

# PET 2001用ハードコピーシステム

八田 信\*・高浪 五男\*\*・井上 克司\*\*

## Hard-copy system for PET 2001

Makoto HATTA, Itsuo TAKANAMI and Katsushi INOUE

### Abstract

We design a hard-copy system for PET2001 which is a popular personal computer. This system is not based on the IEEE-IB, but the usual interface using PIA 6520. The programs controlling a high-speed dot printer are written by machine language. the system can hard-copy (1) BASIC program lists, (2) patters on the video screen, and (3) output data in user's programs.

### 1. はじめに

PET 2001 用ハードコピーとしては、近日中にコモドール社より PET 2020 IEC パスプリンタが発売される予定であるが、比較的高価(約20万円)なものである。一方、私達は先に6502用ハードコピーシステムを<sup>1)</sup>作製したが、PET の CPU が 6502 であることより、前報告の技術を PET のハードコピーに適用できる。そのためには、PET のモニターを解読する必要があるが、これも先に報告した6502用逆アセンブラ<sup>2)</sup>を用いて行なうことができる。文献(1)、(2)を用いて、PET 2001 用ハードコピーシステムを製作したので、その機能、設計法、使用法について述べる。これらを十分理解することにより、PET に種々の(安価な)プリンタを接続できるのはもとより、PET のモニターについての部分的理解も得られる。

今回製作したシステムは、IEEE-IB規格は用いず、従来通りのPIAを用いたインタフェース方式で、プリンターのコントロールは機械語で書いている。この方法は IEEE-IB がまだ十分普及していない現状では最も手軽に製作できるハードコピー システムである。私達は手元にあった PA-6651-A (ドットプリンタ)を用いたが安価なプリンタも接続できる。

システムの機能は

- ① BASIC プログラムリスト、

- ② スクリーン上の画面、

- ③ ユーザプログラムの出力データ、

の各コピーができる。従って、アセンブル リストや逆アセンブルリストのコピーもできる。ただし、グラフィック キャラクターはプリンタの ROM が PET のそれと異なるため、片仮名が印字される。これは、プリンタの ROM を交換すれば勿論グラフィック キャラクターも印字できる。また、リバース文字はプリントできない。

### 2. ハードウェア

用いるプリンタは文献(1)の PA-6651-A である。

Table 1. Correspondence of the terminals of PET IEEE-IB and PIA6520

PET IEEE-IB	PIA 6520
DIO1	PB <sub>0</sub>
DIO2	PB <sub>1</sub>
DIO3	PB <sub>2</sub>
DIO4	PB <sub>3</sub>
DIO5	PB <sub>4</sub>
DIO6	PB <sub>5</sub>
DIO7	PB <sub>6</sub>
DIO8	PB <sub>7</sub>
SRQ	CB1

