OUKA
The University of Osaka
Institutional Knowledge Archive

Title	9600BPSでも使えるFM-8を用いたインテリジェント・ターミナル
Author(s)	嘉納, 秀明
Citation	大阪大学大型計算機センターニュース. 1983, 48, p. 49-61
Version Type	VoR
URL	https://hdl.handle.net/11094/65557
rights	
Note	

The University of Osaka Institutional Knowledge Archive : OUKA

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

The University of Osaka

9600ＢＰＳでも使えるＦＭ－８を用いたインテリジェント・ターミナル

大阪大学工学部
嘉納秀明

1. はじめに

筆者の属する研究室ではＦＭ－８を300ＢＰＳ交換回線ＴＳＳ端末として使用しているが、最近ミニフロッピーディスクが安くなったので、これも購入した。こうなるとデータをディスクに出し入れ出来るようなインテリジェントターミナル化が望ましくなる。折しも本誌Vol.12, No.2 に藤井博氏がＰＣ８８０１のインテリジェントターミナルの記事を書かれたので、早速これをＦＭ－８用に改造してみた。ところがどうもうまく働かない。リストを出力させると10行位でLine buffer overflowを起す。プリンタを使用したり、ディスクを動かすとさらに悪くなる。これではどうにも使えないので、すこしＲＳ２３２Ｃ受信ルーチンを調べてみると、バッファは80_H(注1)しかない上に、ルーチン自体が冗長なことがわかった。またディスクはデータが256バイトたまると書き込みを始めるが、このときＣＰＵはこれにかかりっきりになるので、受信しながらディスクにデータを落すことは不可能であることもわかった。そこでＲＳ２３２Ｃに信号が来るとＩＲＱ 割込みがかかることを利用して、割込ベクトルを書きかえ自作の機械語プログラムに飛ばすこととした。またディスク出力のためにできるだけバッファを大きくとって、一つの通信が終ったところでデータを取り込むことにした。こうして受信部のプログラムを作ると実によく働くようになった。研究室には1200ＢＰＳや9600ＢＰＳの端末もあるので、まず1200 ＢＰＳにつけてみるとこれでもうまくゆく。欲を出して9600ＢＰＳにつけると、プリンタだけならうまくゆくが、ＣＲＴ出力をするとデータ抜けが起る。プリンタ出力とＣＲＴ出力にはＦ－Basic インタープリンター内の機械語サブルーチンを利用したが、9600ＢＰＳに対しては、ＣＲＴ出力ルーチンの割込禁止時間が長すぎたわけである。このためにさらに改良して、ＢＩＯＳルーチンを用いず、直接サブＣＰＵにデータを送るプログラムに改めた。(注2)すると9600ＢＰＳでも難なく通信できるようになった。

今までのインテリジェント端末についての記事はＰＣ系が多いのと、1200ＢＰＳまでのものなので、本プログラムは少なからずおられるＦＭ－８利用者の参考になると思い報告する次第である。

（注１）　80_Hの添字Ｈは１６進表示を意味する。

（注２）　ＦＭ８はメインＣＰＵとサブＣＰＵとからなり、入出力関係はほとんどサブＣＰＵで処理している。メインＣＰＵではＢＩＯＳというルーチンが一括してサブＣＰＵとの通信に当っているが、汎用プログラムなので、特殊な用途には冗長である。

2. Basic プログラムの説明

本プログラムは、Vol.12, No.2の藤井氏のプログラムを下敷にしたものであるが、FM-8用に改造し、何点か独自の部分をつけ加えた。

まず通常の使用法としては、Basicコマンドの〝TERM〟と同じにしてある。すなわちファンクションキーPF6を押すとプリンタ作動、もう一度押すとプリンタ停止であり、PF10はブレイクキーである。他のインテリジェント機能もすべてファンクションキーを押すことによって行う。

