Some Programs for the Epson HX-20

Martin Hepperle, 2025

The popular Japanese computer magazine I/O focused on Japanese computers like the PC-6001, PC-8001, FM-7, FM-8 or the Sharp MZ systems. A very small number of issues also contained programs for the Epson HC-20. The relevant pages have been collected in the following sections.

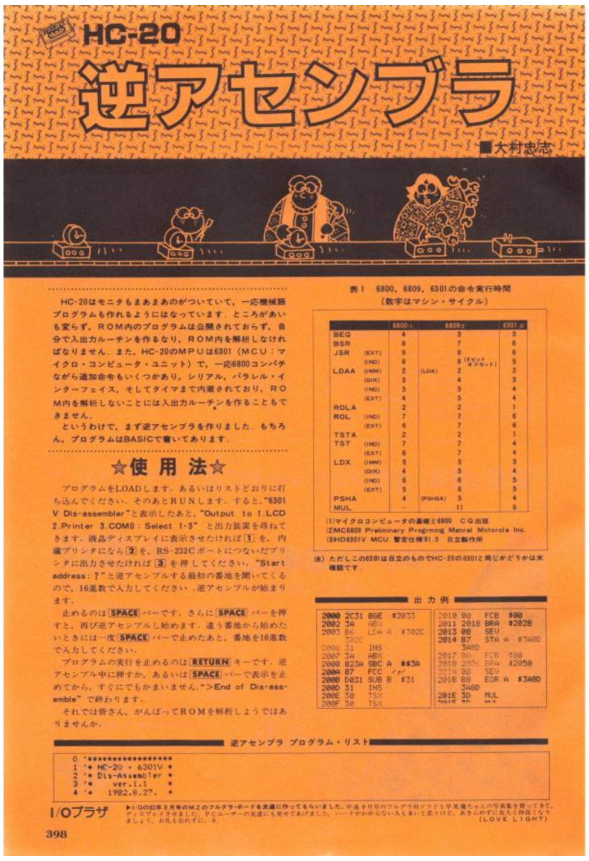
Disassembler

The November 1982 issue of I/O (V7N11) contained a BASIC listing of a simple disassembler for the HC-20. The BASIC program asks for a starting address and then disassembles memory into mnemonics. Output can be directed to the LCD screen, the RS232C interface or the printer.

漢字ROMボード
GP-IBインターフェイス
カラーコントローラ
HOBBY ELECTRONICS JOURNAL
IO
3次元迷路
COSMIC HOOLIGAN
PURSUIT

CONTENTS 1982年11月号

★イラスト=はらJIN+きむらしんじ+馬場隆信+川名保雄


HC-20

逆アセンブラ

■大村忠志

HC-20はモニタもまあまあのがついていて，一応機械語プログラムも作れるようにはなっています．ところがあいも変らず，ROM内のプログラムは公開されておらず，自分で入出力ルーチンを作るなり，ROM内を解析しなければなりません．また，HC-20のMPUは6301（MCU：マイクロ・コンピュータ・ユニット）で，一応6800コンパチながら追加命令もいくつかあり，シリアル，パラレル・インターフェイス，そしてタイマまで内蔵されており，ROM内を解析しないことには入出力ルーチンを作ることもできません．

というわけで，まず逆アセンブラを作りました．もちろん，プログラムはBASICで書いてあります．

☆使用法☆

プログラムをLOADします．あるいはリストどおりに打ち込んでください．そのあとRUNします．すると，"6301V Dis-assembler"と表示したあと，"Output to 1.LCD 2.Printer 3.COM0：Select 1-3" と出力装置を尋ねてきます．液晶ディスプレイに表示させたければ[1]を，内蔵プリンタになら[2]を，RS-232Cポートにつないだプリンタに出力させたければ[3]を押してください．"Start address：？"と逆アセンブルする最初の番地を聞いてくるので，16進数で入力してください．逆アセンブルが始まります．

止めるのは[SPACE]バーです．さらに[SPACE]バーを押すと，再び逆アセンブルし始めます．違う番地から始めたいときには一度[SPACE]バーで止めたあと，番地を16進数で入力してください．

プログラムの実行を止めるのは[RETURN]キーです．逆アセンブル中に押すか，あるいは[SPACE]バーで表示を止めてから，すぐにでもかまいません．">End of Dis-assemble" で終わります．

それでは皆さん，がんばってROMを解析しようではありませんか．

表1　6800，6809，6301の命令実行時間
（数字はマシン・サイクル）

		6800(1)		6809(2)	6301(3)
BEQ		4		3	3
BSR		8		7	6
JSR	(EXT)	9		8	6
	(IND)	8		8（8ビットオフセット）	5
LDAA	(IMM)	2	(LDA)	2	2
	(DIR)	3		4	3
	(IND)	5		5	4
	(EXT)	4		5	4
ROLA		2		2	1
ROL	(IND)	7		7	6
	(EXT)	6		7	6
TSTA		2		2	1
TST	(IND)	7		7	4
	(EXT)	6		7	4
LDX	(IMM)	3		3	3
	(DIR)	4		5	4
	(IND)	6		6	5
	(EXT)	5		6	5
PSHA		4	(PSHSA)	5	4
MUL		—		11	6

(1)マイクロコンピュータの基礎と6800　CQ出版
(2)MC6809 Preliminary Progrming Manral Motorola Inc.
(3)HD6301V MCU 暫定仕様'81.5　日立製作所

注）ただしこの6301は日立のものでHC-20の6301と同じかどうかは未確認です．

出力例

```
2000 2C31 BGE  $2033      2010 00   FCB  $00
2002 3A   ABX             2011 2018 BRA  $202B
2003 B6   LDA A  $302C    2013 0B   SEV
     302C                 2014 B7   STA A  $3A8D
2006 31   INS                  3A8D
2007 3A   ABX             2017 00   FCB  $00
2008 823A SBC A  #$3A     2018 2036 BRA  $2050
200A 87   FCC  /・/       201A 0B   SEV
200B D031 SUB B  $31      201B B8   EOR A  $3A8D
200D 31   INS                  3A8D
200E 30   TSX             201E 3D   MUL
200F 30   TSX
```

逆アセンブラ プログラム・リスト

```
0 '******************
1 '* HC-20 - 6301V *
2 '* Dis-Assembler *
3 '*    ver.1.1    *
4 '*  1982.8.27.   *
```

I/Oプラザ　▶I/Oの82年5月号のMZのフルグラ・ボードを友達に作ってもらいました．[illegible]　(LOVE LIGHT)

