



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	<a href="https://hdl.handle.net/11094/65557">https://hdl.handle.net/11094/65557</a>
rights	
Note	

*The University of Osaka Institutional Knowledge Archive : OUKA*

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

The University of Osaka

## 9600 BPS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300 BPS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<sub>H</sub><sup>(注1)</sup>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換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、一つの通信が終わ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1200 BPSや9600 BPS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9600 BPS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、9600 BPS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。<sup>(注2)</sup>すると9600 BPS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、1200 BPS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<sub>H</sub>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との通信に当たっているが、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。

PF 1 ..... ログオン

PF 2 ..... ID番号送信

PF 3 ..... RUN (CR) . 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。

PF 4 ..... EDIT. スクリーンエディットを行う。

PF 5 ..... 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。

PF 6 ..... プリンタのオンオフ。

PF 7 ..... 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。

PF 8 ..... ディスクにあるデータをホストに送信。

PF 9 ..... プリンタを一行80字にするか132字にするかの選択キー。

PF 10 ..... 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:LN6=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 "Q", 1, "COM0:(F7E1)": OPEN "I", 2, "COM0:(F7E1)":GO TO 110
100 OPEN "Q", 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 &HFFFB,&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
3 0	80 文字、25 行モードにし、2500 <sub>H</sub> 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。
4 0	プリンター、ディスクなどのフラグを 0 にセット。
6 0	全二重か半二重かの選択、センター用には 0 を入力。
7 0	BPS 入力。
8 0 } 1 0 0 }	300BPS に対しては SLOW モード、それ以外には FAST モードで、RS 232 C ポートを open。
1 1 0	受信用機械語ルーチンを読み込む。
1 2 0	IRQ 割込ベクトルを書きかえる。
1 4 0 } 2 5 0 }	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。
2 6 0	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 }	<input type="text" value="←"/> キーまたはBSキーならバックスペース処理、 <input type="text" value="→"/> , <input type="text" value="↑"/> , <input type="text" value="↓"/> 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なる。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
4 6 0	Edit する行数を入力。
4 8 0	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。
4 9 0	スクリーンエディットを可能にするための INPUT 文。Edit する文の最終行より下に $\textcircled{\text{CR}}$ を入力すると Edit が終りになる。 <sup>(注3)</sup>
5 0 0	Edit を行った文の画面上の一行を読みとる。
5 1 0	読みとったデータの格納番地を調べて、一文字ずつ取り出す。
5 2 0	通信回線に一文字ずつ送る。
5 3 0	一行送ったあと改行を行う。
5 4 0 } 5 5 0 }	ホストから * 印が送られて来るのをまつ。
5 7 0	Edit 終了を知らせて、リターン。
5 9 0 } 6 9 0 }	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ため、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  $\textcircled{\text{CR}}$  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 &H6FBB :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	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で後遺症が残る。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。
830	
860	ID番号の送出。各自のID番号をΔ印のところに書き込んで下さい。
890	ブレイク信号送出。FD07 <sub>H</sub> 番地に2E <sub>H</sub> 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回位づける必要がある。
900	

```

920 'File Logging
930 IF LOING THEN 1010 ELSE INPUT "Enter write file name";F$
940 IF F$="FILES" OR F$="files" THEN FILE$: GOTO 930
950 F$="0:"+F$:OPEN "0",5,""+F$: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
9 3 0	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。
9 5 0	INPUTされた名前でディスクファイルをOPEN。
9 6 0	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) 返事もらう時間待ち。
9 7 0	バッファの始め番地をI1として記憶。
9 8 0	ホストにリスト出力の要求。
1 0 1 0	二度目にPF7キーが押されたときに来るルーチン。
1 0 2 0	バッファの終り番地をI2として記憶。
1 0 3 0 } 1 0 5 0 }	バッファからデータを取り出して、ディスクに書き込む。
1 0 7 0	終了後ディスクファイルを閉じる。
1 0 9 0	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";" "; "F";"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$="O:"+F$: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
1 1 1 0	ディスク内のデータをホストに送るルーチン。
1 1 3 0	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。
1 1 5 0	読むべきファイル名を聞く。
1 1 7 0	INPUTされたファイル名でディスクファイルOPEN。
1 1 8 0	ディスクから一文字ずつ読む。(300, 1200BPS用ならばLINE INPUT を用いて一括入力してもよい)
1 2 0 0	ホストヘデータを一文字ずつ送る。
1 2 2 0	終了後ファイルを閉じる。
1 2 9 0	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 <sub>H</sub> 番地を0にし、プリンタをclose。
1 3 0 0	プリンターを使用するなら、LISINGと6F87 <sub>H</sub> 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$(%HOF):KEY 6,"PRINTER 132":RETURN