\* 大学院電気工学専攻

\*\* 電子工学科

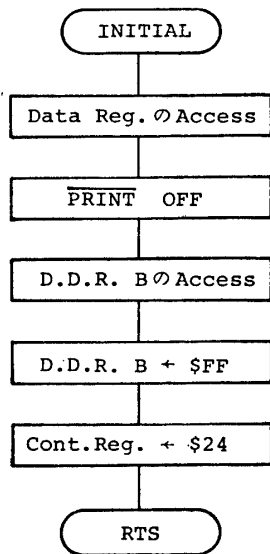


Fig.1 Flow-chart for the initial setting

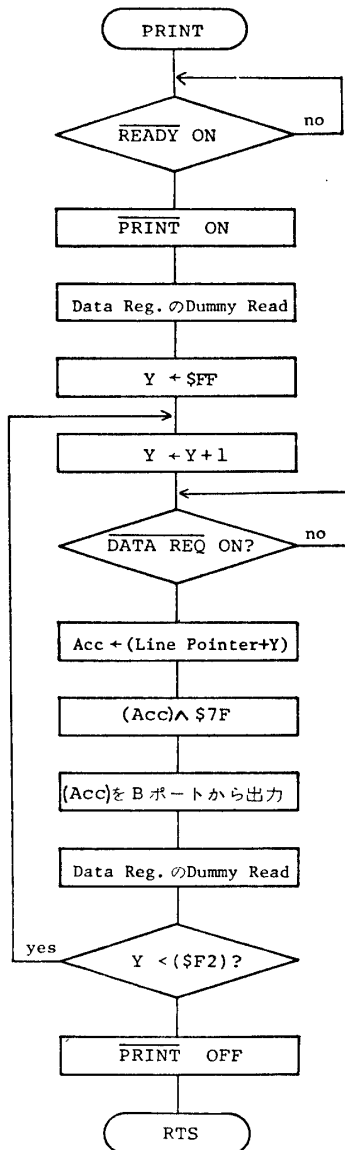


Fig.2 Flow-chart for print-out of one line

プリンタと PET とのインタフェースは文献 (1) のものがそのまま用いられる。PET の端子名称と PIA 6520 のそれとの対応は表 1 の通りである。

PIA 6520 とインタフェースとの接続は文献 (1) にあるので、それに従って PET とインタフェースとの接続を行なえばよい。

### 3. ソフトウェア

#### 3.1 初期設定

- $\overline{\text{PRINT}}$  信号を OFF.
- B ポートを出力ポートとする.
- OCB2 は Output Level Handshake mode にする.
- OCB1 は Interrupt Disable, Negative Edge On mode にする.
- 初期設定を終えたらデータレジスタをアクセスする.

以上の初期設定のフローチャートを Fig.1 に、プログラムリストを Fig.3 に示す。

#### 3.2 1 行のプリントアウト

プリントを開始する前に 1 行分のデータはあらかじめ特定のバッファエリアにストアしておく。このエリアの先頭アドレスは \$01 と \$02 番地のポインタ (Line Pointer) によって指定される。したがって、バッファエリアを任意の場所に置くことができる。例えば、ポインタが Video RAM エリアを指定すれば、スクリーン上の文字をプリントできる。

033A	A9 24	LDA #24	データレジスタのアクセス
033C	8D 23E8	STA #E823	
033F	A9 80	LDA #80	$\overline{\text{PRINT}}$ OFF
0341	8D 22E8	STA #E822	
0344	A9 20	LDA #20	DDR のアクセス
0346	8D 23E8	STA #E823	
0349	A9 FF	LDA #FF	B ポートを出力ポートに
034B	8D 22E8	STA #E822	
034E	A9 24	LDA #24	コントロールレジスタの設定
0350	8D 23E8	STA #E823	
0353	60	RTS	
0354	AD 23E8	LDA #E823	) if $\overline{\text{READY}}$ = OFF then 0354
0357	10 FB	BPL #0354	
0359	A9 20	LDA #20	$\overline{\text{PRINT}}$ ON
035B	8D 22E8	STA #E822	
035E	AD 22E8	LDA #E822	
0361	A0 FF	LDY #FF	Y = 0
0363	C8	INY	
0364	AD 23E8	LDA #E823	) if $\overline{\text{DATA REQ}}$ = OFF then 0364
0367	10 FB	BPL #0364	
0369	B1 01	LDA (\$01), Y	Acc + (Line Pointer + Y),
036B	29 7F	AND #7F	
036D	8D 22E8	STA #E822	1 文字データを出力
0370	AD 26E8	LDA #E826	
0373	C4 F2	CPY #F2	
0375	30 EC	BMI #0363	) if Y < (End Pointer) then 0363
0377	A9 A0	LDA #A0	$\overline{\text{PRINT}}$ OFF
0379	8D 22E8	STA #E822	
037C	AD 22E8	LDA #E822	
037F	60	RTS	

Fig.3 Program list for print-out of one line

1 行の印字文字数は \$F2 番地 (End Pointer) の内容で決定される。スクリーンの 1 行は 40 文字であるが、プリンタは 80 文字印字できるので、スクリーンで 2 行