398

I/O

逆アセンブラ プログラム・リスト

```
5 '* by オオムラ タダシ  *
6 '******************
10 GOTO1000:'---> Initialize
19 '
20 '===== Addressing Mode =====
30 D=PEEK(A):A=A+1:B$=B$+FNR$:O$=O$+" #$"+FNR$:RETURN:'<---Immediate---
35 D=PEEK(A):C$=FNR$:O$=C$+" #$"+FNR$
37 D=PEEK(A+1):A=A+2:C$=C$+FNR$:O$=O$+FNR$:RETURN:'<---16bit-Immediate---
40 D=PEEK(A):A=A+1:B$=B$+FNR$:O$=O$+" $"+FNR$:RETURN:'<---Direct---
50 D=PEEK(A):A=A+1:B$=B$+FNR$:O$=O$+" $"+FNR$+",X":RETURN:'<---Indexed---
60 D=PEEK(A):C$=FNR$:O$=O$+" $"+FNR$:D=PEEK(A+1):A=A+2
65 C$=C$+FNR$:O$=O$+FNR$:RETURN:'<---Extended---
70 O$="A":RETURN:'<---Acc.A---
75 O$="B":RETURN:'<---Acc.B---
80 IF D<32 THEN M$="FCB":O$=" $"+FNR$ ELSE M$="FCC":O$=" /"+CHR$(D)+"/"
85 RETURN:'<---Undefined OP code---
99 '
100 '===== Main-routine =====
110 A$=FNA$(STR$(A))+" ":O$="":C$=""
120 D=PEEK(A):A=A+1:B$=FNR$:U=D¥16:L=D MOD 16
130 IF U<4THEN GOSUB500 ELSE M=(U MOD4)+1:IF U<8 THEN GOSUB400 ELSE GOSUB300
149 '
150 '===== Print routine =====
160 ON P GOSUB180,200,230:Y$=INKEY$:IF Y$=CHR$(13)THEN280ELSEIF Y$<>" "THEN110
162 IF P>1THEN PRINT"Start address:";
164 Y$=INKEY$:IF Y$=""THEN164 ELSE IF Y$=" "THEN110 ELSE IF Y$=CHR$(13)THEN280
165 IF P=1 THENPRINT:PRINT"Start address: ";Y$;:ELSEPRINTY$;
166 A$=INKEY$:IF A$=""THEN166 ELSEIF A$<>CHR$(13)THEN Y$=Y$+A$:PRINTA$;:GOTO166
168 V=VAL("&H"+Y$):IF V>-1 AND V<&H4F THEN162 ELSE A=V:IF P=3 THEN GOSUB240
169 GOTO100
170 '--- to LCD ---
180 PRINT:PRINTA$;B$;C$;TAB(5);M$;" ";O$;:RETURN
190 '--- to PRINTER ---
200 LPRINT A$;B$;TAB(10);M$;" ";O$:IF C$<>""THENLPRINTTAB(5);C$
210 RETURN
220 '--- to COM0: ---
230 PRINT#3,A$;B$;C$;TAB(20);M$;" ";O$:G=G+1:IF G<60 THENRETURN
240 PRINT#3,CL$;"*6301V* DIS-ASSEMBLE LIST                    by T.OMURA     ";DATE$
250 PRINT#3,CHR$(10):G=0:RETURN
269 '
270 '===== E N D =====
280 PRINT:PRINT">END of Dis-assemble":END
289 '
290 ' ===== Mnemonics =====
300 '--- 2 オペランド ---
310 M$=N$(L,5)
315 ON L GOTO 320,320,340,320,320,320,350,320,320,320,220,360,370,380,390
320 IF U<12THEN O$="A " ELSE O$="B "
330 ON M GOTO 30,40,50,60
335 ON M GOTO 35,40,50,60
340 IF U>11 THEN M$="ADDD":GOTO335 ELSE 335
350 IF M=1 THEN80 ELSE320
360 IF U>11 THEN M$="LDD":GOTO335 ELSE335
370 IF U>11 THEN M$="STD":IF M=1 THEN80 ELSE330
375 IF M=1 THEN M$="BSR":GOTO520 ELSE330
380 IF U>11 THEN M$="LDX":GOTO335 ELSE335
390 IF U>11 THEN M$="STX"
395 IF M=1 THEN 80 ELSE330
400 '--- 1 オペランド ---
410 M$=N$(L,4):ONL GOTO430,430,420,420,430,420,420,410,420,420,430,420,420,460
420 ON M GOTO 70,75,50,60
430 IF M<3 THEN80 ELSE D=PEEK(A):A=A+1:C$=FNR$:O$="#$"+FNR$:'<---OIM,EIM,etc---
440 D=PEEK(A):A=A+1:C$=C$+FNR$:O$=O$+" $"+FNR$:IF M=3 THEN O$=O$+",X"
450 RETURN
470 '--- JMP ---
480 ON M GOTO80,80,50,60
500 '--- インプライド & ブランチ ---
510 M$=N$(L,U):IF M$=""THEN80 ELSE IF U<>2 THEN RETURN:'<---Implied---
520 D=PEEK(A):A=A+1:B$=B$+FNR$:O$=" $"+FNA$(STR$(A+D)):RETURN:'<---Branch---
999 '
1000 '===== Initialize =====
1010 WIDTH20,20:DIM N$(15,5):CL$=CHR$(12):PRINT" 6301V Dis-Assembler "
1020 DEF FNR$=RIGHT$("0"+HEX$(D),2):DEFFNA$(D$)=RIGHT$("000"+HEX$(VAL(D$)),4)
1030 FOR J=0TO5:FOR I=0TO15:READ N$(I,J):NEXT:NEXT
1050 PRINT"Output to 1.LCD";TAB(10);"2.Printer";TAB(10);"3.COM0:","Select 1-3";
1060 M$=INKEY$:IF M$=""THEN1060 ELSEP=VAL(M$):IF P<1 OR P>3 THENPRINT:GOTO1050
1070 IF P=3 THEN OPEN"O",3,"COM0:":GOSUB240
1100 PRINT:INPUT"Start address:";A$:A=VAL("&H"+A$):IF A>&H4E OR A<0 THEN100
1110 FOR J=1TO3:SOUND16,1:SOUND0,1:NEXT:GOTO1100
2999 '
3000 '===== ニーモニック・データ =====
3010 DATA ,NOP,,,LSRD,ASLD,TAP,TPA,INX,DEX,CLV,SEV,CLC,SEC,CLI,SEI
3020 DATA SBA,CBA,,,,,TAB,TBA,XGDX,DAA,SLP,ABA,,,,
3030 DATA BRA,BRN,BHI,BLS,BCC,BCS,BNE,BEQ,BVC,BVS,BPL,BMI,BGE,BLT,BGT,BLE
3040 DATA TSX,INS,PULA,PULB,DES,TXS,PSHA,PSHB,PULX,RTS,ABX,RTI,PSHX,MUL,WAI,SWI
3050 DATA NEG,AIM,OIM,COM,LSR,EIM,ROR,ASR,ASL,ROL,DEC,TIM,INC,TST,JMP,CLR
3060 DATA SUB,CMP,SBC,SUBD,AND,BIT,LDA,STA,EOR,ADC,ORA,ADD,CPX,JSR,LDS,STS
```

399

Figure 1:　Cover page, table of contents and article from I/O 7/1982.

Translation of the Japanese text

Disassembler

The HC-20 has a decent monitor and which can be used for writing machine language programs. However, the firmware in the ROM is still not publicly available. So you have to write your own input/output routines or analyze the ROM. Also, the HC-20's MCU is a 6301 (MCU: Microcomputer Unit), which is supposedly 6800 compatible but has some additional instructions, serial and parallel interfaces, and even a timer, so you can't write input/output routines without analyzing the ROM.

So, first I made a disassembler. Of course, the program was written in BASIC.

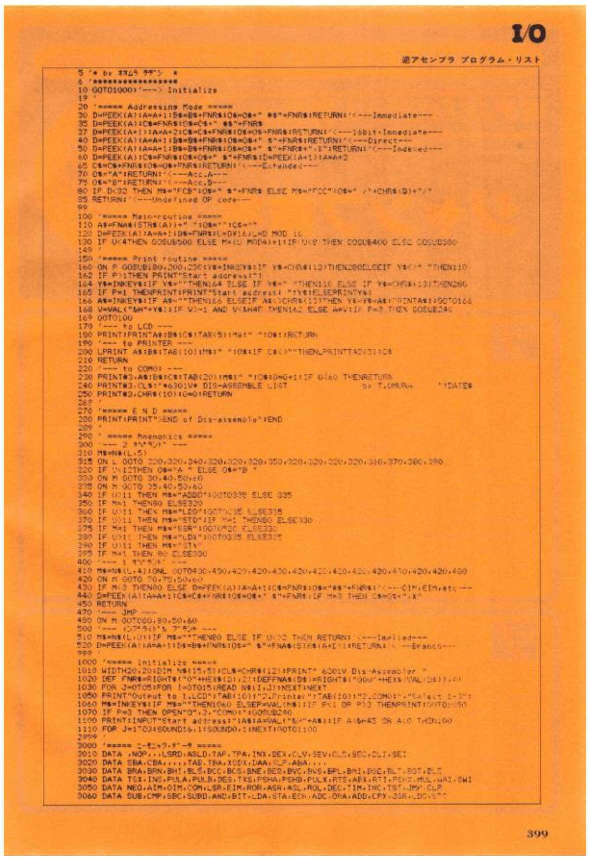
How to Use

LOAD the program or type it in exactly as listed. Then RUN it. It will display "6301V Dis-assembler" and then ask you to select the output device:

```
Output to
1. LCD
2. Printer
3. COM0:
Select 1-3
```

If you want to display on the LCD display, press 1; if you want to use the built-in printer, press 2; if you want to output to a printer connected to the RS-232C port, press 3.

It will ask for the first address to disassemble:

```
Start address: ?
```


Enter it in hexadecimal and disassembly will begin. To stop the disassembly, press the SPACE bar. Press the SPACE bar again to resume disassembly. If you want to start from a different address, stop disassembly with the SPACE bar, then enter the new start address in hexadecimal.

To stop program execution, press the RETURN key, you can either press it while disassembling or immediately after pausing the display with the SPACE bar.

```
End of Dis-assemble
```

So everyone, let's do our best to analyze the ROM.