PF1 …… ログオン

PF2 …… ID番号送信

PF3 …… RUN ㊐. このキーはもともとのキーの定義のまゝであるが、TSS用にも用いることができる。

PF4 …… EDIT. スクリーンエディットを行う。

PF5 …… STOP. ホストと切りはなしたあと、本プログラム自身の後始末をする。

PF6 …… プリンタのオンオフ。

PF7 …… ディスクにTSSデータを書き込む。最初の一押でホストからリストが送られて来、次の一押でバッファにたまったデータをディスクに書き込む。

PF8 …… ディスクにあるデータをホストに送信。

PF9 …… プリンタを一行80字にするか132字にするかの選択キー。

PF10 …… ブレイクキー

このような各キーの役割を覚えておくのは大変面倒なので、スクリーンの下2行に常時ファンクションキーの内容が表示されているようにしてある。従って利用者はここを見て、必要なキーを押せばよい。

なお、300BPSと1200BPSの選択はプログラムで出来るが、9600BPSを利用するときは、FM-8計算機本体うしろのディップスイッチをあらかじめ切り換えなければならない。300と1200に対してはディップスイッチ1，6がONであり、9600に対してはディップスイッチ4，6がONであるので使用前に切り換えておいて下さい。

以下にプログラムを順に説明する。

```
10 'Terminal program for FM-8
20 'Initial Set
30 WIDTH 80,25:CLEAR &H200,&H2500 :GOSUB 1320 :CONSOLE 0,23,1,1
40 CR$=CHR$(&H0D):LOING=0:IND=0:LISING=0:LNG=0:DIM A%(40):DIM C%(80)
50 'RS232C open
60 INPUT "ZEN20(1) OR HAN20(0)";II
70 INPUT "BAUT RATE";I
80 IF I=300 THEN 100 ELSE 90
90  OPEN "O", 1, "COM0:(F7E1)": OPEN "I", 2, "COM0:(F7E1)":GO TO 110
100 OPEN "O", 1, "COM0:(S7E1)": OPEN "I", 2, "COM0:(S7E1)"
110 FOR I=&H6F00 TO &H6F91:READ D$:DA=VAL("&H"+D$):POKE I,DA:NEXT
120 POKE &HFFF8,&H6F:POKE &HFFF9,&H00
130 ON COM(0) GOSUB 380
140 KEY(1) ON:KEY(2) ON:KEY(4) ON:KEY(5) ON
150 KEY(6) ON:KEY(7) ON:KEY(8) ON:KEY(9) ON:KEY(10) ON
160 ON ERROR  GOTO   590
170 ON KEY(1) GOSUB  710 :'LOGON
180 ON KEY(2) GOSUB  850 :'USER ID
190 ON KEY(4) GOSUB  450 :'EDIT
200 ON KEY(5) GOSUB  750 :'STOP
210 ON KEY(6) GOSUB 1280 :'PRINTER
220 ON KEY(7) GOSUB  920 :'LIST LOGING
230 ON KEY(8) GOSUB 1110 :'FILE READ
240 ON KEY(9) GOSUB 1380 :'80 OR 132
250 ON KEY(10)GOSUB  890 :'BREAK
260 PRINT "LOG-ON PLEASE"
270 '
```

行番号	説　　明
30	80文字、25行モードにし、2500_H 以下をバッファ用に確保し、ファンクションキーをセットする。
40	プリンター、ディスクなどのフラグを0にセット。
60	全二重か半二重かの選択、センター用には0を入力。
70	BPS入力。
80〜100	300BPSに対してはSLOWモード、それ以外にはFASTモードで、RS232Cポートをopen。
110	受信用機械語ルーチンを読込む。
120	IRQ割込ベクトルを書きかえる。
140〜250	ファンクションキーによる割込の行先の定義と、割込受付開始。
260	準備完了、ログオンをどうぞ。

```
280 'Main program
290 COM(0) ON
300 K$=INKEY$:IF K$="" THEN 380
310 'Send routine
320 IF K$=CHR$(&H1D) OR K$=CHR$(&H8) THEN  GOSUB 410:  GOTO 300
330 IF K$=CHR$(&H1C) OR K$=CHR$(&H1E) OR K$=CHR$(&H1F) THEN 300
340 PRINT#1,K$;:IF II=0 THEN PRINT K$;
350 IF LISING THEN PRINT#3, K$;
360 '
370 'Receive routine
380 EXEC &H6F20:ON IND GOTO 980,1150,1080,560
390 GO TO  300
400 '
410 'Back space routine
420 PRINT#1, CHR$(&H8);:LOCATE POS(0)-1,CSRLIN
430 PRINT " ";:LOCATE POS(0)-1,CSRLIN:RETURN
440 '
```