に亘る内容は 1 行に印字することもできる。\$F2 番地の内容はモニタープログラムによって制御されている。

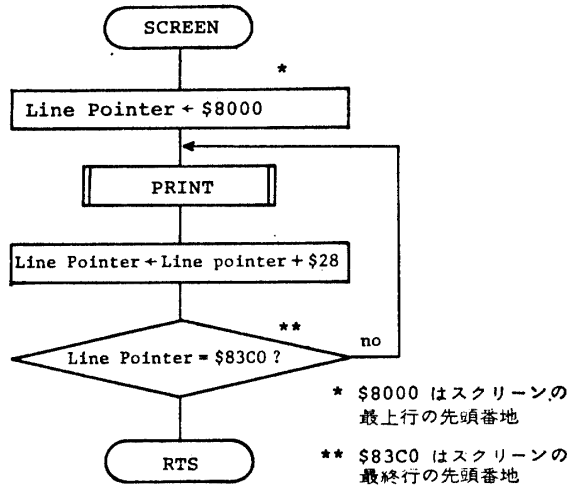


Fig. 4 Elow-chart for hard copy of the screen

```

PRINT SCREEN
033A 03A7

033A A9 24 LDA #24
033C 8D 23E8 STA #E823
033F A9 80 LDA #80
0341 8D 22E8 STA #E822
0344 A9 20 LDA #20
0346 8D 23E8 STA #E823
0349 A9 FF LDA #FF
034B 8D 22E8 STA #E822
034E A9 24 LDA #24
0350 8D 23E8 STA #E823
0353 60 RTS

0354 AD 23E8 LDA #E823
0357 10 FB BPL #0354

0359 A9 20 LDA #20
035B 8D 22E8 STA #E822
035E AD 22E8 LDA #E822
0361 A0 FF LDY #FF
0363 C9 TAY
0364 AD 23E8 LDA #E823
0367 10 FB BPL #0364

0369 B1 01 LDA (#01),Y
036B 29 7F AND #7F
036D 8D 22E8 STA #E822
0370 AD 26E8 LDA #E826
0373 C4 F2 CPY #F2
0375 30 EC BMI #0363

0377 A9 A0 LDA #A0
0379 8D 22E8 STA #E822
037C AD 22E8 LDA #E822
037F 60 RTS

0380 20 3A03 JSR #033A
0383 A9 80 LDA #80
0385 85 02 STA #02
0387 A9 00 LDA #00
0389 85 01 STA #01
038B 20 5403 JSR #0354
038E 98 TYA
038F 30 SEC
0390 65 01 ADC #01
0392 85 01 STA #01
0394 A5 02 LDA #02
0396 69 00 ADC #00
0398 85 02 STA #02
039A A5 02 LDA #02
039C C9 83 CMP #83
039E 30 EB BMI #038B
03A0 A5 01 LDA #01
03A2 C9 98 CMP #98
03A4 D0 E5 BNE #038B

03A6 60 RTS
    
```

PIA Bポートの初期設定  
Line Pointer = \$8000  
一行プリント  
Line Pointer Line Pointer+End Pointer  
Line Pointer = \$83C0?

Fig.5 Program list for hard-copy of screen

1 行のプリントアウトのフローチャートを Fig.2 にプログラム リストを Fig.3 に示す。このルーチンはハードコピーをとる上で最も基本となるもので、このルーチンを中心として画面のハードコピーやリストのコピーのプログラムは構成される。

### 3.3 スクリーンのハードコピー

プログラム中にサブルーチンの形で挿入することによって、その時点のスクリーン上の画面をプリントアウトする。ここでは画面全体のコピールーチンを示すが、プログラムの僅かな修正で特定の行をコピーするようにできる。

このルーチンのフローチャートを Fig.4 に、プログラムリストを Fig.5 に示す。

使い方は、プログラム中のコピーしたい画面を表示している位置に『SYS (896)』を挿入すればよい、使

```

10 PRINT"DEMO 4.0"
15 GOSUB700
20 T$=" ":S$=" ":GOSUB6000
30 GOSUB1000:REM LOGO
31 SYS(896)
40 T$="R X":S$="R X":GOSUB6000
60 GOSUB1300:REM WHAT YOU GET
61 SYS(896)
62 GOSUB3000:REM LUNAR LANDER
63 SYS(896)
70 T$="R * X":S$="R * X":GOSUB6000
90 GOSUB1600:REM FUTURE PLANS
91 SYS(896)
94 T$=" ":S$=" ":GOSUB6000
95 GOSUB1800:REM BAR GRAPH
96 SYS(896)
100 T$=" ":S$=" ":GOSUB6000
101 GOSUB4000:REM I/O INFO
102 SYS(896)
170 GOSUB5000:REM ENTERPRISE
171 SYS(896)
180 GOSUB9000:REM TIME
182 SYS(896)
190 GOSUB7000:REM R2D2
191 SYS(896)
500 GOTO20
    
```

Fig.6 An example of hard-coping the screens displayed in the execution of the DEMO tape

用例として、PET DEMO テープの画面をコピーする場合を Fig.6 に示す。(Fig.6は以下で述べる‘リストのハードコピー’プログラムによってプリントしたものである)