Table 1: Instruction execution cycles for the 6800, 6809 and 6301

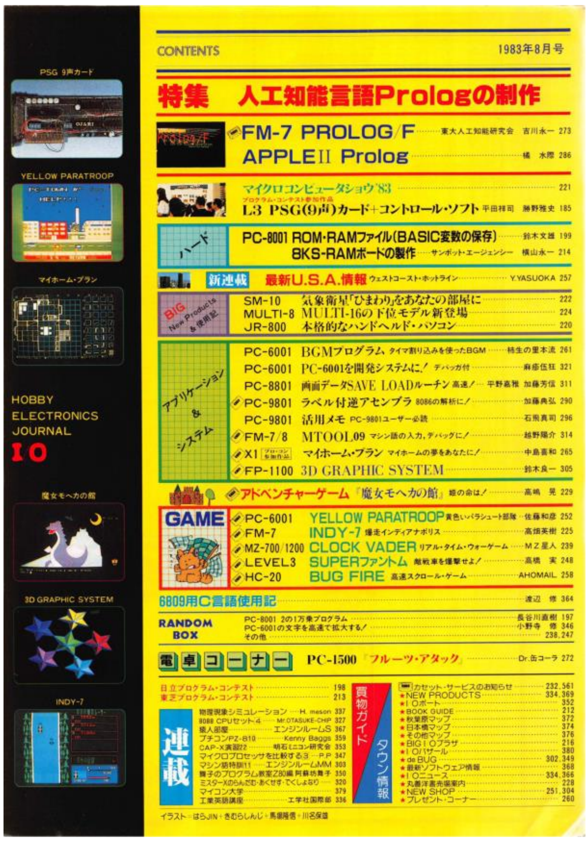
Instruction	6800	6809	6301
BEQ	4	3	3
BSR	8	7	6
JSR (Ext)	9	8	6
JSR (Ind)	8	8	5
LDAA (Imm)	2	2	2
LDAA (Dir)	3	4	3
LDAA (Ind)	5	5	4
LDAA (Ext)	4	5	4
ROLA	2	2	1
ROL (Ind)	7	7	6
ROL (Ext)	6	7	6
TSTA	2	2	1
TST (Ind)	7	7	4
TST (Ext)	6	7	4
LDX (Imm)	3	3	3
LDX (Dir)	4	5	4
LDX (Ind)	6	6	5
LDX (Ext)	5	6	5
PSHA	4	5	4
MUL	-	11	6

```
0 '*****************
1 '* HC-20 - 6301V *
2 '* Dis-Assembler *
3 '*    ver.1.1    *
4 '*   1982.8.27.  *
5 '* by ????????   *
6 '*****************
10 GOTO1000:'---> Initialize
19 '
20 '===== Addressing Mode =====
30 D=PEEK(A):A=A+1:B$=B$+FNR$:O$=O$+" #$"+FNR$:RETURN:'<---Immediate---
35 D=PEEK(A):C$=FNR$:O$=O$+" #$"+FNR$
37 D=PEEK(A+1)*A=A+2:C$=C$+FNR$:O$=O$+FNR$:RETURN:'<---16bit-Immedaite---
40 D=PEEK(A):A=A+1:B$=B$+FNR$:O$=O$+" $"+FNR$:RETURN:'<---Direct---
50 D=PEEK(A):A=A+1:B$=B$+FNR$:O$=O$+" $"+FNR$+",X":RETURN:'<---Indexed---
60 D=PEEK(A):C$=FNR$:O$=O$+" $"+FNR$:D=PEEK(A+1):A=A+2
65 C$=C$+FNR$:O$=O$+FNR$:RETURN:'<---Extended---
70 O$="A":RETURN:'<---Acc.A---
75 O$="B":RETURN:'<---Acc.B---
80 IF D<32 THEN M$=FCB":O$=" $"+FNR$ ELSE M$="FCC":C$=" /"+CHR$(D)+"/"
85 RETURN:'<---Undefined OP code---
99
100 '===== Main-routine =====
110 A$=FNA$(STR$(A))+" ":O$="":C$=""
120 D=PEEK(A):A=A+1:B$=FNR$:U=d\16:L=D MOD 16
```

```
130 IF U<4 THEN GOSUB 500 ELSE M=(U MOD 4)+1:IF U<8 THEN GOSUB 400 ELSE GOSUB 300
149 '
150 '===== Print routine =====
160 ON P GOSUB 180,200:Y$=INKEY$:IF Y$=CHR$(13) THEN 280 ELSE IF Y$<>" " THEN 110
162 IF P>1 THEN PRINT "Start address:";
Y$=INKEY$:IFY$="" THEN 164 ELSE IF Y$=" " THEN 110 ELSE IF Y$=CHR$(13) THEN 280
165 IF P=1 THEN PRINT:PRINT"Start address: ";Y$;ELSE PRINT Y$;
166 A$=INKEY$:IF A$="" THEN 166 ELSE IF A$<>CHR$(13) THEN Y$=Y$+A$:PRINT A$;:GOTO 166
168 V=VAL("&H"+Y$):IF V>-1 AND V<&H4F THEN 162 ELSE A=V:IF P=3 THEN GOSUB 240
169 GOTO 100
170 '--- to LCD ---
180 PRINT:PRINT A$;B$;C$;TAB(5);M$;" ";O$;:RETURN
190 '--- to PRINTER ---
200 LPRINT A$;B$;TAB(10);M$;" ";O$;:IF C$<>"" THEN LPRINT TAB(5);C$
210 RETURN
220 '--- to COM0: ---
230 PRINT#3,A$;B$;C$;TAB(20);M$;" ";O$:G=G+1:IF G<60 THEN RETURN
240 PRINT#3,CL$;"*6301V DIS-ASSEMBLE LIST              by T.OMURA     ";DATE$
250 PRINT#3,CHR$(10):G=0:RETURN
269 '
270 '===== E N D =====
280 PRINT:PRINT">END of Dis-assemble":END
289 '
290 '===== Mnemonics =====
300 '--- 2 bytes ---
310 M$=N$(L,5)
315 ON L GOTO 320,320,340,320,320,320,350,320,320,320,320,360,370,380,390
320 IF U<12 THEN O$="A " ELSE O$="B "
330 ON M GOTO 30,40,50,60
335 ON M GOTO 35,40,50,60
340 IF U>11 THEN M$="ADDD":GOTO 335 ELSE 335
350 IF M=1 THEN 80 ELSE 320
360 IF U>11 THEN M$="LDD":GOTO 335 ELSE 335
370 IF U>11 THEN M$="STD":IF M=1 THEN 80 ELSE 330
375 IF M=1 THEN M$="BSR":GOTO 520 ELSE 330
330 IF U>11 THEN M$="LDX":GOTO 335 ELSE 335
390 IF U>11 THEN M$="STX"
395 IF M=1 THEN 80 ELSE 330
400 '--- 1 byte ---
410 M$=N$(L,4):ON L GOTO 430,430,420,420,430,420,420,420,420,420,430,420,420,480
420 IN M GOTO 70,75,50,60
430 IF M<3 THEN 80 ELSE D=PEEK(A):A=A+1:C$=FNR$:O$="#$"+FNR$:'<---OIM,EIM,etc---
440 D)PEEK(A):A=A+1:C$=C$+FNR$:O$=O$+" $"+FNR$:IF M=3 THEN O$=O$+",X"
450 RETURN
470 '--- JMP ---
490 ON M GOTO 80,80,50,60
500 '--- ---
510 M$=N$(L,U):IF M$="" THEN 80 ELSE IF U<>2 THEN RETURN:'<---Implied---
520 D=PEEK(A):A=A+1:B$=B$+FNR$:O$=" $"+FNA$(STR$(A+D)):RETURN:'<---Branch---
999 '
1000 '===== Initialize =====
1010 WIDTH 20,20:DIM N$(15,5):CL$=CHR$(12):PRINT" 6301V Dis-Assembler "
1020 DEF FNR$=RIGHT$("0"+HEX$(D),2):DEF FNA$(D$)=RIGHT$("000"+HEX$(VAL(D$)),4)
1030 FOR J=0 TO 5:FOR I=0 TO 15:READ N$(I,J):NEXT:NEXT
1050 PRINT"Output to 1,LCD";TAB(10);"2.Printer";TAB(10);"3.COM0:","Select 1-3";
1060 M$=INKEY$:IG M$="" THEN 1060 ELSE P=VAL(M$):IF P<1 OR P>3 THEN PRINT:GOTO 1050
1070 IF P=3 THEN OPEN"O",3,"COM0:":GOSUB 240
1100 PRINT:INPUT"Start address:";A$:A=VAL("&H"+A$):IF A>&H48 OR A<0 THEN 100
1110 FOR J=1 TO 3:SOUND 16,1:SOUND 0,1:NEXT:GOTO 1100
2999 '
3000 '===== =====
3010 DATA ,NOP,,,LSRD,ASLD,TAP,TPA,INX,DEX,CLV,SEV,CLC,SEC,CLI,SEI
3020 DATA SBA,CBA,,,,,TAB,TBA,XGDX,DAA,SLP,ABA,,,,
3030 DATA BRA,BRN,BHI,BLS,BCC,BCS,BNE,BEQ,BVC,BVS,BPL,BMI,BGE,BLT,BGT,BLE
3040 DATA TSX,INS,PULA,PULB,DES,TXS,PSHA,PSHB,PULX,RTS,ABX,RTI,PSHX,MUL,WAI,SWI
3050 DATA NEG;AIM,OIM,COM,LSR,EIM,ROR,ASR,ASL,ROL,DEC,TIM,INC,TST,JMP,CLR
3060 DATA SUB,CMP,SBC,SUBD,AND,BIT,LDA,STA,EOR,ADC,ORA,ADD,CPX,JSR,LDD,STS
```

Bug Fire Game

The August 1983 issue (V8N8) contained a BASIC listing of the HC-20 adaptation of a game “BUG FIRE!”, by the author AHOMAIL. The original program for the PC-8001 had been published in previous issues of I/O magazine 3/1981 and 1/1983. A large part of this program was written in machine code, requiring a lengthy HEX dump to be entered by the reader.


PSG 9声カード
YELLOW PARATROOP
マイホーム・プラン
HOBBY ELECTRONICS JOURNAL IO
魔女モヘカの館
3D GRAPHIC SYSTEM
INDY-7

イラスト=はらJIN+きむらしんじ+馬場隆信+川名保雄

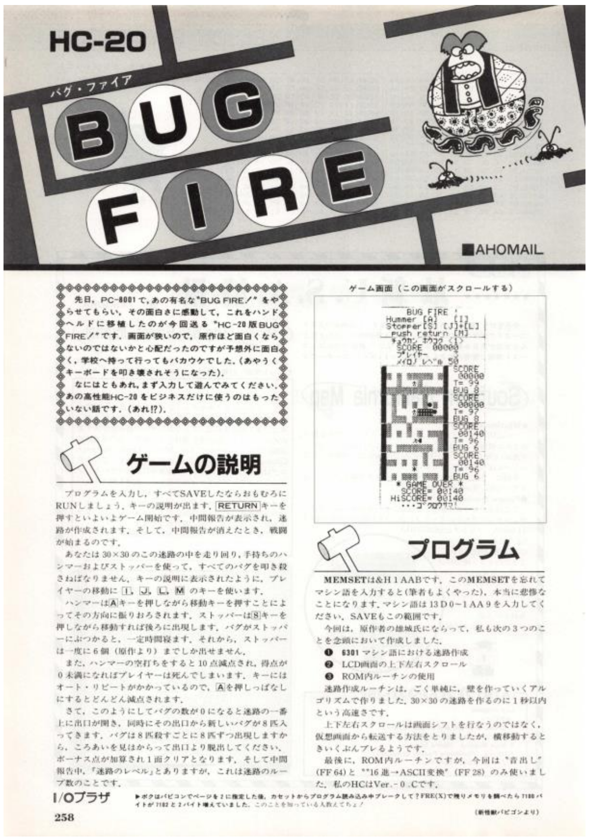
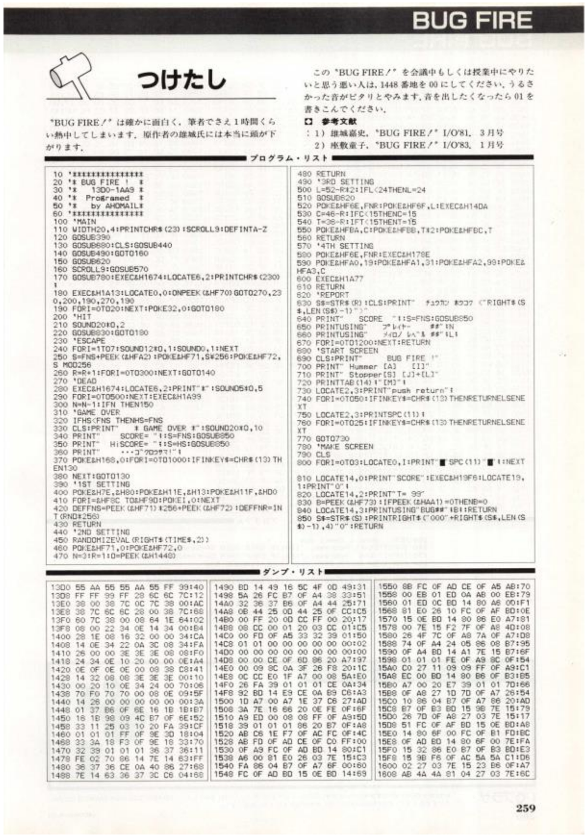
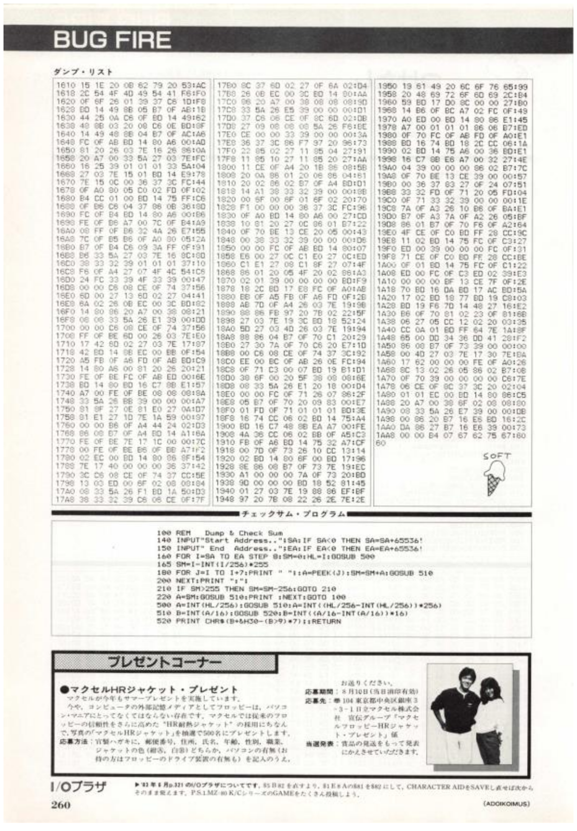