1420 PRINT#3,CHR$(%H12):KEY 6,"PRINTER 80":RETURN
1430 OPEN "O",3,"LPT0:(W)":PRINT#3,CHR$(%HOF):CR#:CLOSE 3
1440 KEY 6,"PRINTER 132":RETURN
1450 OPEN "O",3,"LPT0:(W)":PRINT#3,CHR$(%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
1 3 2 0 } 1 3 6 0 }	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。
1 3 8 0 } 1 4 6 0 }	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、0FH を出力すると縮小小文字 132 字に、12H 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、キ一文字列を書きかえる。
1 4 7 0 } 1 5 6 0 }	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 2500H～6EFFH まで、IRQ 割込処理ルーチンの格納ポインタは 6F83H、84H 番地、出力ルーチンの get ポインタは 6F85H、86H 番地である。プリンタのオンオフフラグは 6F87H 番地を用いる。

6F00	FE		6F83
6F03	F6		FD07
6F06	C5		02
6F08	27		15
6F0A	B6		FD06
6F0D	81		7F
6F0F	27		0E
6F11	A7	C0	
6F13	1183		6F00
6F17	26		03
6F19	CE		2500
6F1C	FF		6F83
6F1F	3B		
6F20	FE		6F85
6F23	1183		6F83
6F27	27		24
6F29	A6	C0	
6F2B	81		0F
6F2D	27		10
6F2F	8E		6F7E
6F32	A7	04	
6F34	BD		6F4E
6F37	7D		6F87
6F3A	27		03
6F3C	BD		D89D
6F3F	1183		6F00
6F43	26		03
6F45	CE		2500
6F48	FF		6F85
6F4B	20		D6
6F4D	39		
6F4E	34	02	
6F50	B6		FD
6F52	1F	8B	
6F54	108E		FC80
6F58	96		05
6F5A	2B		FC
6F5C	1A		50
6F5E	86		80
6F60	97		05
6F62	C6		08
6F64	5A		
6F65	27		04
6F67	96		05
6F69	2A		F9
6F6B	C6		05
6F6D	A6	80	
6F6F	A7	A0	
6F71	5A		
6F72	26		F9
6F74	86		00
6F76	97		05
6F78	1C		AF
6F7A	1F	8B	
6F7C	35	82	
6F88	1A		10
6F8A	86		2E
6F8C	B7		FD07
6F8F	1C		EF
6F91	39		

LDU	#6F83
LDB	#FD07
BITE	#02
BEQ	#6F1F
LDA	#FD06
CMFA	#07F
BEQ	#6F1F
STA	,U+
CMFU	#6F00
BNE	#6F1C
LDU	#2500
STU	#6F83
RTI	
LDU	#6F85
CMFU	#6F83
BEQ	#6F4D
LDA	,U+
CMFA	#0F
BEQ	#6F3F
LDX	#6F7E
STA	#04,X
JSR	#6F4E
TST	#6F87
BEQ	#6F3F
JSR	#D89D
CMFU	#6F00
BNE	#6F4B
LDU	#2500
STU	#6F85
BRA	#6F23
RTS	
FSHS	A
LDA	#FD
TFR	A,DP
LDY	#FC80
LDA	#05
BMI	#6F58
ORCC	#50
LDA	#80
STA	#05
LDB	#08
DECB	
BEQ	#6F6B
LDA	#05
BPL	#6F64
LDB	#05
LDA	,X+
STA	,Y+
DECB	
BNE	#6F6D
LDA	#00
STA	#05
ANDCC	#AF
TFR	A,DP
PULS	A,PC
ORCC	#10
LDA	#2E
STA	#FD07
ANDCC	#EF
RTS	

IRQ割込処理ルーチン。6F83<sub>H</sub>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<sub>H</sub>を調べ受信可ならFD06<sub>H</sub>からデータをAレジにとり、7F<sub>H</sub>を除去後、バッファにデータを格納、格納番地が6EFF<sub>H</sub>になると次を2500<sub>H</sub>にする。

MAINルーチン。GETポインタは6F85<sub>H</sub>。0F<sub>H</sub>の除去を行い、6F7E<sub>H</sub>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<sub>H</sub>を利用する。

メッセージ: [00, 00, 03, 01, ]

常に00      CRT出力      データ数      データ

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<sub>H</sub>~84<sub>H</sub>に格納する。サブCPUと会話中はFIRQ割込を禁止する。FD05<sub>H</sub>に80<sub>H</sub>を書きこむと、サブCPUがHALTになり、0<sub>H</sub>を書きこむとサブCPUのHALTが解除される。





ブレークルーチン。RS232Cステータス番地FD07<sub>H</sub>に2E<sub>H</sub>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のいずれかを入力する。ただ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番号も異なるので、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、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。<sup>(注4)</sup> 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。すべての訂正が終わったら、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ず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、ここには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, 1200 BPSではいつでも有効であるが、9600 BPS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、縮小文字で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。9600 BPS 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かる。

### 6) その他

本プログラムはセンタートとの交信以外にマイコン間のコミュニケーションにも使える。例えば異なる種類のミニフロピディスクのプログラムをコミュ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、9600 BPS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9600 BPSでもプリンタを使うと通信はとくに終わ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、1200 BPSとあまり変わらないというのが実感である。9600 BPSの場合プリンタ自体にバッファでもない効率はわるい。

本文を書くにあたっていろいろご教示いただ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だ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 割込がかかるので、これならリスト出力中でも確実に割込める。ただし、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 9600 BPS 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 &HFFFB,&H6F:POKE &HFFF9,&H00:POKE &HFFF6,&H6F:POKE &HFFF7,&H8B
250 ON KEY(10)GOSUB B90 :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,80,34,02,34,04,4E,1E,8B,86,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 ブレークがかかり、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 PF 10 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、9600 BPS 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 行でデータを一字ずつデータに書き込んでいる部分を SAVEM " " + F\$, I 1, I 2, I 1 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 6F 83, 6F 84 および 6F 85, 6F 86 番地にそれぞれ 2500 を POKE して、常に I 1 = 2500<sub>H</sub> 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。