行番号	説　　明
290	通信開始。
300	キーボード入力。
320 330	←キーまたはBSキーならバックスペース処理、→, ↑, ↓ キーを無効にする。
340	キーボード入力の画面表示、全二重のときはエコーバックがあるので表示せず、
350	プリントオンならプリント出力。
380	受信機械語ルーチン、よそから呼ばれたときはINDの値によって帰る先が異る。通常は300へもどる。
420 430	バックスペース処理。

```
440 '
450 'Edit routine
460 INPUT "How many lines";E:IF E=0 THEN RETURN
470 IF E>23 THEN 460
480 PRINT "*";:Y=CSRLIN:LOCATE 1,Y:Y=Y-E-4
490 LINE INPUT W$:IF INSTR(W$,"¥")=0 THEN 490
500 FOR J=1 TO E: GET@ (0,Y)-(79,Y),A%
510 FOR K=0 TO 39:D=VARPTR(A%(K)):C%(K*2)=PEEK(D):C%(K*2+1)=PEEK(D+1):NEXT
520 FOR K=0 TO 79:PRINT#1,CHR$(C%(K));:PRINT CHR$(C%(K));:NEXT
530 PRINT#1,CR$;:PRINT CR$;
540 FOR I=1 TO 50: NEXT
550 IF PEEK(&H6F84)-PEEK(&H6F86)=0 THEN 540 ELSE EXEC &H6F20
560 NEXT J
570 IND=0:PRINT "EDIT END":PRINT "*";:RETURN
580 '
590 'Error routine
600 IF ERR=61 THEN PRINT       "Line buffer overflow": GOTO 670
610 IF ERR=62 OR ERR=56 THEN PRINT   "Bad file name": GOTO 680
620 IF ERR=63 THEN PRINT             "File not found": GOTO 680
630 IF ERR=64 THEN PRINT        "File Already EXists": GOTO 680
640 IF ERR=65 THEN PRINT             "Directory Full": GOTO 680
650 IF ERR=66 THEN PRINT  "Too Many Open Disk Files": GOTO 680
660 IF ERR=67 THEN PRINT           "Disk space full": GOTO 680
670 ON ERROR GOTO 0
680 RESUME 690
690 GO TO 300
700 '
```

行番号	説　　明
460	Editする行数を入力。
480	カーソルの位置を基点として記憶する。Editする文はそこから4行上にある。
490	スクリーンエディトを可能にするためのINPUT文。Editする文の最終行より下に¥ (CR) を入力するとEditが終りになる。(注3)
500	Editを行った文の画面上の一行を読みとる。
510	読みとったデータの格納番地を調べて、一文字づつ取り出す。
520	通信回線に一文字づつ送る。
530	一行送ったあと改行を行う。
540〜550	ホストから＊印が送られて来るのをまつ。
570	Edit終了を知らせて、リターン。
590〜690	エラー処理ルーチン。主としてディスクに関するエラーに対応。

（注3） INPUT文を用いているので、255文字以上は入力出来ない上、バッファエリアを大きくとっているので、あまり多くの訂正はできない。一度に訂正出来る行数は10行位である。途中で (CR) を入力すると、基点がずれるので注意。

```
710 'Log-on routine
720 PRINT#1,"$";"$";"$";"C";"O";"N";",";"T";"S";"S";",";",";",";"J";"I";"S";CR$;
730 PRINT "$$$CON,TSS":RETURN
740 '
750 'Stop routine
760 KEY(1) OFF:KEY(2) OFF:KEY(4) OFF:KEY(5)  OFF
770 KEY(6) OFF:KEY(7) OFF:KEY(8) OFF:KEY(9) OFF:KEY(10) OFF
780 CLOSE 1:CLOSE 2:CLOSE 3:CLOSE 4:CLOSE 5:ON ERROR GOTO 0
790 KEY 6,"PRINT": KEY 7,"LIST"+CHR$(34)+"LPT0:"+CHR$(34)+CHR$(13)
800 KEY 8,"KEY":KEY 9,"LOAD":KEY 10,"HARDC"+CHR$(13):KEY 1,"AUTO"
810 KEY 2,"LIST"+CHR$(13):KEY 4,"GO TO ":KEY 5,"CONT"+CHR$(13)
820 CONSOLE 0,25,0,0: POKE &HFFF8,1: POKE &HFFF9,&HDD:CLEAR,&H5000
830 END
840 '
850 'Send user-id
860 PRINT#1,"6";"0";"9";"2";"Δ";"Δ";"Δ";"Δ";"Δ";"Δ";CR$;
870 PRINT "6092ΔΔΔΔΔ":RETURN
880 '
890 'Send Break
900 EXEC &H6F88 :RETURN
910 '
```