Figure 2: Cover page, table of contents and article from I/O 7/1982.

Translation of the Japanese text:

BUGFIRE!

by AHOMAIL

I was asked to play the famous BUGFIRE! on the PC-8001 the other day. I was impressed by the fun, which was transplanted to the handheld and implemented in the HC-20 version.

I was worried that the screen would not be as interesting as the original because it was small, but it was unexpectedly interesting, and it was a laughing stock when I took it to school (I almost broke the keyboard).

Anyway, please try playing by entering it first. It is a waste to use the high-performance HC-20 only for business.

Game Description

Enter the program. If you have saved everything, let's RUN it: a description of the keys will appear. Press the return key to start the game. A mid-term report is presented, then a maze is created. Afterwards, when the interim report disappeared, the battle begins.

You run around this 30 x 30 maze, using your man with a hammer and stoppers. We have to kill all the bugs. Use the four keys [J]-[L] left-right and [I]-[M] up-down to move the player.

The hammer will be swung in the direction of motion by pressing a movement key while holding the [A] key.

The stopper will appear behind you if you move while pressing the [S] key. When the bug hits the first stopper, it sleeps for a certain amount of time. You can only use six stoppers at once (like in the original).

Also, if you miss with the hammer, you lose 10 points. If the score is less than 0, the player dies. Because the key has auto-repeat, if you keep hammering the bug, you will get fewer points.

Finally, when the number of bugs is zero, the exit at the top of the maze opens. At the same time, eight new bugs come out of the exit. Because the bugs appear in groups of eight, kill eight of them to escape through the exit. Bonus points will be added and one level will be cleared.

In the interim report, it says "Maze Level". This is the number of loops in the maze.

MEMSET is `&H1AAB`. If you forget this MEMSET and enter machine language (I often do it), it really gets ugly. The machine language routines are from `&H13D0` to `&H1AA6`. SAVEM is also in this range.

I followed the original author, Mr. Oshiro, but I also created the following three things myself:

- 6301 Maze creation in machine language,
- up, down, left, right scrolling on the LCD screen,
- use of routines in ROM.

For creating a maze, I used a very simple algorithm that creates the walls. The high speed routine takes less than 1 second to create a 30×30 maze.

The left, right, up and down scrolling is not done by shifting the screen, but by transferring it from the virtual screen. It seems to be a little sluggish when moving horizontally.

Finally, the ROM routines: this time I used only “Sound output” (`&HFF64`) and “hexadecimal to ASCII conversion” (`&HFF28`).

"BUGFIRE" is certainly interesting. Even I get excited about it for about an hour. I really respect the author of the original work.

You can play this "BUGFIRE" during a meeting or during a class, but silently. Please set the value at `&H1448` to `0` to mute the sound of the bell. If you want to make a sound, please write a `1`.

References

1) Yuji Kasai, BUGFIRE!, I/O magazine, 3/1981.

2) Zashiki Tojo, BUGFIRE!, I/O magazine, 1/1983.

```
10 ' MH: Typed in from I/O Magazine
10 ' MH: Requires 32 KB RAM in HX-20
10 ' MH: Could be slightly compacted by removing space characters
10 ' ***************
20 ' * BUG FIRE !  *
30 ' * 13D0-1AA6   *
40 ' * Programed   *
50 ' * by AHOMAIL  *
60 ' ***************
100 ' MAIN
105 MEMSET &H1AA7
106 LOADM "PAC0:BUGFIRE.LOD"
107 REM GOSUB 1110
110 WIDTH 20,4 : PRINT CHR$(23) : SCROLL 9 : DEFINT A-Z
120 GOSUB 390
130 GOSUB 680 : CLS : GOSUB 440
140 GOSUB 490 : GOTO 160
150 GOSUB 620
160 SCROLL 9 : GOSUB 570
170 GOSUB 780 : EXEC &H1674 : LOCATE 6,2 : PRINT CHR$(230);
180 EXEC &H1A13 : LOCATE 0,0 : ON PEEK(&HF70) GOTO 270,230,200,190,270,190
190 FOR I=0 TO 20 : NEXT : POKE 32,0 : GOTO 180
200 ' HIT
210 SOUND 20*O,2
220 GOSUB 830 : GOTO 180
230 ' ESCAPE
240 FOR I=1 TO 7 : SOUND 12*O,1 : SOUND 0,1 : NEXT
250 S=FNS+PEEK(&HFA2) : POKE &HF71,S\256 : POKE &HF72,S MOD 256
260 R=R+1 : FOR I=0 TO 300 : NEXT : GOTO 140
270 ' DEAD
280 EXEC &H1674 : LOCATE 6,2 : PRINT "*" : SOUND 5*O,5
290 FOR I=0 TO 500 : NEXT : EXEC &H1A99
300 N=N-1 : IF N THEN 150
310 ' GAME OVER
320 IF HS<FNS THEN HS=FNS
330 CLS : PRINT "    * GAME OVER *" : SOUND 20*O,10
340 PRINT "    SCORE= "; : S=FNS : GOSUB 850
350 PRINT "  HiSCORE= "; : S=HS : GOSUB 850
360 PRINT "    Gokurosama!";
370 POKE &H168,0 : FOR I=0 TO 1000 : IF INKEY$=CHR$(13) THEN 130
380 NEXT : GOTO 130
390 ' 1ST SETTING
400 POKE &H7E,&H80 : POKE &H11E,&H13 : POKE &H11F,&HD0
410 FOR I=&HF8C TO &HF9D : POKE I,0 : NEXT
420 DEF FNS=PEEK(&HF71)*256+PEEK(&HF72) : DEF FNR=INT(RND*256)
430 RETURN
440 ' 2ND SETTING
450 RANDOMIZE VAL(RIGHT$(TIME$,2))
460 POKE &HF71,0 : POKE &HF72,0
470 N=3 : R=1 : O=PEEK(&H1448)
480 RETURN
490 ' 3RD SETTING
500 L=52-R*2 : IF L<24 THEN L=24
510 GOSUB 620
520 POKE &HF6E,FNR : POKE &HF6F,L : EXEC &H14DA
530 C=46-R : IF C<15 THEN C=15
540 T=36-R : IF T<15 THEN T=15
550 POKE &HFBA,C : POKE &HFBB,T*2 : POKE &HFBC,T
560 RETURN
```

```
570 ' 4TH SETTING
580 POKE &HF6E,FNR : EXEC &H178E
590 POKE &HFA0,19 : POKE &HFA1,31 : POKE &HFA2,99 : POKE &HFA3,C
600 EXEC &H1A77
610 RETURN
620 ' REPORT
630 S$=STR$(R) : CLS : PRINT "     CONTROL <"RIGHT$(S$,LEN(S$)-1)">"
640 PRINT "   SCORE  "; : S=FNS : GOSUB 850
650 PRINT USING "      PLAYER ##";N
660 PRINT USING "       LEVEL ##";L;
670 FOR I=0 TO 1200 : NEXT : RETURN
680 ' START SCREEN
690 CLS : PRINT"     BUG FIRE !"
700 PRINT " Hummer [A]   [I]"
710 PRINT " Stopper[S] [J]+[L]"
720 PRINT TAB(14);"[M]";
730 LOCATE 2,3 : PRINT "press return";
740 FOR I=0 TO 50 : IF INKEY$=CHR$(13) THEN RETURN ELSE NEXT
750 LOCATE 2,3 : PRINT SPC(11);
760 FOR I=0 TO 25 : IF INKEY$=CHR$(13) THEN RETURN ELSE NEXT
770 GOTO 730
780 ' MAKE SCREEN
790 CLS
800 FOR I=0 TO 3 : LOCATE 0,I : PRINT "#"SPC(11)"#" : NEXT
810 LOCATE 14,0 : PRINT "SCORE" : EXEC &H19F6 : LOCATE 18,1 : PRINT "0";
820 LOCATE 14,2 : PRINT "T= 99"
830 B=PEEK(&HF73) : IF PEEK(&HAA1)=0 THEN B=0
840 LOCATE 14,3 : PRINT USING "BUG##";B; : RETURN
850 S$=STR$(S) : PRINT RIGHT$("000"+RIGHT$(S$,LEN(S$)-1),4)"0" : RETURN
1050 REM --- Epson HX-20      ---
1060 REM --- M. Hepperle 2024 ---
1070 REM --- adjust BASIC starting address
1080 REM --- load the code bytes
1090 REM --- Hex Code Loader ---
1110 N%=0
1120 READ C$
1130 IF C$="DONE" THEN 1180
1140 N%=N%+1 : IF N%=16 THEN PRINT HEX$(A%);CHR$(1); : N%=0
1150 C%=VAL("&H"+C$)
1160 IF LEN(C$)=4 THEN A%=C% : GOTO 1120
1170 POKE A%,C% : A%=A%+1 : GOTO 1120
1180 PRINT
1190 REM SAVEM "CAS0:BUGFIRE.LOD",&H13D0,&H1AA6
1195 RETURN
1200 DATA 13D0,55,AA,55,55,AA,55,FF,99
1201 DATA FF,FF,99,FF,28,6C,6C,7C
1202 DATA 38,00,38,7C,0C,7C,38,00
1203 DATA 38,7C,6C,6C,28,00,38,7C
1204 DATA 60,7C,38,00,08,64,1E,64
1205 DATA 08,00,22,34,0E,14,34,00
1206 DATA 28,1E,08,16,32,00,00,34
1207 DATA 14,0E,34,22,0A,3C,08,34
1208 DATA 26,00,00,3E,3E,3E,08,08
1209 DATA 24,34,0E,10,20,00,00,0E
1210 DATA 0E,0F,0E,0E,00,08,38,C8
1211 DATA 14,32,08,08,3E,3E,3E,00
1212 DATA 00,20,10,0E,34,24,00,70
1213 DATA 70,F0,70,70,00,08,0E,09
1214 DATA 14,26,00,00,00,00,00,00
1215 DATA 01,37,B6,0F,6E,16,1B,1B
1216 DATA 16,1B,98,09,4C,B7,0F,6E
1217 DATA 33,11,25,03,10,20,FA,39
1218 DATA 01,01,01,FF,0F,9E,3D,18
1219 DATA 33,3A,18,F3,0F,9E,18,33
1220 DATA 32,39,01,01,01,36,37,36
1221 DATA FE,02,70,86,14,7E,14,63
1222 DATA 36,37,36,CE,0A,40,86,27
1223 DATA 7E,14,63,36,37,3C,C6,04
1224 DATA BD,14,49,16,5C,4F,0D,49
1225 DATA 5A,26,FC,B7,0F,A4,38,33
1226 DATA 32,36,37,B6,0F,A4,44,25
1227 DATA 0B,44,25,0D,44,25,0F,CC
1228 DATA 00,FF,20,0D,CC,FF,00,20
1229 DATA 08,CC,00,01,20,03,CC,01
1230 DATA 00,FD,0F,A5,33,32,39,01
1231 DATA 01,01,00,00,00,00,00,00
```