3.4 プログラム リストのハードコピー

プログラムの手順としては、BASIC テキストエリアの内容を翻訳してスクリーンに出力し、それをプリンタにコピーするようになっている。フローチャート

を Fig.7 に示す。これについて重要な点を述べると、

○[テキストの翻訳] 翻訳される文の先頭番地は \$AE と \$AF 番地 (Text Pointer) によってポイントされる。したがって、最初の Text Pointer の値は \$0401 である。コマンドはテキスト内では \$80 以上の16進数で示されるが (例えば ‘INPUT’ は \$85), これらは BASIC 中のコマンドテーブル (\$C092~\$C18E 番地) をサーチして翻訳する。

○[文の終り] 1 バイトの \$00 で示される。

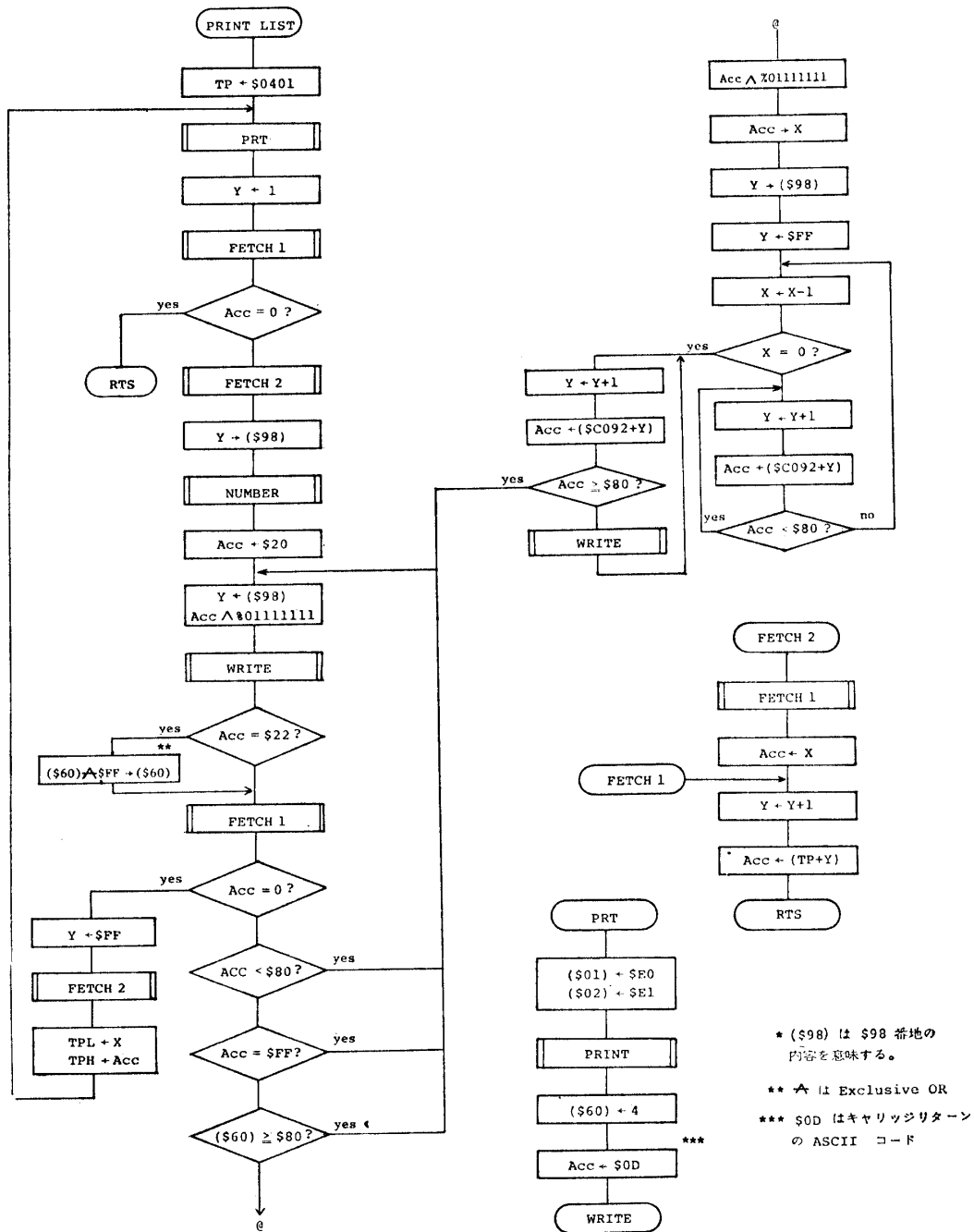


Fig.7 Flow-chart for hard-copy of program lists

10 INPUT A,B\$  
20 PRINT A,B\$  
30 END

PRINT LIST  
033A 0400

Fig.8a. An example of BASIC program

PRINT TEXT 0400 0426	↓ アドレス (16 進表示)	↓ 内容 (16 進表示)	↓ ASCII 文字
0400 00			
0401 0C			
0402 04			} 次の文の先頭アドレス (ここでは \$040C)
0403 0A			} 文番号 (2 進表示) (ここでは 10)
0404 00			コマンド (INPUT)
0405 85			スペース
0406 20			
0407 41	A		
0408 2C	,		
0409 42	B		
040A 24	,		
040B 00			文の終了
040C 17			} 次の文の先頭アドレス (\$0417)
040D 04			
040E 14			} 文番号 (20)
040F 00			コマンド (PRINT)
0410 99			スペース
0411 20			
0412 41	A		
0413 2C	,		
0414 42	B		
0415 24	,		
0416 00			文の終了
0417 1D			} 次の文の先頭アドレス (\$041D)
0418 04			
0419 1E			} 文番号 (30)
041A 00			コマンド (END)
041B 90			:文の終了
041C 00			} プログラムの終了
041E 00			
041F 24	*		
0420 24	*		
0421 24	*		
0422 24	*		
0423 24	*		
0424 24	*		

033A A9 24	LDA #024	
033C 8D 23E8	STA #E823	
033F A9 90	LDA #80	
0341 8D 22E8	STA #E822	
0344 A9 20	LDA #020	
0346 8D 23E8	STA #E823	
0349 A9 FF	LDA #FF	
034B 8D 22E8	STA #E822	
034E A9 24	LDA #024	
0350 8D 23E8	STA #E823	
0353 60	RTS	
0354 AD 23E8	LDA #E823	
0357 10 FB	BPL #0354	
0359 A9 20	LDA #020	
035B 8D 22E8	STA #E822	
035E AD 22E8	LDA #E822	
0361 A0 FF	LDY #FF	
0363 C8	INY	
0364 AD 23E8	LDA #E823	
0367 10 FB	BPL #0364	
0369 B1 01	LDA (#01),Y	
036B 29 7F	AND #7F	
036D 8D 22E8	STA #E822	