行番号	説　　明
720	$$$CON, TSS, , JISを送るログオン手続き、一字ごとにしているのは9600BPSに対するタイミングのためで、300, 1200では一括して送ることが出来る。
750〜830	本プログラムを終了したあとマイコンを切らずに続けて使う場合、本プログラムで定義した種々のことをもとに戻さないと後遺症が残る。そのためキー割込をOFFにし、PFキー文字を元に戻し、ベクトルも書きなおす。
860	ID番号の送出。各自のID番号を△印のところに書き込んで下さい。
890〜900	ブレーク信号送出。$FD07_H$番地に$2E_H$を書き込めば良いのでBasic でも書けるが、その場合は書込を30回位つゞける必要がある。

```
920 'File Loging
930 IF LOING THEN 1010 ELSE INPUT "Enter write file name";F$
940 IF F$="FILES" OR F$="files" THEN FILES: GOTO 930
950 F$="O:"+F$:OPEN "O",5,""+F$:LOING=1:IND=1
960 PRINT#1,"P";"T";"O";"N";" ";"P";"P";"T";CR$;
970 FOR I=1 TO 1000:NEXT I:RETURN
980 I1=PEEK(&H6F85)*256+PEEK(&H6F86):IND=0
990 PRINT#1,"L";"I";"S";"T";CR$;:GO TO 380
1000 '
1010 'Write File on the Disk
1020 I2=PEEK(&H6F85)*256+PEEK(&H6F86): IF I1<I2 THEN 1030 ELSE 1040
1030 FOR I=I1 TO I2-2:    PRINT#5,CHR$(PEEK(I));:NEXT I :GO TO 1060
1040 FOR I=I1 TO &H6EFF:  PRINT#5,CHR$(PEEK(I));:NEXT I
1050 FOR I=&H2500 TO I2-2:PRINT#5,CHR$(PEEK(I));:NEXT I
1060 PRINT F$; "  File close"
1070 CLOSE#5: LOING=0:IND=3: RETURN
1080 '
1090 IND=0:PRINT#1,"P";"T";"O";"F";"F";CR$;:GO TO 380
1100 '
```

行番号	説　　明
930	初めてPF7キーを押した（LOING=0）のであれば、ディスクにSAVEするときの名前をたずねる。
950	INPUTされた名前でディスクファイルをOPEN。
960 970	送信を紙テープ形式で行うようホストに知らせる。（文末に$$$EOF をつけてもらうため）　返事をもらう時間待ち。
980	バッファの始め番地をＩ１として記憶。
990	ホストにリスト出力の要求。
1010	二度目にPF7キーが押されたときに来るルーチン。
1020	バッファの終り番地をＩ２として記憶。
1030 1050	バッファからデータを取り出して、ディスクに書き込む。
1070	終了後ディスクファイルを閉じる。
1090	ホストに紙テープリーダー使用終了を知らせる。

```
1110 'Read File from the Disk
1120 IF LOING THEN 1250
1130 IND=2:PRINT#1,"R";"E";"A";"D";" ";"P";"P";"T";CR$;
1140 FOR I=1 TO 1000:NEXT I:RETURN
1150 INPUT "Enter read file name";F$
1160 IF F$="FILES" OR F$="files" THEN FILES: GOTO 1150
1170 F$="0:"+F$:OPEN "I",4,""+F$: LOING=1
1180 L$=INPUT$(1,#4)
1190 IF EOF(4) OR LOING=0 THEN 1230
1200 PRINT L$;:PRINT#1,L$;
1210 IF LISING THEN PRINT#3,L$
1220 GOTO 1180
1230 IND=0:CLOSE#4:LOING=0:PRINT L$:GO TO 380
1240 '
1250 'Stop the read file routine
1260 PRINT F$; "  File close":LOING=0:RETURN
1270 '
1280 'Printer Routine
1290 IF LISING THEN LISING=0:POKE &H6F87,0:CLOSE 3 :RETURN
1300 LISING=1:POKE &H6F87,1 :OPEN "O",3,"LPT0:(W)" :RETURN
1310 '
```