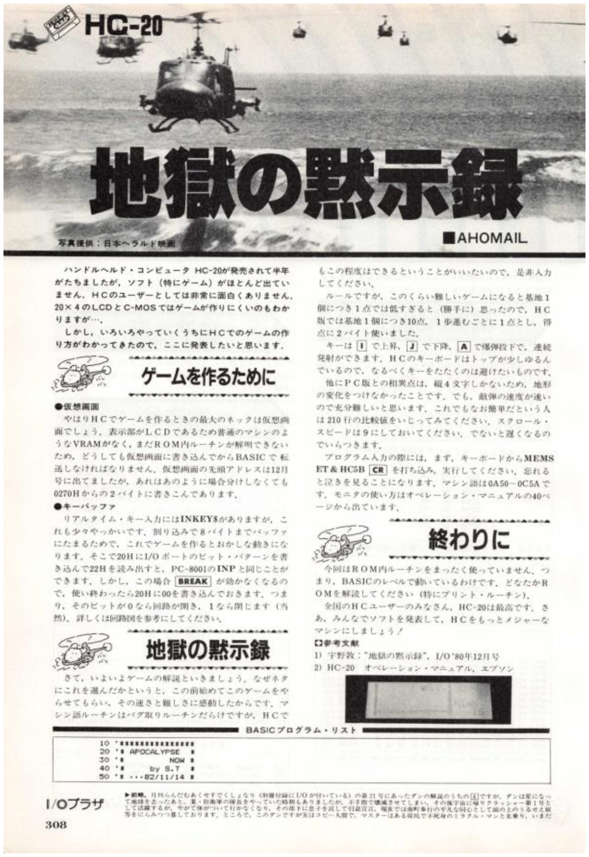
```
1232 DATA 00,00,00,00,00,00,00,00
1233 DATA 00,00,CE,0F,6D,86,20,A7
1234 DATA 00,09,8C,0A,3F,26,F8,20
1235 DATA 0C,CC,E0,1F,A7,00,08,5A
1236 DATA 26,FA,39,01,01,01,CE,0A
1237 DATA 92,BD,14,E9,CE,0A,B9,C6
1238 DATA 1D,A7,00,A7,1E,37,C6,27
1239 DATA 3A,7E,16,66,20,0E,FE,0F
1240 DATA A9,ED,00,08,08,FF,0F,A9
1241 DATA 39,01,01,01,86,20,B7,0F
1242 DATA AB,C6,1E,F7,0F,AC,FC,0F
1243 DATA AB,FD,0F,AD,CE,0F,C0,FF
1244 DATA 0F,A9,FC,0F,AD,BD,14,80
1245 DATA A6,00,81,E0,26,03,7E,15
1246 DATA FA,86,04,B7,0F,A7,6F,00
1247 DATA FC,0F,AD,BD,15,0E,BD,14
1248 DATA 8B,FC,0F,AD,CE,0F,A5,AB
1249 DATA 00,EB,01,ED,0A,AB,00,EB
1250 DATA 01,ED,0C,BD,14,80,A6,00
1251 DATA 81,E0,26,10,FC,0F,AF,BD
1252 DATA 15,0E,BD,14,80,86,E0,A7
1253 DATA 00,7E,15,F2,7F,0F,A8,4D
1254 DATA 26,4F,7C,0F,A8,7A,0F,A7
1255 DATA 74,0F,A4,24,05,86,08,B7
1256 DATA 0F,A4,BD,14,A1,7E,15,B7
1257 DATA 01,01,01,FE,0F,A9,8C,0F
1258 DATA C0,27,11,09,09,FF,0F,A9
1259 DATA EC,00,BD,14,80,B6,0F,B3
1260 DATA A7,00,20,E7,39,01,01,7D
1261 DATA 0F,A8,27,1D,7D,0F,A7,26
1262 DATA 10,86,04,B7,0F,A7,86,20
1263 DATA B7,0F,B3,BD,15,9B,7E,15
1264 DATA 26,7D,0F,A8,27,03,7E,15
1265 DATA 51,FC,0F,AF,BD,15,0E,BD
1266 DATA 14,80,6F,00,FC,0F,B1,FD
1267 DATA 0F,AD,BD,14,80,6F,00,7E
1268 DATA 15,32,86,E0,B7,0F,B3,BD
1269 DATA 15,9B,F6,0F,AC,5A,5A,C1
1270 DATA 02,27,03,7E,15,23,B6,0F
1271 DATA AB,4A,4A,81,04,27,03,7E
1272 DATA 15,1E,20,0B,62,79,20,53
1273 DATA 2C,54,4F,4D,49,54,41,F6
1274 DATA 0F,6F,26,01,39,37,C6,1D
1275 DATA BD,14,49,8B,05,B7,0F,AB
1276 DATA 44,25,0A,C6,0F,BD,14,49
1277 DATA 48,8B,03,20,08,C6,0E,BD
1278 DATA 14,49,48,8B,04,B7,0F,AC
1279 DATA FC,0F,AB,BD,14,80,A6,00
1280 DATA 81,20,26,03,7E,16,26,86
1281 DATA 20,A7,00,33,5A,27,03,7E
1282 DATA 16,25,39,01,01,01,33,5A
1283 DATA 27,03,7E,15,01,BD,14,E9
1284 DATA 7E,15,0C,00,36,37,3C,FC
1285 DATA 0F,A0,80,05,C0,02,FD,0F
1286 DATA B4,CC,01,00,BD,14,75,FF
1287 DATA 0F,B6,C6,04,37,86,0B,36
1288 DATA FC,0F,B4,BD,14,80,A6,00
1289 DATA FE,0F,B6,A7,00,7C,0F,B4
1290 DATA 08,FF,0F,B6,32,4A,26,E7
1291 DATA 7C,0F,B5,B6,0F,A0,80,05
1292 DATA B7,0F,B4,C6,09,3A,FF,0F
1293 DATA B6,33,5A,27,03,7E,16,8C
1294 DATA 38,33,32,39,01,01,01,37
1295 DATA F6,0F,A4,27,07,4F,4C,54
1296 DATA 24,FC,33,39,4F,33,39,00
1297 DATA 00,00,C6,08,CE,0F,74,37
1298 DATA 6D,00,27,13,6D,02,27,04
1299 DATA 6A,02,26,0B,EC,00,3C,BD
1300 DATA 14,80,86,20,A7,00,38,08
1301 DATA 08,08,33,5A,26,E1,39,00
1302 DATA 00,00,C6,08,CE,0F,74,37
1303 DATA FF,0F,BE,6D,00,26,03,7E
1304 DATA 17,42,6D,02,27,03,7E,17
1305 DATA 42,BD,14,8B,EC,00,BB,0F
1306 DATA A5,FB,0F,A6,FD,0F,AB,BD
1307 DATA 14,80,A6,00,81,20,26,20
```

```
1308 DATA FE,0F,BE,FC,0F,AB,ED,00
1309 DATA BD,14,80,BD,16,C7,8B,E1
1310 DATA A7,00,FE,0F,BE,08,08,08
1311 DATA 33,5A,26,BB,39,00,00,00
1312 DATA 81,8F,27,0E,81,E0,27,0A
1313 DATA 81,E1,27,1D,7E,1A,59,00
1314 DATA 00,00,B6,0F,A4,44,24,02
1315 DATA 86,08,B7,0F,A4,BD,14,A1
1316 DATA FE,0F,BE,7E,17,1C,00,00
1317 DATA 00,FE,0F,BE,B6,0F,BB,A7
1318 DATA 02,EC,00,BD,14,80,86,8F
1319 DATA 7E,17,40,00,00,00,36,37
1320 DATA 3C,C6,08,CE,0F,74,37,CC
1321 DATA 13,03,ED,00,6F,02,08,08
1322 DATA 08,33,5A,26,F1,BD,1A,50
1323 DATA 38,33,32,39,C6,06,CE,0F
1324 DATA 8C,37,6D,02,27,0F,6A,02
1325 DATA 26,0B,EC,00,3C,BD,14,80
1326 DATA 86,20,A7,00,38,08,08,08
1327 DATA 33,5A,26,E5,39,00,00,00
1328 DATA 37,C6,06,CE,0F,8C,6D,02
1329 DATA 27,09,08,08,08,5A,26,F6
1330 DATA CE,00,00,33,39,00,00,00
1331 DATA 36,37,3C,86,F7,97,20,96
1332 DATA 22,85,02,27,11,85,04,27
1333 DATA 11,85,10,27,11,85,20,27
1334 DATA 11,CE,0F,A4,20,1B,86,08
1335 DATA 20,0A,86,01,20,06,86,04
1336 DATA 20,02,86,02,B7,0F,A4,BD
1337 DATA 14,A1,38,33,32,39,00,00
1338 DATA 00,6F,00,6F,01,6F,02,20
1339 DATA F1,00,00,00,36,37,3C,FC
1340 DATA 0F,A0,BD,14,80,A6,00,27
1341 DATA 10,81,20,27,0C,86,01,B7
1342 DATA 0F,70,BE,13,CE,20,05,00
1343 DATA 00,38,33,32,39,00,00,00
1344 DATA 00,00,FC,0F,AB,BD,14,80
1345 DATA E6,00,27,0C,C1,E0,27,0C
1346 DATA C1,E1,27,08,C1,8F,27,07
1347 DATA 86,01,20,05,4F,20,02,86
1348 DATA 02,01,39,00,00,00,00,BD
1349 DATA 18,2C,BD,17,E8,FC,0F,A0
1350 DATA BB,0F,A5,FB,0F,A6,FD,0F
1351 DATA AB,7D,0F,A4,26,03,7E,19
1352 DATA 88,86,FB,97,20,7B,02,22
1353 DATA 27,03,7E,19,3C,BD,18,52
1354 DATA 5D,27,03,4D,26,03,7E,19
1355 DATA 88,86,04,B7,0F,70,C1,20
1356 DATA 27,30,7A,0F,70,C6,20,E7
1357 DATA 00,C6,08,CE,0F,74,37,3C
1358 DATA EE,00,BC,0F,AB,26,0E,FC
1359 DATA 0F,71,C3,00,07,BD,19,B1
1360 DATA 38,6F,00,20,5F,38,08,08
1361 DATA 08,33,5A,26,E1,20,18,00
1362 DATA 00,00,FC,0F,71,26,07,86
1363 DATA 05,B7,0F,70,20,09,83,00
1364 DATA 01,FD,0F,71,01,01,01,BD
1365 DATA 16,74,CC,06,02,BD,14,75
1366 DATA BD,16,C7,48,8B,EA,A7,00
1367 DATA 4A,36,CC,06,02,BB,0F,A5
1368 DATA FB,0F,A6,BD,14,75,32,A7
1369 DATA 00,7D,0F,73,26,10,CC,13
1370 DATA 02,BD,14,80,6F,00,BD,17
1371 DATA 8E,86,08,B7,0F,73,7E,19
1372 DATA A1,00,00,00,7A,0F,73,20
1373 DATA 9D,00,00,00,BD,18,52,81
1374 DATA 01,27,03,7E,19,88,86,EF
1375 DATA 97,20,7B,08,22,26,2E,7E
1376 DATA 19,61,49,20,6C,6F,76,65
1377 DATA 20,48,69,72,6F,6D,69,2C
1378 DATA 59,BD,17,D0,8C,00,00,27
1379 DATA 14,B6,0F,BC,A7,02,FC,0F
1380 DATA A0,ED,00,BD,14,80,86,E1
1381 DATA A7,00,01,01,01,86,06,B7
1382 DATA 0F,70,FC,0F,AB,FD,0F,A0
1383 DATA BD,16,74,BD,18,2C,CC,06
```

```
1384 DATA 02,BD,14,75,A6,00,36,BD
1385 DATA 16,C7,8B,E6,A7,00,32,27
1386 DATA 04,39,00,00,00,86,02,B7
1387 DATA 0F,70,BE,13,CE,39,00,00
1388 DATA 00,36,37,83,27,0F,24,07
1389 DATA 33,32,FD,0F,71,20,05,FD
1390 DATA 0F,71,33,32,39,00,00,00
1391 DATA 7A,0F,A3,26,10,B6,0F,BA
1392 DATA B7,0F,A3,7A,0F,A2,26,05
1393 DATA 86,01,B7,0F,70,F6,0F,A2
1394 DATA 4F,CE,0F,C0,BD,FF,28,CC
1395 DATA 11,02,BD,14,75,FC,0F,C3
1396 DATA ED,00,39,00,00,00,FC,0F
1397 DATA 71,CE,0F,C0,BD,FF,28,CC
1398 DATA 0E,01,BD,14,75,FC,0F,C1
1399 DATA ED,00,FC,0F,C3,ED,02,39
1400 DATA 00,00,00,BF,13,CE,7F,0F
1401 DATA 70,BD,16,DA,BD,17,AC,BD
1402 DATA 17,02,BD,18,77,BD,19,C8
1403 DATA BD,19,F6,7D,14,48,27,16
1404 DATA B6,0F,70,81,02,23,0F,81
1405 DATA 06,27,05,CC,12,02,20,03
1406 DATA CC,0A,01,BD,FF,64,7E,1A
1407 DATA 65,00,DD,34,36,DD,41,28
1408 DATA 86,08,B7,0F,73,39,00,00
1409 DATA 00,4D,27,03,7E,17,30,7E
1410 DATA 17,62,00,00,00,FE,0F,A0
1411 DATA 8C,13,02,26,05,86,02,B7
1412 DATA 0F,70,39,00,00,00,00,C6
1413 DATA 06,CE,0F,8C,37,3C,20,02
1414 DATA 01,01,EC,00,BD,14,80,86
1415 DATA 20,A7,00,38,6F,02,08,08
1416 DATA 08,33,5A,26,E7,39,00,00
1417 DATA 00,86,20,B7,16,E6,BD,16
1418 DATA DA,86,27,B7,16,E6,39
1419 DATA DONE
```

