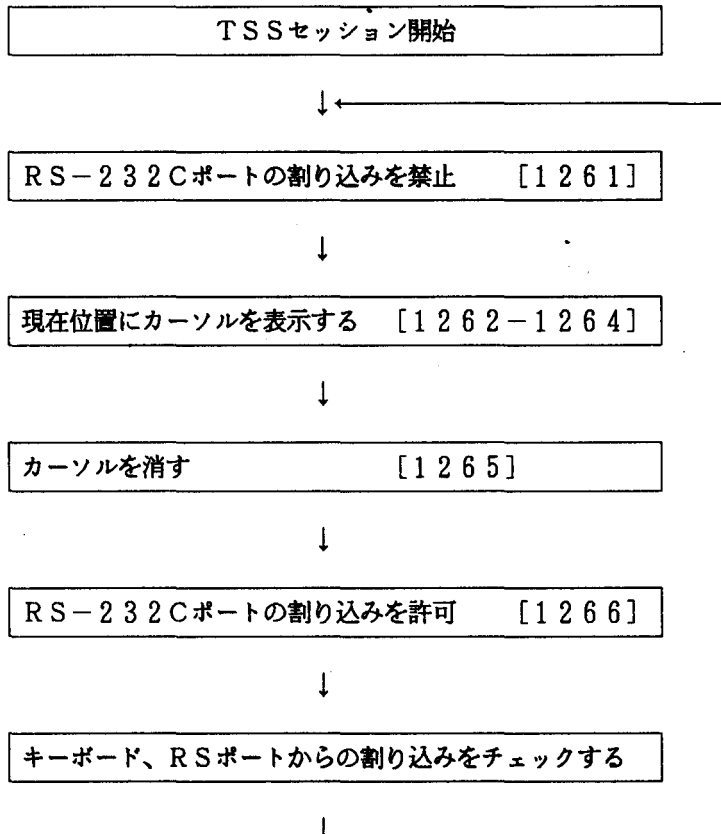
をプログラムの先頭に入れる。

### 2.3 カーソルの表示 (関連行 1261~1266)

文献(1)に掲載のN88-BASIC(86)プログラムでのBIOSコールによるカーソルの表示はそのままでもN88-日本語BASIC(86)(MS-DOS版)のインタプリタまたはコンパイラでエラーの発生はない。また、通常のカーソル表示の回復には問題なく使用できるが、通信プログラムのような高速度処理を必要とする場合には問題があり、2400bpsでの受信では文字落としが発生する。機械語ルーチンであるにもかかわらず、このようなトラブルがおこるのは一見奇異な気がするが、この部分のソフトウェア割り込みが、MS-DOS上に仮想的につくられたBASIC環境上で動いているので、カーソル回復処理にかなりの時間を消費しているものと思われる。

そこで、カーソルの現在位置をさがしだし、テキスト画面のアトリビュートに属性を直接書き込むように変更する。

以下に簡単なフローチャートを示す。 [ ] 内の数字は関連する行番号である。



VRAMのテキストと文字属性（アトリビュート）のベースアドレスは下表のとおりであり、CRTの左上角を原点（0，0）として、オフセットアドレスは行位置（0～24）×160＋水平位置（0～79）×2で計算される。例えば、本プログラムで用いたカーソルは、白のアンダーラインであり、（11101001）b =（E9）hを、文字属性として直接VRAMに書き込んでいる。詳細は文献（2）を参照されたい。

#### VRAMのアドレス（ベース値）

| ページ | テキスト   | アトリビュート |
|-----|--------|---------|
| 1   | &hA000 | &hA200  |
| 2   | &hA100 | &hA300  |

#### 文字属性

| 7 | 6 | 5 | 4   | 3                 | 2              | 1              | 0                |
|---|---|---|-----|-------------------|----------------|----------------|------------------|
| 緑 | 赤 | 青 | グラフ | アンダ<br>ーライ<br>ン=1 | リバー<br>ス<br>=1 | プリン<br>ク<br>=1 | シーク<br>レット<br>=0 |