0370 AD 26E8	LDA #E826	
0373 CA F2	CPY #F2	
0375 30 EC	BMI #0363	
0377 A9 00	LDA #00	
0379 8D 22E8	STA #E822	
037C AD 22E8	LDA #E822	
037F 60	RTS	
0380 20 3A03	JSR #033A	PIA B ポートの初期化
0383 A9 04	LDA #04	
0385 85 AF	STA #AF	テキストポインタ + \$0401
0387 E6 AE	INC #AE	
0389 20 EC03	JSR #03EC	プリントアウト
038C A0 01	LDY #01	
038E B1 AE	LDA (#AE),Y	
0390 F0 ED	BEQ #037F	if (テキストポインタ+2) = 0 then RTS 終了判断
0392 20 E403	JSR #03E4	文番号の fetch
0395 84 98	STY #98	Y の save
0397 20 9FDC	JSR #DC9F	文番号の表示
039A A9 20	LDA #20	Acc + スペース
039C A4 98	LDY #98	
039E 29 7F	AND #7F	
03A0 20 49CA	JSR #CA49	Acc の内容をスクリーンに表示
03A3 C9 22	CMP #22	
03A5 D0 06	BNE #03AD	if Acc = " then #03AD
03A7 A5 60	LDA #60	
03A9 A9 FF	EOR #FF	
03AB 85 60	STA #60	引用符があれば #60 の内容を 反転させる
03AD C8	INY	
03AE B1 AE	LDA (#AE),Y	
03B0 D0 0B	BNE #03BD	文の終了判断
03B2 A0 FF	LDY #FF	
03B4 20 E403	JSR #03E4	テキストポインタ + 次の文のアドレス
03B7 86 AE	STX #AE	
03B9 85 AF	STA #AF	
03BB 10 CC	BNE #03B9	
03BD 10 E1	BPL #03A0	if Acc < #80 then #03A0
03BF C9 FF	CMP #FF	
03C1 F0 DD	BEQ #03A0	
03C3 24 60	BIT #60	引用符ならば #03A0へ
03C5 C0 D9	BMI #03A0	
03C7 29 7F	AND #7F	
03C9 AA	TAX	
03CA 84 98	STY #98	
03CC A0 FF	LDY #FF	
03CE CA	DEX	
03CF 30 03	BMI #03D9	コマンドの解決
03D1 C8	INY	
03D2 B9 92C0	LDA #C092,Y	
03D5 10 FA	BPL #03D1	
03D7 30 F5	BMI #03CE	
03D9 C8	INY	
03DA B9 92C0	LDA #C092,Y	
03DD 30 D0	BMI #07D0	
03DF 20 49CA	JSR #CA49	
03E2 D0 F5	BNE #03D9	
03E4 20 E303	JSR #03E8	2 バイトフエッチ
03E7 AA	TAX	
03E9 C8	INY	
03EB B1 AE	LDA (#AE),Y	1 バイトフエッチ
03ED 60	RTS	
03EC A6 E1	LDX #E1	Line Pointer の設定
03EE A5 E0	LDA #E0	
03F0 85 01	STA #01	
03F2 86 02	STX #02	
03F4 20 5403	JSR #0354	1 行のプリント
03F7 A9 04	LDA #04	#60のリセット
03F9 85 60	STA #60	
03FB A9 00	LDA #00	
03FD 4C 49CA	JMP #CA49	CR LF の出力