Apocalypse Now Game

The 3/1983 issue of I/O (V8N3) contained a listing of another game by AHOMAIL for the HC-20. The BASIC code was augmented by a short machine code part.



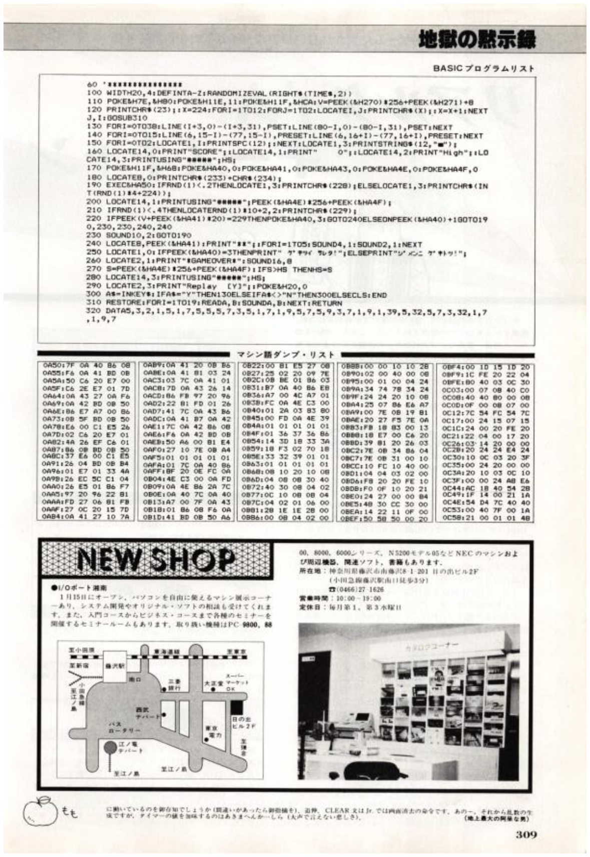


Figure 3: Cover page, table of contents and article from I/O 3/1983.

Translation of the Japanese Text

Apocalypse of Hell

AHOMAIL

Half a year has passed since the release of the HC-20 handheld computer, but there are hardly any software (especially games) released. As an HC user, I find it very uninteresting. I understand that it is difficult to make games with a 20 x 4 LCD and CMOS.

However, as I continued to work on it, I began to understand how to make games in HC, so I would like to announce it here.

Virtual Screen

The biggest bottleneck when making a game on the HC is the virtual screen.

Because the display is an LCD, there is no VRAM like in normal machines, and the routines in the ROM have not yet been clarified, so you have to write to the virtual screen and then transfer it with BASIC. The top address of the virtual screen was in the December issue, but there is no need to distinguish between cases like that; it's just written to the two bytes starting from $0270.

Key Buffer

There is INKEY$ for real-time key input, bit this is also a bit troublesome. The buffer stores up to 8 bytes when an interrupt occurs, so if you create a game using this, it will behave strangely. So if you write the I/O port bit pattern to $20 and read $22, you can achieve the same thing as INP on the PC-8001. However, in this case, BREAK will not work, so when you're done, write $00 to $20. In other words, if the bit is 0, the circuit is open, and if it is 1, it is closed (obviously). For details, please rfer to the circuit diagram.