#### 2.4 ガベージコレクション対策（関連行 1281）

ガベージコレクション対策としてFRE関数を用いているが、N88-BASIC（86）とN88-日本語BASIC（86）（MS-DOS版）ではFRE関数の引き数の機能が異なる。コンパイラの場合、FRE（2）ではエラーが発生するので、FRE（0）に変更する。

#### 2.5 ファイルディスクリプタの変更（関連行 1440, 1550, 1610, 1635, 2191-2202）

スタンドアロンのN88-BASIC（86）のファイル・ディスクリプタのドライブ指定は、数字1, 2, …であるが、N88-日本語BASIC（86）（MS-DOS版）では、アルファベット A, B, …である。

なお、ファイル送受信の前にディスクのディレクトリを画面に表示するが、フロッピーディスクを挿入していない場合は、エラー処理ルーチンによりメッセージを表示して、復帰するように変更した。またMS-DOSコマンドモードに同様のファイル指定に関するワイルドカードをサポートし、またテンプレート機能により、既入力ファイル指定と異なった部分のみを入力すればよい。ファイル指定の例を以下に示す。[ ]内の文字は、すでに設定してあるフ

ファイル・ディスクリプタでありドライブ名、ファイル名、拡張子のいずれか変更の必要がなければ、空白により再入力省略できる。

|                   |                        |
|-------------------|------------------------|
| [*.*]: A:.FOR     | ドライブ名: Aで拡張子がFORを表示    |
| [A:*.FOR]: TSS*   | 上記に加えファイル名がTSS*を表示     |
| [A:TSS*.FOR]: *.* | ドライブ名、ファイル名の指定を取り消して表示 |

## 2. 6 ファンクションキーの再設定 (関連行 2501~2510)

N88-BASIC (86) とN88-日本語BASIC (86) (MS-DOS版) のワークエリアは異なる。従ってワークエリアからファンクションキーの情報を読み取り直接メモリーに書き込む方法は、BASICモードからMS-DOSモードに戻る時に暴走する。従ってKEY文によりファンクションキーの内容を再設定するように変更する。

## 2. 7 バグの訂正 (関連行 2000)

文献(1)に掲載のインテリジェントTSSターミナル・プログラムでバグが発見されたので以下のように訂正する。

```
IF EOF(3) THEN RETURN
```

↓

```
IF EOF(3) THEN FOR I = 1 TO 20 : NEXT I : GOTO *JNKBY
```

1250行でエラーラップされた場合、ラベル\*RSERRにジャンプし更に\*RSOUTにジャンプすると、そこでエラーコードが3になり再び\*RSERRにジャンプし無限ループに陥る。そこでRETURNがエラーの原因にならないようにGOTOに変更する。また、ファイル送信時にハンドシェイクのタイミングがあわず画面が乱れるのを防止するためFORループをいれる。

## 2. 8 ポーレートの設定

SYSTEMコマンドによりN88-日本語BASIC (86) (MS-DOS版) からMS-DOSのコマンドモードに戻し、SPEED.COMを実行し以下に示すようにパラメータを設定する。

A>speed

SPBED Version 2.0

RS232C-0 2400 BITS -7 PARITY-EVEN STOP-1 NONE

### 3. モデム使用の感想

今回、本学情報処理センターよりモデムを借りだし、データ転送速度 2400 bpsで、ファイル（データセット）の送受信に使用したので、その使用感を述べる。

モデムは、コンピュータ側が扱うデジタル信号を電話回線が扱うアナログ信号に変換するという基本的な機能は音響カブラと同じであるが、音響カブラが受話器からの音をデジタル信号に変換してコンピュータに送るのに対して、モデムでは、パソコンに電話回線を直結することができる。すなわち、電話回線のアナログ信号とコンピュータが扱うデジタル信号とを音を介在させることなく直接変換するので、外部の騒音や振動などによる誤動作の確率は低減し、音響カブラよりも高い転送速度が可能になるという特徴をもっている。