Fig.8b The format in the BASIC text for the BASIC program in Fig.8a

○[プログラムの終り] 2 バイトの \$00 で示される。(したがって、文の終りと合わせるとプログラムの終りは 3 バイトの \$00 が続くことになる。Fig.8 参照)。

○[サブルーチン] BASIC モニターのサブルーチンから次の二つを用いる。

WRITE (開始番地 \$CA49) : ACC の内容をスクリーンに表示する。

NUMBER (開始番地 \$DC9F) : ACC とレジスタ X の内容から文番号を計算しスクリーンに表示する。

このハードコピールーチンの使い方も、SYS コマンドによって (896)<sub>10</sub>番地からプログラムをスタートさせる。ここではリスト全体をコピーするプログラムを Fig.9 に示す。

4. プログラムの LOAD と SAVE

今まで述べてきたプログラムを書き込むエリアはカセット #2 のバッファエリアを用いる。しかし、この

Fig.9 Program list for hard-copy of BASIC program list

エリアは198バイトしかないため、リストコピーかスクリーンコピーのどちらか一方のプログラムだけしか入らない。したがって、場合に応じて必要なプログラムをLOADして使うことにする。

#### 4.1 機械語プログラムのカセットへのSAVE

PETではBASICテキストエリアのプログラムのSAVEは説明書にある通り容易であるが、他のエリアの機械語プログラムのSAVEについては明らかにされていない。ファイル文を使ってSAVEすることは可能であるが、実際に使用してみると、長いプログラムの場合には読みエラーが続出し、実用性に乏しい。そこで、このような場合でも信頼性高く、任意の番地から任意の番地までの内容をSAVEするプログラムを示す。

PETのモニター及びBASICを解読していくうちに、SAVEについて次の事が明らかになった。

- デバイスナンバーは\$F1番地に、
- SAVEの開始番地は\$F7、\$F8番地に、
- SAVEの終了番地は\$E5、\$E6番地に、
- ネーム長は\$EE番地に、