Apocalypse of Hell

Now, let's finally explain the game. The reason why I chose this game is its speed and difficulty. The machine language routines are full of bug fix routines.

I just want to let you know that I can do this much, so please enter it.

As for the rules, I (arbitrarily) thought that 1 point per base was too low for a game this difficult, so in the HC version I set it at 10 points per base and 1 point for each step taken and used 2 bytes for scoring.

The keys are [I] up, [J] down and [A] to drop bombs, allowing you to fire continuously. The top of the HC keyboard is a little loose, so you want to avoid hitting the keys as much as possible. Another difference from the PC version is that there are only four characters vertically, so the terrain was not changed. However, the enemy bullets are fast, so I think it is difficult enough. If you still think this is easy, try changing the comparison value on line 210.

Make sure the scroll speed is set to 9. Otherwise it will be slow and frustrating.

When entering a program, first type MEMSET &HCB5 into the keyboard and the execute it. If you forget, you will be in trouble. The machine code is in $0A50…$0C5A. How to use the monitor is described on page 40 of the operation manual.

At the End

This time, I didn't use any routines in ROM. In other words, I'm running at the BASIC level. Someone please decipher the ROM (especially the print routine).

To all HC users across the country, the HC-20 is the best. Let's all release our software together and make the HC a more popular machine!

References

1) Atsushi Uno, "Apocalypse Now", I/O, December 1980 issue.

2) HC-20 Operation Manual, Epson.

```
10 '**************
20 '* APOCALYSE  *
30 '*        NOW *  I/O Magazine 03/1983
40 '*    by S.T  *  I rise
50 '*  82/11/14  *  J fall
60 '**************  A drop bombs
70 MEMSET &H0C5D
70 REM LOADM "CAS0:VRAM.BIN",&H0A50
80 GOSUB 2000
100 WIDTH20,4:DEFINTA-Z:RANDOMIZEVAL(RIGHT$(TIME$(,2))
110 POKE&H7E,&H80:POKE&H11E,11:POKE&H11F,&HCA:V=PEEK(&H270)*256+PEEK(&H271)+8
120 PRINT CHR$(23);:X=224:FORI=1TO12:FORJ=1TO2:LOCATEI,J:PRINTCHR$(X);:X=X+1:NEXTJ,I:GOSUB310
130 FORI=0TO38:LINE(I+3,0)-(I+3,31),PSET:LINE(80-I,0)-(80-I,31),PSET:NEXT
140 FORI=0TO15:LINE(6,15-I)-(77,15-I),PRESET:LINE(6,616+I)-(77-16+I),PRESET:NEXT
150 FORI=0TO2:LOCATE1,I:PRINTSPC(12);:NEXT:LOCATE1,3:PRINTSTRING$(12,"#");
160 LOCATE14,0:PRINT"SCORE";:LOCATE14,1:PRINT"
0";:LOCATE14,2:PRINT"High";:LOCATE14,3:PRINTUSING"#####";HS;
170 POKE&H11F,&H68:POKE&HA40,0:POKE&HA41,0:POKE&HA4E,0:POKE&HA4F,0
180 LOCATE8,0:PRINTCHR$(233)+CHR$(234);
190
EXEC&HA50:IFRND(1)<.2THENLOCATE1,3:PRINTCHR$(228);ELSELOCATE1,3:PRINTCHR$(INT(RND(1)*4+224));
200 LOCATE1,4:PRINTUSING"#####";PEEK(&HA4E)*256+PEEK(&HA4F);
210 IFRND(1)<.4THENLOCATERND(1)*10+2,2:PRINTCHR$(229);
220 IFPEEK(V+PEEK(&HA41)*20)=229THEN
POKE&HA40,3:GOTO240ELSEONPEEK(&HA40)+1GOTO190,230,230,240,240
230 SOUND10,2:GOTO190
240 LOCATE8,PEEK(&HA41):PRINT"**";:FORI=1TO5:SOUND4,1:SOUND2,1:NEXT
250 LOCATE1,0:IFPEEK(&HA40)=3THENPRINT" Kiwai Leta!";ELSEPRINT"Shi Mennike Kitotsu!";
260 LOCATE2,1:PRINT"*GAMEOVER*":SOUND16,8
270 S=PEEK(&HA4E)*256+PEEK(&HA4F):IFS>HSTHENHS=S
280 LOCATE14,3:PRINTUSING"#####";HS;
290 LOCATE2,3:PRINT"Replay  <Y>";:POKE&H20,0
300 A$=INKEY$:IFA$="Y"THEN130ELSEIFA$<>"N"THEN300ELSECLS:END
310 RESTORE:FORI=1TO19:READA,B:SOUNDA,B:NEXT:RETURN
320 DATA5,3,2,1,5,1,7,5,5,5,7,3,5,1,7,1,9,5,7,5,9,3,7,1,9,1,39,5,32,5,7,3,32,1,7,1,9,7
2000 REM --- Epson HX-20      ---
2001 REM --- M. Hepperle 2024 ---
2002 REM --- adjust BASIC starting address
2003 REM --- load the code bytes
2004 REM --- Hex Code Loader ---
2005 MEMSET &H0C5D
2006 N%=0
2007 READ C$
2008 IF C$="DONE" THEN 2013
2009 N%=N%+1 : IF N%=10 THEN PRINT "."; : N%=0
2010 C%=VAL("&H"+C$)
2011 IF LEN(C$)=4 THEN A%=C% : N%=0 : GOTO 2007
2012 POKE A%,C% : A%=A%+1 : GOTO 2007
2013 PRINT
2014 REM SAVEM "CAS0:VRAM.BIN",&H0A50,&H0C5C
2015 END
2016 DATA 0A50,7F,0A,40,86,0B
2017 DATA F6,0A,41,BD,0B
2018 DATA 50,C6,20,E7,00
2019 DATA C6,2E,E7,01,7D
2020 DATA 0A,43,27,0A,F6
2021 DATA 0A,42,BD,0B,30
2022 DATA 86,E7,A7,00,86
2023 DATA 0B,5F,BD,0B,50
2024 DATA E6,00,C1,E5,26
2025 DATA 02,C6,20,E7,01
```

```
2026 DATA 4A,26,EF,C6,01
2027 DATA 86,0B,BD,0B,50
2028 DATA 37,E6,00,C1,E5
2029 DATA 26,04,BD,0B,B4
2030 DATA 01,E7,01,33,4A
2031 DATA 26,EC,5C,C1,04
2032 DATA 26,E5,01,86,F7
2033 DATA 97,20,96,22,81
2034 DATA FD,27,06,81,FB
2035 DATA 27,0C,20,15,7D
2036 DATA 0A,41,27,10,7A
2037 DATA 0A,41,20,0B,B6
2038 DATA 0A,41,81,03,24
2039 DATA 03,7C,0A,41,01
2040 DATA 7D,0A,43,26,14
2041 DATA 86,FB,97,20,96
2042 DATA 22,81,FD,01,26
2043 DATA 41,7C,0A,43,B6
2044 DATA 0A,41,B7,0A,42
2045 DATA 7C,0A,42,86,08
2046 DATA F6,0A,42,BD,0B
2047 DATA 50,A6,00,81,E4
2048 DATA 27,10,7E,0B,A4
2049 DATA 01,01,01,01,01
2050 DATA 01,7C,0A,40,86
2051 DATA 8F,20,0E,FC,0A
2052 DATA 4E,C3,00,0A,FD
2053 DATA 0A,4E,86,2A,7C
2054 DATA 0A,40,7C,0A,40
2055 DATA A7,00,7F,0A,43
2056 DATA 01,86,08,F6,0A
2057 DATA 41,BD,0B,50,A6
2058 DATA 00,81,E5,27,08
2059 DATA 25,02,20,09,7E
2060 DATA 0B,BE,01,86,03
2061 DATA B7,0A,40,86,E8
2062 DATA A7,00,4C,A7,01
2063 DATA FC,0A,4E,C3,00
2064 DATA 01,2A,03,83,80
2065 DATA 00,FD,0A,4E,39
2066 DATA 01,01,01,01,01
2067 DATA 01,36,37,36,86
2068 DATA 14,3D,18,33,3A
2069 DATA 18,F3,02,70,18
2070 DATA 33,32,39,01,01
2071 DATA 01,01,01,01,01
2072 DATA 08,10,20,10,08
2073 DATA 04,08,08,30,40
2074 DATA 40,30,08,04,02
2075 DATA 0C,10,08,08,04
2076 DATA 04,02,01,06,00
2077 DATA 28,1E,1E,28,00
2078 DATA 00,08,04,02,00
2079 DATA 00,00,10,10,28
2080 DATA 02,00,40,00,0B
2081 DATA 00,01,00,04,24
2082 DATA 34,74,78,34,24
2083 DATA 24,24,20,10,08
2084 DATA 25,07,86,E6,A7
2085 DATA 00,7E,0B,19,81
2086 DATA 20,27,F5,7E,0A
2087 DATA FB,18,83,00,13
2088 DATA 18,E7,00,C6,20
2089 DATA 39,81,20,26,03
2090 DATA 7E,0B,34,86,04
2091 DATA 7E,0B,31,00,10
2092 DATA 10,FC,10,40,00
2093 DATA 04,04,03,02,00
2094 DATA F8,20,20,FE,10
2095 DATA F0,0F,10,20,21
2096 DATA 24,27,00,00,84
2097 DATA 48,30,CC,30,00
2098 DATA 14,22,11,0F,00
2099 DATA 50,58,50,00,20
2100 DATA 00,1D,15,1D,20
2101 DATA 1C,FE,20,22,04
```

```
2102 DATA 80,40,03,0C,30
2103 DATA 00,07,08,40,C0
2104 DATA 40,40,80,00,08
2105 DATA 0F,00,08,07,00
2106 DATA 7C,54,FC,54,7C
2107 DATA 00,24,15,07,15
2108 DATA 24,00,20,FE,20
2109 DATA 22,04,00,17,20
2110 DATA 03,14,20,00,00
2111 DATA 20,24,24,E4,24
2112 DATA 10,0C,03,20,3F
2113 DATA 00,24,20,00,00
2114 DATA 20,10,03,0C,10
2115 DATA 00,00,24,A8,E6
2116 DATA AC,18,40,54,28
2117 DATA 1F,14,00,21,1A
2118 DATA 54,D4,7C,40,40
2119 DATA 00,40,7F,00,1A
2120 DATA 21,00,01,01,48 : REM 0C58 ... 0C5C
2121 DATA DONE
```