ここで、使用した機種構成は、つぎのとおりである。

|              |          |                                |
|--------------|----------|--------------------------------|
| モデム          | : FACOM  | 1935HA 全二重<br>(MODEM 2400)     |
| パソコン         | : NEC製   | PC-8801, PC-9801<br>PC-9801 M2 |
| ディスク・ドライブ    | : NEC製   | PC-80S31, PC-8881              |
| ディスプレイ (CRT) | : NEC製   | PC-8050n, PC-8853              |
| プリンター        | : EPSON製 | MP-100 III                     |

なお、ターミナル用ソフトウェアは、文献(1)、(3)および本報MS-DOS版(コンパイラによる)を用いた。

これまでに、著者らは、音響カブラ(エプソン製 CP-20 データ転送速度 300 bps, RACAL製 V13412J 1200 bps)の使用経験があるが、セッション開設時、音響カブラへの受話器の密着などの煩わしさがあった。しかしモデムでは、発信音の後、データボタンを押下するだけでよく、その操作性は、はるかに音響カブラを凌いでいる。また、RACAL製音響カブラ(1200 bps)でしばしば起こっていた開設時の不通状態もモデムでは一切なく、その信頼性は、数段に向上している。

今回使用した転送速度2400 bps(FACOM1935HA)の場合、300 bps(エプソン製CP-20)に見られるような一行の表示が、数文字ずつかたまってもバラバラと表示されるようなこ

とはなく、あたかも一行同時に出画するように思われるほど高速である。その高速度受信によるパソコン側での読み落としは、受信データをCRTに表示するのみの場合、PC9801及びPC8801のスタンドアロン版(BASICインタプリタ)およびMS-DOSコンパイラ版では、現在までは発生していない。

ところが、データを受信しながら、続いてディスクに書き込む場合、CRTのスクリーン上では、改行コード以前のデータの途中で改行するなど、スクリーンが乱れることがある。これは、ディスク上の空領域の物理的配置にも依存するのであろうが、ディスク・ドライブのヘッドがシークしデータを書き込む間、パソコンのCPUはデータ表示に関する仕事をやっていないようなのでこのような現象が発生すると考えられる。ただし、これはスクリーン上だけの事であって、実際にディスクに書き込まれたデータには何の欠落も無く問題はないことが確認された。この点に関しては、文献(4)などのRAMディスクの援用により解決できることも確認した。

一方、MS-DOS上のBASICインタプリタ版のTSSプログラムの場合、以上の問題に加えて、ファイル送信時にも問題があることが判明した。データの送信に先立ち、ホストからの行番号を受信するプロトコルを採用しているが、これがMS-DOSのインタプリタ版ではうまくいかない。より高速なコンパイラ版では、送信時に問題になるようなことは起きていない。

また使用したプリンターの仕様は、印字速度 100 CPS (普通文字)、改行時間 約200 ms/行であるが、このプリンターを、データ受信しながら使用した場合、約4~5行を印字した後データを読み落としたり、最悪の場合はハングアップして、パソコン側で line buffer overflow を起こし、セッションを中断してしまうこともあった。これを避けるためには、on line で使用するさいは、プリンタにバッファを持たせることが一案として考えられる。

以上が2400bpsモデムの使用感であるが、特にプリンターの同時使用を考えなければ、セッション開設時の操作性の向上や、送受信の高速化、文字化けなどの誤動作の消滅などがみられ、パソコンを用いたTSSインテリジェントターミナルの利用者にとって、非常に有益なものであると思われる。今後は、モデムの低価格化が切望される。

最後に、本学情報処理センターの今回のシステム拡張に伴い、カナコードのサポートに関して変更があったので記しておく。

READYモードで、

SETCODE I(KANA) D(JIS)

を入力の後、作成するデータセット名が ABC. FORTの場合、

EDIT ABC FORT ASIS

と入力すると、これ以降カナがサポートされる。



#### 4) おわりに

本報告は、とりあえず、パソコン上でFORTRANプログラムを開発するが、先ざきでは、プログラムの配列変数などの仕様拡張をしたいという希望をもつユーザーのために役立つ目的で発表したものである。本プログラムを利用されたいとお考えの方は、著者までご一報ください、プログラムソースに限り、フロッピーディスク（8インチ、5インチ2HDまたは2DDに限る）で提供が可能である。