行番号	説　　明
1110	ディスク内のデータをホストに送るルーチン。
1130	データを紙テープリーダーから読むことをホストに知らせる。
1150	読むべきファイル名を聞く。
1170	INPUTされたファイル名でディスクファイルOPEN。
1180	ディスクから一文字づつ読む。（300，1200BPS用ならばLINE INPUTを用いて一括入力してもよい）
1200	ホストへデータを一文字づつ送る。
1220	終了後ファイルを閉じる。
1290	プリンターの使用を中止するなら、フラグ変数LISINGとフラグ$6F87_H$番地を0にし、プリンタをclose。
1300	プリンターを使用するなら、LISINGと$6F87_H$番地を1にし、プリンタをOPEN。

```
1320 'Menu
1330 KEY 6,"Printer 80": KEY 7,"List Save":KEY 8,"List Out"
1340 KEY 9,"80 OR 132":KEY 10,"Break"
1350 KEY 1,"Logon": KEY 2,"User ID": KEY 4,"Edit": KEY 5,"STOP"
1360 RETURN
1370 '
1380 'Printer 80 or 132
1390 IF LNG THEN LNG=0:IF LISING THEN 1420 ELSE 1450
1400 LNG=1:IF LISING THEN 1410 ELSE 1430
1410 PRINT#3,CHR$(&H0F):KEY 6,"PRINTER 132":RETURN
1420 PRINT#3,CHR$(&H12):KEY 6,"PRINTER  80":RETURN
1430 OPEN "O",3,"LPT0:(W)":PRINT#3,CHR$(&H0F);CR$:CLOSE 3
1440 KEY 6,"PRINTER 132":RETURN
1450 OPEN "O",3,"LPT0:(W)":PRINT#3,CHR$(&H12);CR$:CLOSE 3
1460 KEY 6,"PRINTER  80":RETURN
1470 DATA FE,6F,83,F6,FD,07,C5,02,27,15,B6,FD,06,81,7F,27
1480 DATA 0E,A7,C0,11,83,6F,00,26,03,CE,25,00,FF,6F,83,3B
1490 DATA FE,6F,85,11,B3,6F,83,27,24,A6,C0,81,0F,27,10,8E
1500 DATA 6F,7E,A7,04,BD,6F,4E,7D,6F,87,27,03,BD,D8,9D,11
1510 DATA 83,6F,00,26,03,CE,25,00,FF,6F,85,20,D6,39,34,02
1520 DATA 86,FD,1F,8B,10,8E,FC,80,96,05,2B,FC,1A,50,86,80
1530 DATA 97,05,C6,08,5A,27,04,96,05,2A,F9,C6,05,A6,80,A7
1540 DATA A0,5A,26,F9,86,00,97,05,1C,AF,1F,8B,35,82,00,00
1550 DATA 03,01,00,25,00,25,00,00,1A,10,86,2E,B7,FD,07,1C
1560 DATA EF,39
```

行番号	説　　明
1320〜1360	30行の初期設定のときにGosubでやって来るルーチン。 PFキーの文字列を定義する。
1380〜1460	プリンターの一行を80字にするか132字にするかきめる。プリンタがMP-80の系統なら、$0F_H$を出力すると縮少小文字132字に、12_H を出力すると、通常文字80字になる。プリンタがどちらの字数かを表示するために、キー文字列を書きかえる。
1470〜1560	入力ルーチンのための機械語データ