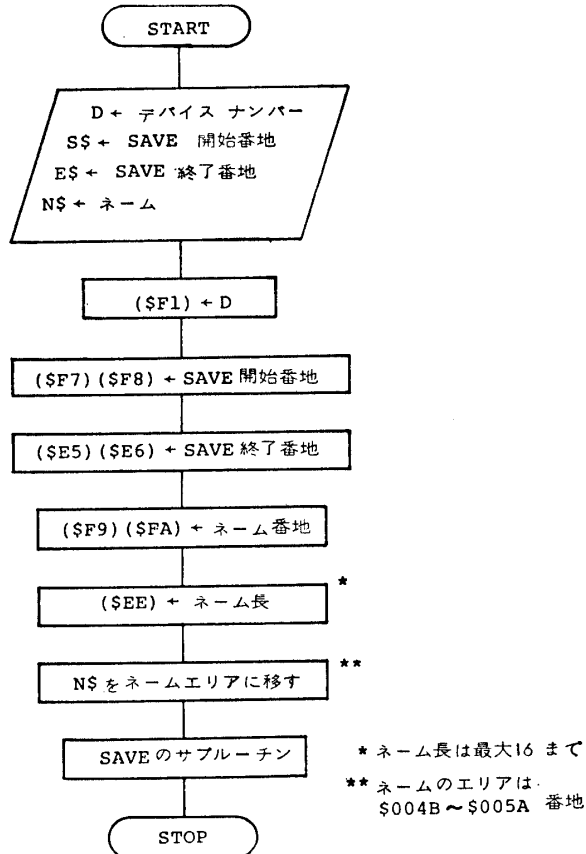


Fig.10 Flow-chart for saving machine language programs into the cassette taper recorder

それぞれストアする。

- ネームの格納場所は\$F9、\$FA番地で示される。
- 以上の5項目を設定してSAVEのサブルーチン(\$F6B1)へジャンプする。

以上のルーチンのフローチャートをFig.10に、プログラムリストをFig.11に示す。

カセットからのLOADの方法はBASICプログラムの場合と全く同じである。

#### 4.2 使用方法

?が画面に現われ入力待ちとなるので、

- デバイス ナンバー (10進数)、
- SAVE 開始番地 (16進数)、
- SAVE 終了番地 (16進数)、
- ネーム

○(CR): キャリッジリターン

の順でコンマをはさみながら入力する。例えば、\$033Aから\$03FF番地のプログラムに“PROG”というネームをつけてカセット #1にSAVEする場合は、

```

RUN CR
? 1, 033A, 03FF, PROG CR
  
```

```

1 REM***SAVE MACHINE***
10 INPUT D,$E,$N$
20 POKE241,D:REM DEVICE NUMBER
30 H=248:A$=S$
40 GOSUB 150
50 H=230:A$=E$
60 GOSUB 150
70 POKE249,75:REM NAME POINTER L
80 POKE250,0 :REM NAME POINTER H
90 L=LEN(N$)
100 POKE238,L:REM LENGTH OF THE NAME
110 FORK=1TOL
120 D=ASC(MID$(N$,K,1))
130 POKE1007+K,D
140 NEXTK
142 SYS(63153):REM SAVE ROUTIN
144 STOP
150 FORI=1T02:C=0
160 FORJ=1T02
170 B=ASC(MID$(A$,J+2*I-2,1))
180 IFB>=65 THEN B=B-7
190 B=B-48
200 C=16^(2-J)*B+C
210 NEXTJ
220 POKEH,C
230 H=H-1
240 NEXTI
250 RETURN
  
```

Fig.11 Program list for saving a machine language program into the cassette tape recorder

のようにする。(下線を引いた所を入力する)

## 5. む す び

パーソナル コンピュータ PET 2001 の周辺機能の拡張の一環として、これに高速のドットプリンタを接続して、ハードコピーシステムを作製した。PETのハードとソフトに関する資料はほとんど入手できないため、これのモニターの解読作業に多くの時間をとられた。しかし、モニター解読作業によってコンピュータの内部の仕組みが除々にわかってくるのは楽しみでもある。今後はフロップディスク、高速の PTR などの接続を考えている

## 参 考 文 献

- 1) 八田, 高浪, 井上: MCS-6502 を CPUとするマイクロコンピュータシステム—高速ドットプリンタ PA-6651-Aとのインターフェース—: 山口大学工学部研究報告, 29, 103 (1978).
- 2) 八田, 高浪, 井上: MCS-6502 マイクロコンピュータの逆アセンブラー—山口大学工学部研究報告, 29, 113 (1978).
- 3) 八田, 井上, 高浪: 6502用逆アセンブラー: トランジスタ技術, CQ 出版社, 15, 270~281 (1978).
- 4) 八田, 高浪, 井上: PET 2001 用逆アセンブラー: 山口大学工学部研究報告, 29, 299 (1979).
- 5) MCS6500 Microcomputer Family Hardware Manual: MOS TECHNOLOGY, INC. 1976.
- 6) PET 2001 ユーザーズ マニュアル: コモドールジャパン(株)

(昭和 53 年 10 月 14 日 受理)