#### 参考文献

- (1) 金丸, 「PC-9800 TSSインテリジェントターミナル用プログラム」,  
センターレポート, 第4号 (1983).
- (2) 藤田, 幸田, 「PC-Techknow9800」, システムソフト.
- (3) 清木, 芳本, 「PC8800/PC8000シリーズによるテキスト編集および,  
TSSインテリジェントターミナルのためのプログラム」,  
センターレポート, 第3号 (1982).
- (4) 松沼, 「RAM-DISK」, アスキー, .8-12 (1984).

表1 PC-9800用TSSプログラム (MS-DOS版)

```

1000 'FILE NAME : "TSS-98MS"
1010 ' TSS Intelligent Terminal by PC-9801 ( TSS-98 )
1020 '      Ver. 1.0 ( '82.8.30 ) [ Programmed by Y.SEIKI & T.YOSHIMOTO for
PC8801 & PC8001]
1025 '      Ver. 2.0 ( '83. 8. 1 ) [ Rewritten by K.KANEMARU for PC-9801]
1026 '      Ver. 2.1 ( '83.10.26 ) [ CSR BLINK & MOVE is supported for 1200
bps]
1027 '      Ver. 3.0 ( '85. . . ) [ MS-DOS VERSION ]
1028 REM $FILE.3
      (同時にオープンするファイルの数を設定する。)
1029 DEFINT A-Z
1030 CLEAR 1000:WIDTH 80,20:CONSOLE 0,20,1,1:M$=""

```

\*\*\* 1032-1038 Delete \*\*\*

(BIOSコールによるカーソル点滅は行わない。)

```

1040 OPEN "COM:E72XS" AS #3 : FM$="Fnnnn/PASSWORD" :B=1 ' (B=1,E72XS --> 300 or
1200 bps. B=0,E71XS --> 300 bps)
1050 COLOR 7:PRINT CHR$(12):"**** PC-9801 :TSS Intelligent Terminal for FACOM
M-360 ****":PRINT TAB( 5):"TEL. (****) ---> 1200 bps (****-****) ---> 2400
bps":PRINT
1060 COLOR 2:PRINT " BREAK :G-x ":COLOR 7:PRINT " ==> ":COLOR 3:PRINT "TSS Se
ssion Auto Start ":COLOR 7:PRINT " ( LOGON TSS Fnnnn/PASSWORD )"
1070 COLOR 6:PRINT " Return key ==> ":COLOR 5:PRINT "TSS Session Manual Start
":COLOR 7:PRINT "( BREAK---LOGON TSS Fnnnn )"
1080 '
1090 'AUTO-TSS :BREAK ==> TSS Session Auto Start
1100 B$=INKEY$:DEF SEG=&H4000:IF B$=CHR$(129) THEN POKE &H1E0,ASC("&"):OUT &H32.
&H3F:BEEP:POKE &H1E0,0:FOR T=0 TO 500:NEXT:OUT &H32.&H37
1110 IF B$=CHR$(13) THEN GOSUB*SBEEP:M$="PLEASE LOGON TSS!":GOTO 1150
1120 IF LOC(3) THEN SS=LOC(3):A$=INPUT$(SS,#3):LF=INSTR(A$,CHR$(10))
1130 IF LF THEN COLOR 7:PRINT:PRINT "START:":COLOR 5:PRINT " LOGON TSS ":
BEEP:PRINT #3."LOGON TSS "+FM$ ELSE 1100

```

\*\*\* 1140 Delete \*\*\*

(ドライブ名の指定は [2191-2202] で行う。)

```

1150 COLOR 6: PRINT CHR$(12)::COLOR 2:PRINT " BREAK:G-x ":
1160 COLOR 7: PRINT "PRN: ":COLOR 5:PRINT "f*1 ":COLOR 7:PRINT "0 ":
1170 COLOR 6: PRINT "OPEN-FR: f*2/":COLOR 5:PRINT " FR ":COLOR 7:PRINT "#0/0
":