3. 機械語のプログラムの説明

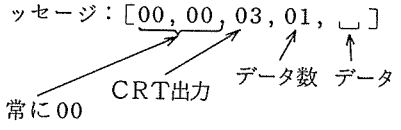
次に機械語の簡単な説明をする。なおデータのバッファは2500_H〜$6EFF_H$まで、IRQ割込処理ルーチンの格納ポインタは$6F83_H$、84_H番地、出力ルーチンのgetポインタは$6F85_H$、86_H番地である。プリンタのオンオフフラグは$6F87_H$番地を用いる。

```
6F00  FE      6F83      LDU   $6F83
6F03  F6      FD07      LDB   $FD07
6F06  C5      02        BITB  #$02
6F08  27      15        BEQ   $6F1F
6F0A  B6      FD06      LDA   $FD06
6F0D  81      7F        CMPA  #$7F
6F0F  27      0E        BEQ   $6F1F
6F11  A7  C0            STA   ,U+
6F13  1183    6F00      CMPU  #$6F00
6F17  26      03        BNE   $6F1C
6F19  CE      2500      LDU   #$2500
6F1C  FF      6F83      STU   $6F83
6F1F  3B                RTI
```

IRQ割込処理ルーチン。6F83$_H$番地に格納してあるポインタをUレジにとり、RS232CのステータスFD07$_H$を調べ受信可ならFD06$_H$からデータをAレジにとり、7F$_H$を除去後、バッファにデータを格納、格納番地が6EFF$_H$になると次を2500$_H$にする。

```
6F20  FE      6F85      LDU   $6F85
6F23  11B3    6F83      CMPU  $6F83
6F27  27      24        BEQ   $6F4D
6F29  A6  C0            LDA   ,U+
6F2B  81      0F        CMPA  #$0F
6F2D  27      10        BEQ   $6F3F
6F2F  8E      6F7E      LDX   #$6F7E
6F32  A7  04            STA   $04,X
6F34  BD      6F4E      JSR   $6F4E
6F37  7D      6F87      TST   $6F87
6F3A  27      03        BEQ   $6F3F
6F3C  BD      D89D      JSR   $D89D
6F3F  1183    6F00      CMPU  #$6F00
6F43  26      03        BNE   $6F48
6F45  CE      2500      LDU   #$2500
6F48  FF      6F85      STU   $6F85
6F4B  20      D6        BRA   $6F23
6F4D  39                RTS
```

MAINルーチン。GETポインタは6F85$_H$。0F$_H$の除去を行い、6F7E$_H$から5バイトのサブCPUへのメッセージをもってCRT出力ルーチンに行く。プリンタ出力はインタープリンタ内のルーチンD89D$_H$を利用する。




```
6F4E  34  02            PSHS  A
6F50  B6      FD        LDA   #$FD
6F52  1F  8B            TFR   A,DP
6F54  108E    FC80      LDY   #$FC80
6F58  96      05        LDA   $05
6F5A  2B      FC        BMI   $6F58
6F5C  1A      50        ORCC  #$50
6F5E  86      80        LDA   #$80
6F60  97      05        STA   $05
6F62  C6      08        LDB   #$08
6F64  5A                DECB
6F65  27      04        BEQ   $6F6B
6F67  96      05        LDA   $05
6F69  2A      F9        BPL   $6F64
6F6B  C6      05        LDB   #$05
6F6D  A6  80            LDA   ,X+
6F6F  A7  A0            STA   ,Y+
6F71  5A                DECB
6F72  26      F9        BNE   $6F6D
6F74  86      00        LDA   #$00
6F76  97      05        STA   $05
6F78  1C      AF        ANDCC #$AF
6F7A  1F  8B            TFR   A,DP
6F7C  35  82            PULS  A,PC
```

CRT一文字出力ルーチン。CRTに一文字出力を行うため、メッセージをサブCPUと共通番地FC80$_H$～84$_H$に格納する。サブCPUと会話中はFIRQ割込を禁止する。FD05$_H$に80$_H$を書きこむと、サブCPUがHALTになり、0$_H$を書きこむとサブCPUのHALTが解除される。

```
6F8B  1A      10        ORCC  #$10
6F8A  86      2E        LDA   #$2E
6F8C  B7      FD07      STA   $FD07
6F8F  1C      EF        ANDCC #$EF
6F91  39                RTS
```