```

```

1180 COLOR 4: PRINT "OPEN-FS: f*3/":COLOR 5:PRINT " FS ":COLOR 7:PRINT "#0/0
":
1190 COLOR 1: PRINT "END:G-¥":LOCATE 65,18:COLOR 7:PRINT "L-EDT: ^":LOCATE 0,1
1200 CONSOLE 2,16,1,1:GOSUB *FKKEY:COLOR 7:PRINT:PRINT M$."(":"TIMES:")"
1210 '
1211 KEY (1) ON:KEY (2) ON:KEY (3) ON:ON KEY GOSUB *FK1,*FK2,*FK3
1212 '
1213 ON COM GOSUB *RSOUT
      (COM ON を消す。)
1214 '
1220 LP=0:F1=0:F2=0:FR=0:FS=0:CL=7 : DV$ = "" : FANDE$ = "*,*"
      (ドライブ名、ファイル名、拡張子の初期設定。)
1230 '
1240 *INKEY ' MAIN ROUTINE
1250 ON ERROR GOTO *RSERR
1260 '
1261 COM STOP
1262 DEF SEG = &HA200
1263 ADR = CSRLIN*&HA0+POS(0)*&H2
1264 POKE ADR,&HE9 : FOR I = 1 TO 20 : NEXT I
1265 POKE ADR,&HE1
1266 COM ON
      (RSポートの割り込みを禁止し、現在位置にカーソルを1回点滅し、
      RSポートの割り込みを許可する。)
1270 IF FS THEN GOSUB *FSEND
1280 COLOR 5:B$=INKEY$
      (CSBL%(5)...USR0(0)を消す。)
1281 T11=FRE(0)
      (引き数を0にする。強制的にガベージコレクションを行う。)
1285 IF B$ = "" THEN GOTO *INKEY
1290 IF B$>="a" AND B$<="z" THEN B$=CHR$(ASC(B$)-32):GOTO 1330 'small chr ==>
CAP. CHR
1300 IF B$=CHR$(8) THEN PRINT #3,CHR$(8):: PRINT CHR$(29):" ":CHR$(29)::GOTO 128
0 '[BS] KEY
1305 IF B$=CHR$(94) THEN B$="":GOSUB *LEDT ' ^ :L-EDT
1310 IF B$=CHR$(129) THEN B$="":GOSUB *BREAK 'G-x
1320 IF B$=CHR$(241) THEN *ENDS 'G-¥ :end of TSS
1325 IF B$=>CHR$(28) AND B$=<CHR$(31) OR B$=CHR$(127) THEN B$="":GOTO *INKEY'<-,
->,UP/DOWN,[DEL] keys are not available.
1326 IF B$ = CHR$(&HD) THEN PRINT #3,CHR$(&HD) : PRINT CHR$(&HD) : GOTO *INKEY
'RET
      (キャリッジ・リターンをサポートするようにする。)
1330 IF B$<>" " THEN PRINT B$::PRINT #3,B$:

```

```

1340 GOTO #INKEY
1350 '
1360 #BREAK
1370 COM STOP:DEF SEG = &HA000 : POKE &HFO0,ASC("**"):BEEP:CLOSE #1,2:OUT &H32,&H
3F: FR$="":FS$="":F1=0:F2=0:FR=0:FS=0:CL=7
1380 GOSUB 2270:GOSUB 2330:KEY 2."OPN-FR":KEY 3."OPN-FS":X=POS(0):Y=CSRLIN
1390 LOCATE 0,1:PRINT SPACE$(80):LOCATE 8,18:PRINT SPACE$(20)::LOCATE X,Y:OUT &
H32,&H37:POKE &HFO0,0:COM ON:RETURN
1400 '
1410 #OPENFR '(OPEN FR$ FOR OUTPUT AS #1)
1420 IF FS THEN RETURN
1430 ON ERROR GOTO 1500:X=POS(0):Y=CSRLIN:LOCATE 12,18:COLOR 7:PRINT "**":LOCATE
X,Y
1440 GOSUB #DIR
      (ディレクトリを表示。)
1450 COLOR 6:FR$="":PRINT "*** File Name to Receive *** ":COLOR 7:INPUT FR$:X$=
FR$:GOSUB#FLCHK:FR$=X$: IF BF THEN GOSUB#SBEEP:COLOR 2:PRINT "Illegal File Name!"
":GOTO 1450
1460 IF FR$="" THEN GOSUB#SBEEP:COLOR 2:PRINT "*** Cancelled ! ***":X=POS(0):Y=C
SRLIN:LOCATE 12,18:PRINT " ":LOCATE X,Y:GOTO 1510
1470 IF FR$=FS$ THEN GOSUB#SBEEP:COLOR 3:PRINT "*** File Name ":COLOR 4:PRINT F
R$:COLOR 3:PRINT " already Used! ***":GOTO 1450
1480 OPEN FR$ FOR OUTPUT AS #1:F1=1:CL=6:GOSUB 2270
1490 X=POS(0):Y=CSRLIN:LOCATE 25,1:COLOR 7:PRINT "("FR$)":KEY 2."FR-off":
LOCATE 12,18:PRINT " ":LOCATE X,Y:GOTO 1510
1500 IF ERL=1480 THEN GOSUB#SBEEP:COLOR 2:PRINT "Abnormal File Name! ERR NO.=":
:COLOR 6:PRINT ERR:RESUME 1450
1510 ON ERROR GOTO 0:RETURN
1520 '
1530 #OPENFS '(OPEN FS$ FOR INPUT AS #2)
1540 ON ERROR GOTO 1630:X=POS(0):Y=CSRLIN:LOCATE 19,18:COLOR 7:PRINT "**":LOCATE
X,Y
1550 GOSUB #DIR
      (ディレクトリを表示。)
1560 COLOR 4:FS$="":PRINT "*** File Name to Send *** ":COLOR 7:INPUT FS$:X$=FS$
:GOSUB#FLCHK:FS$=X$: IF BF THEN GOSUB#SBEEP:COLOR 2:PRINT "Illegal File Name!":G
OTO 1560
1570 IF FS$="" THEN GOSUB#SBEEP:COLOR 2:PRINT "*** Cancelled ! ***":X=POS(0):Y=C
SRLIN:LOCATE 19,18:PRINT " ":LOCATE X,Y:GOTO 1640
1580 IF FS$=FR$ THEN GOSUB#SBEEP:COLOR 3:PRINT "*** File Name ":COLOR 6:PRINT F
R$:COLOR 3:PRINT " already Used! ***":GOTO 1560
1590 P=INSTR(FS$,"")
1600 Q=INSTR(FS$,"#")

```

```

1610 ON ERROR GOTO 1630 : OPEN FS$ FOR INPUT AS #2:F2=2:CL=4:GOSUB 2330
      (エラートラップのジャンプ先を設定する。)
1620 X=POS(0):Y=CSRLIN:LOCATE 48,1:COLOR 7:PRINT "("FS$:")":KEY 3,"FS-off":
LOCATE 19,18:PRINT " "":LOCATE X,Y:GOTO 1640
1630 IF ERR = 53 THEN GOSUB*SBEEP:COLOR 2:PRINT "File not found ":RESUME 1560
      (ERL = 1610 を ERR = 53 に変更する。)
1640 ON ERROR GOTO 0:RETURN
1650 '

```

\*\*\* 1660-1980 ( T S S - 9 8 の対応する行に同じ) \*\*\*

```

1990 *RSOUT
2000 IF EOF(3) THEN FOR I = 1 TO 20 : NEXT I : GOTO *INKEY
      (ハンドシェイクのタイミングをとる。バグを訂正する。)
2010 SS=LOC(3)
2013 A$=INPUT$(SS,#3):IF FS THEN GOSUB *INSKF
2020 COLOR 7:PRINT A$:
2030 IF LP THEN LPRINT A$:
2035 NN=INSTR(A$,CHR$(0)):IF NN THEN A$=A$+" "
2040 IF FR THEN PRINT #1,A$:
2050 GOTO 2000
2060 '
2070 *RSERR
2080 IF ERR=23 THEN OUT &H32,&H3F:GOSUB*BREAK:RESUME*INKEY
2090 RESUME *RSOUT
2100 '
2110 F$="":PRINT #3,RT$:TIME$
2120 *SBEEP
2130 BEEP 1:FOR T=0 TO 50:NEXT:BEEP 0:FOR T=0 TO 30:NEXT:RETURN
2140 '
2150 *TIME
2160 RT=VAL(MID$(RT$,7,2)):ST$=TIME$:ST=VAL(MID$(ST$,7,2)):TIME=ST-RT
2170 IF TIME<0 THEN TIME=TIME+60
2180 RETURN
2190 '
2191 *DIR
2192 ON ERROR GOTO 2202
2193 PRINT "Input " : COLOR 2 : PRINT "directory " : COLOR 7 : PRINT "file des
cripter ["+DV$+FANDE1$"] : " :
2194 INPUT "",FANDE1$ : IF FANDE1$ = "" THEN GOTO 2198
2195 SEMIC = INSTR(FANDE1$,".") : IF SEMIC = 0 THEN GOTO 2198 ELSE DV$ = LEFT$(F
ANDE1$,SEMIC) : FANDE1$ = RIGHT$(FANDE1$,LEN(FANDE1$)-SEMIC) : IF DV$ = "*" THE
N DV$ = ""

```

```

2196 IF RIGHT$(FANDE1$.1) = "." THEN FANDE1$ = FANDE1$+MID$(FANDE$.INSTR(FANDE$.
".")+1.LEN(FANDE1$)) : GOTO 2198
2197 IF LEFT$(FANDE1$.1) = "." THEN FANDE1$ = MID$(FANDE$.1.INSTR(FANDE$.".")-1)
+FANDE1$ : GOTO 2197
2198 IF FANDE1$ <> "" THEN FANDE$ = FANDE1$
2199 IF DV$ = "" OR DV$ = "a:" OR DV$ = "A:" THEN PRINT:COLOR 7:PRINT "DRIVE-A":
COLOR 5:FILES "A:"+FANDE$
2200 IF DV$ = "" OR DV$ = "b:" OR DV$ = "B:" THEN PRINT:COLOR 7:PRINT "DRIVE-B":
COLOR 5:FILES "B:"+FANDE$
2201 RETURN
2202 IF ERR = 62 THEN COLOR 5 : PRINT "Disk offline" : RESUME NEXT
(ディスクットのディレクトリを表示する。)
2203
2205 *FK1
2210 IF LP=0 THEN LP=1:KEY 1,"PR- on" ELSE LP=0:KEY 1,"PR-off"
2220 COM STOP:COLOR 7:POKE &HB4A,ASC("&")*LP:X=POS(0):Y=CSRLIN:LOCATE 20,0:PRINT
LP:LOCATE X,Y:COM ON:RETURN

```

\*\*\* 2230-2437 (TSS-98の対応する行に同じ)\*\*\*

```

2440 *ENDS
2450 COLOR 7 : WIDTH 80,20:ON ERROR GOTO 0
2460 PRINT CHR$(12)"TSS SESSION ENDED!".( ":TIME$:" )
2490 CLOSE:COM OFF:KEY OFF
2500 GOSUB*SBEEP:GOSUB*SBEEP:CONSOLE 0,20,1,1
2501 KEY 1, "load "+CHR$(&H22)
2502 KEY 2, "auto "
2503 KEY 3, "go to "
2504 KEY 4, "list "
2505 KEY 5, "run"+CHR$(&HD)
2506 KEY 6, "save "+CHR$(&H22)
2507 KEY 7, "key "
2508 KEY 8, "print "
2509 KEY 9, "edit ."
2510 KEY 10, "cont"+CHR$(&HD)
(ファンクションキーの内容の再設定)
2520 END 'end of TSS-98 '83.10.26 VER. 2.1

```