ブレークルーチン。RS232Cステータス番地FD07$_H$に2E$_H$を送る。この間、IRQ割込禁止

4. 使 用 法

本プログラムをロードしRUNさせると、半二重か全二重かを聞いて来る。センターのTSSは半二重なので0を入力する。(例えば阪大情報処理教育センターは全二重なので1を入力する。)次にボーレイトを聞いて来るので、300, 1200, 9600のいづれかを入力する。たゞしこの段階でマイコン本体うらのディップスイッチが正しくセットされているか確認する必要がある。次にログオンをうながされるので、まず電話をかけ通信ラインをONにしてから、PF1を押す。たゞちに次のID番号を聞かれるのでPF2を押す。(大勢で使う場合はID番号も異るので、この段階からキーボードからの入力になる。)以下ターミナルとしてホストとキーボードで交信する。Byeコマンドを入力してホストと切れたあとは電源を切って終りにするか、さもなくばPF5を押して本プログラムを終了する。途中で何か不都合があり、ちょっとBasicモードに戻りたいときは、STOPキーを押すとBasicモードに戻るので、そこで作業をしたのち、CONT CR とすればまた交信状態に帰る。これで復帰できないときはPF3を押してRUNさせてもよい。

次にインテリジェント機能について順に説明する。

1) Edit

まず、List A-B でEditに必要な行を画面に出力しておく。次いでPF4を押すとEditする行数を聞いてくるので、行数を入力するとカーソルが点滅をはじめる。そこで、[←], [→], [↑], [↓] などのキーや、DEL, INSキーを用いて画面エディットを行う。(注4) キーを押す回数が255回以上になるとオーバーフローするので、一度にEditする行数は10行以下にした方がよい。すべての訂正が終ったら、Edit終了を知らせるために¥を入力し CR を押すと画面に表示されている行を一行づつセンターに送ってEditが完了する。¥を入力する場所は修正する文の最後の行より下の行でなくてはならない。

2) ホストのプログラムをディスクにSAVEする

まずOLDコマンドでカレントファイルにSAVEするプログラムを出しておく。PF7を押すと、ファイル名を聞いてくる。これはディスクにSAVEするときの名前である。名前はすでにディスクに登録されているものであってはならない。ファイル内にどんな名前が登録されているかを知りたければ、このときFILESと入力すると登録されているファイル名が画面にあらわれるので、そこにない名前をつければよい。ファイル名を入力すると画面にプログラムのリストが表示され始め、最後に$$$EOFで止まる。ここでもう一度PF7を押すとバッファから、ディスクにデータが書き込まれ、FILE CLOSE が表示されて終了する。あまり長いプログラムは一度にSAVEで

(注4) 挿入(INS)を行う時は、あらかじめその行の不用な部分の空白(たとえばその行の文章の後につづく空白)を挿入の個数だけ消去(DEL)した後で行う必要がある。

きず、分割してSAVEしなければならない。

3）ディスク上のプログラムをホストに送る

まずNEWコマンドでカレントファイルをなくしておく。次いでPF8を押すとファイル名を聞いてくるので、送るべきディスク上のファイル名を入力する。FILESと入力してディスクに登録されているファイル名を確認することもできる。$$$EOFが出てビルドモードで表示がとまれば終りである。

4）プリンタ

プリンタのONとOFFはTERMモードで用いたときと同様、PF6を押すとONもう一度押すとOFFである。300，1200BPSではいつでも有効であるが、9600BPSではリストをとっている途中などは受付られない。表示がとまって入力待ちになっているときだけ有効である。プリンタがMP-80系であるとき、縮少文字で132字を一行とする。あるいは一行を80字とする選択はPF9を押して出来る。現在がどちらのモードであるかはPF6の表示文字にあらわれるのでそれを見て知ることができる。

5）ブレーク

PF10を押すとビルドモードにもどる。そこで次のコマンドを入力すると、しばしばUNKNOWNという返事が返って来るがもう一度コマンドを入力すると正常になる。9600BPSではリスト出力中にブレークをかけても停止がおこらず全部リストが出たあとでブレークがかゝる。

6）その他

本プログラムはセンターとの交信以外にマイコン間のコミニケーションにも使える。例えば異る種類のミニフロッピディスクのプログラムをコミニケーションラインを通じて、別のディスクに格納することも出来る。

5. おわりに

以上FM-8用のインテェリジェントターミナルについて説明したが、便利に使うためにはさらに改良すべき点も多々あることと思う。大部分 Basic で書かれ、機械語の部分も簡単なものなので利用者の方々が各自使い勝手のよいように改良してゆかれることを期待する。特に、送信の方は Basic のまゝなので、9600BPS に十分対応出来ていない点、ディスクに格納するプログラムはデータファイルとして扱われるのでLOAD コマンドなどを用いて中味を見ることが出来ない点、ディスクへの書き込みがオンラインであく量に制限のある点、グラフィック機能のない点などが改良を待つところである。また9600BPSでもプリンタを使うと通信はとっくに終っているのにプリンターが、バッファにたまったデータを出力するのに時間がかかり、1200BPSとあまり変らないというのが実感である。9600BPSの場合プリンタ自体にバッファでもないと効率はわるい。

本文を書くにあたっていろいろご教示いたゞいた計算センター藤井博氏とプログラムの作成に協力いたゞいた大阪大学工学部産業機械工学科の大学院生小田晃一君に厚く感謝いたします。

6. 追　記

原稿提出後、本プログラムを9600 BPS で使用しているとき、ブレークがほとんどきかないという点を改良した。その方法はSTOPキーをブレークキーとして使用することである。STOPキーを押すとハード的にFIRQ割込がかゝるので、これならリスト出力中でも確実に割込める。たゞし、使用中に本プログラムを一時止めるための、本来のSTOPキーの役割も残したいので、PF−10のキーを押すと本来のSTOPキーにもどるように工夫した。またSTOPキーがどちらの状態にあるかをPF−10のファンクションキー表示に表示させることにしている。

以下に9600BPSで使用するブレークのためのプログラムの変更部分のみをのせる。

```
 45 ISTP=0
110 FOR I=&H6F00 TO &H6FB4:READ D$:DA=VAL("&H"+D$):POKE I,DA:NEXT
120 POKE &HFFF8,&H6F:POKE &HFFF9,&H00:POKE &HFFF6,&H6F:POKE &HFFF7,&H8B
250 ON KEY(10)GOSUB  890 :'Change STOP
890 'Change STOP routine
900 IF ISTP THEN 903
901 POKE &HFFF6,&H01:POKE &HFFF7,&HE0:ISTP=1
902 KEY 10,"Change STOP(B)":RETURN
903 POKE &HFFF6,&H6F:POKE &HFFF7,&H8B:ISTP=0
904 KEY 10,"Change STOP(T)":RETURN

1550 DATA 03,01,00,25,00,25,00,00,34,02,34,3C,EC,66,EF,66
1560 DATA CA,80,34,02,34,04,4F,1F,8B,B6,ED,04,85,02,26,12
1570 DATA C6,2E,F7,FD,07,F7,FD,07,F7,FD,07,F6,FD,04,C5,02
1580 DATA 27,F9,7E,CC,81
```

上にみるように、コンソールの表示がSTOP(T)のときにSTOPキーを押すとTSSブレークがかゝり、STOP(B)のときにSTOPキーを押すとBasicプログラムがSTOPする。この切り換えをPF10のキーを押すことで行うものである。はじめはSTOP(T)にセットしている。

この方法で、9600BPSでリスト出力中にSTOPキーを押すと、たゞちには画面表示はとまらない。10数行表示してからビルドモードの＊印がかえって来る。これはブレークはすぐにきいているのであるが、そのときすでにバッファに入っている分を出力してから表示がとまるので、あせらずにすこし待つとよい。

この他、1010行から1050行でデータを一字づつデータに書き込んでいる部分をSAVEM""+F$, I1, I2, I1 として一挙にセーブ出来ることがわかり現在はその方式に変更している。こうすると950行は不用になるので、ここで6F83, 6F84 および6F85, 6F86 番地にそれぞれ2500をPOKEして、常にI1＝2500_Hにセットして、1030～1050 行のような場合分けはする必要がなくなった。