VRAM Access

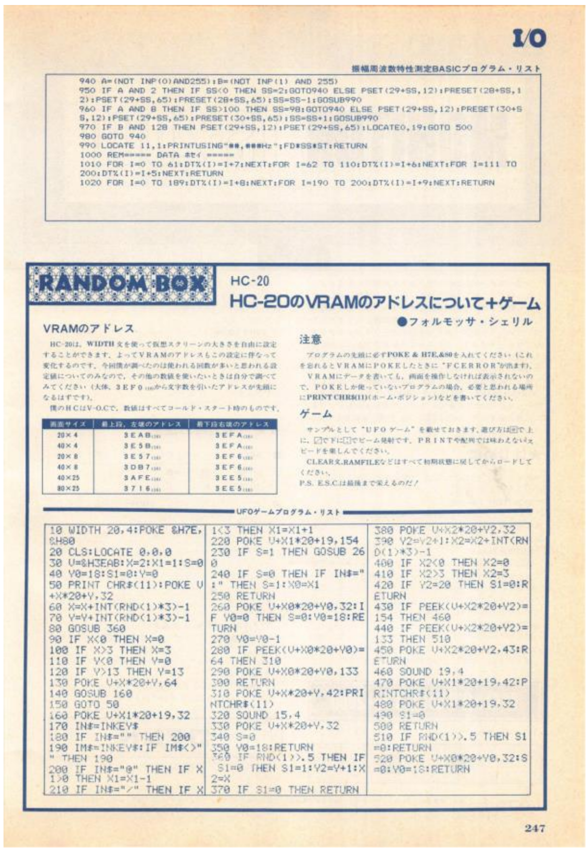
The “Random Box” section of the December 1982 issue of I/O (V7N12) contained a short BASIC program demonstrating how to access the virtual RAM of the HC-20.

ISSN 0387-6675
ホビー・エレクトロニクスの情報誌
1982 12 VOL.7 NO.12
I/O アイ・オー
Microcomputer
Synthesizer
TV Game
Robot
Laser
特集 ビジネス用簡易言語の制作
PC-8801/縦横計算プログラムレス・プログラム
FP-1100全回路図
MZ-80B MZ-2000 モニタ/TOS
PC-6001 ショート・コマンド
PASOPIA システム・サブルーチン
VIC BUILDING ALIEN
FM-8 ホルホル
PC-6001 ピラミッド
MERRY CHRISTMAS

水溶液シミュレーション
CLIST
ピラミッド
HOBBY ELECTRONICS JOURNAL I/O
UFO WARS
ホルホル
DINING KITCHEN

CONTENTS 1982年12月号

★イラスト=はらJIN+きむらしんじ+馬場隆信+川名保雄



I/O

振幅周波数特性測定BASICプログラム・リスト

```
940 A=(NOT INP(0)AND255):B=(NOT INP(1) AND 255)
950 IF A AND 2 THEN IF SS<0 THEN SS=2:GOTO940 ELSE PSET(29+SS,12):PRESET(28+SS,1
2):PSET(29+SS,65):PRESET(28+SS,65):SS=SS-1:GOSUB990
960 IF A AND 8 THEN IF SS>100 THEN SS=98:GOTO940 ELSE PSET(29+SS,12):PRESET(30+S
S,12):PSET(29+SS,65):PRESET(30+SS,65):SS=SS+1:GOSUB990
970 IF B AND 128 THEN PSET(29+SS,12):PSET(29+SS,65):LOCATE0,19:GOTO 500
980 GOTO 940
990 LOCATE 11,1:PRINTUSING"##,###Hz";FD*SS*ST:RETURN
1000 REM===== DATA ホセイ =====
1010 FOR I=0 TO 61:DT%(I)=I+7:NEXT:FOR I=62 TO 110:DT%(I)=I+6:NEXT:FOR I=111 TO
200:DT%(I)=I+5:NEXT:RETURN
1020 FOR I=0 TO 189:DT%(I)=I+8:NEXT:FOR I=190 TO 200:DT%(I)=I+9:NEXT:RETURN
```

RANDOM BOX

HC-20

HC-20のVRAMのアドレスについて+ゲーム

●フォルモッサ・シェリル

VRAMのアドレス

HC-20は、WIDTH文を使って仮想スクリーンの大きさを自由に設定することができます。よってVRAMのアドレスもこの設定に伴って変化するのです。今回僕が調べたのは使われる回数が多いと思われる設定値についてのみなので、その他の数値を使いたいときは自分で調べてみてください（大体、3EF0(16)から文字数を引いたアドレスが先頭になるはずです）。

僕のHCはV-0.Cで、数値はすべてコールド・スタート時のものです。

画面サイズ	最上段，左端のアドレス	最下段右端のアドレス
20×4	3EAB(16)	3EFA(16)
40×4	3E5B(16)	3EFA(16)
20×8	3E57(16)	3EF6(16)
40×8	3DB7(16)	3EF6(16)
40×25	3AFE(16)	3EE5(16)
80×25	3716(16)	3EE5(16)

注意

プログラムの先頭に必ずPOKE & H7E,&80を入れてください（これを忘れるとVRAMにPOKEしたときに"FCERROR"が出ます）。

VRAMにデータを書いても、画面を操作しなければ表示されないので、POKEしか使っていないプログラムの場合、必要と思われる場所にPRINT CHR$(11)(ホーム・ポジション)などを書いてください。

ゲーム

サンプルとして"UFOゲーム"を載せておきます。遊び方は@で上に、/で下に:でビーム発射です。PRINTや配列では味わえないスピードを楽しんでください。

CLEAR文,RAMFILEなどはすべて初期状態に戻してからロードしてください。

P.S. E.S.Cは最後まで栄えるのだ！

UFOゲームプログラム・リスト

```
10 WIDTH 20,4:POKE &H7E,
&H80
20 CLS:LOCATE 0,0,0
30 U=&H3EAB:X=2:X1=1:S=0
40 Y0=18:S1=0:Y=0
50 PRINT CHR$(11):POKE U
+X*20+Y,32
60 X=X+INT(RND(1)*3)-1
70 Y=Y+INT(RND(1)*3)-1
80 GOSUB 360
90 IF X<0 THEN X=0
100 IF X>3 THEN X=3
110 IF Y<0 THEN Y=0
120 IF Y>13 THEN Y=13
130 POKE U+X*20+Y,64
140 GOSUB 160
150 GOTO 50
160 POKE U+X1*20+19,32
170 IN$=INKEY$
180 IF IN$="" THEN 200
190 IM$=INKEY$:IF IM$<>"
" THEN 190
200 IF IN$="@" THEN IF X
1>0 THEN X1=X1-1
210 IF IN$="/" THEN IF X
1<3 THEN X1=X1+1
220 POKE U+X1*20+19,154
230 IF S=1 THEN GOSUB 26
0
240 IF S=0 THEN IF IN$="
:" THEN S=1:X0=X1
250 RETURN
260 POKE U+X0*20+Y0,32:I
F Y0=0 THEN S=0:Y0=18:RE
TURN
270 Y0=Y0-1
280 IF PEEK(U+X0*20+Y0)=
64 THEN 310
290 POKE U+X0*20+Y0,133
300 RETURN
310 POKE U+X*20+Y,42:PRI
NTCHR$(11)
320 SOUND 15,4
330 POKE U+X*20+Y,32
340 S=0
350 Y0=18:RETURN
360 IF RND(1)>.5 THEN IF
 S1=0 THEN S1=1:Y2=Y+1:X
2=X
370 IF S1=0 THEN RETURN
380 POKE U+X2*20+Y2,32
390 Y2=Y2+1:X2=X2+INT(RN
D(1)*3)-1
400 IF X2<0 THEN X2=0
410 IF X2>3 THEN X2=3
420 IF Y2=20 THEN S1=0:R
ETURN
430 IF PEEK(U+X2*20+Y2)=
154 THEN 460
440 IF PEEK(U+X2*20+Y2)=
133 THEN 510
450 POKE U+X2*20+Y2,43:R
ETURN
460 SOUND 19,4
470 POKE U+X1*20+19,42:P
RINTCHR$(11)
480 POKE U+X1*20+19,32
490 S1=0
500 RETURN
510 IF RND(1)>.5 THEN S1
=0:RETURN
520 POKE U+X0*20+Y0,32:S
=0:Y0=18:RETURN
```

247

Figure 4: Cover page, table of contents and article from I/O 12/1982.

Translation of the Japanese text:

RANDOM BOX – HC-20 About VRAM addressing

Formosa Cheryl

VRAM Address

The HC-20 allows you to freely set the size of the virtual screen using the WIDTH statement. Therefore, the VRAM address also changes according to this setting. This time I only looked up the setting values that are likely to be used frequently. If you want to use other values, please look them up yourself (the address starts with $3EF0 minus the number of characters should generally be used).

My HC-20 i version 0.C. and all numbers are from a cold start.

WIDTH	Top Left	Bottom Right
20 x 4	3EAB	3EFA
40 x 4	3E5B	3EFA
20 x 8	3E57	3EF6
40 x 8	3DB7	3EF6
20 x 25	3AFE	3EE5
80 x 25	3716	3EE5

Notice

Be sure to put `POKE &H7E,&H80` at the beginning of your program (if you forget this, an “FC ERROR” will appear when you POKE into VRAM).

Even if you write data into VRAM, it will not be displayed unless you operate the screen. So if your program only uses POKE, insert a `PRINT CHR$(11)` (home cursor) or use a similar screen control code to refresh the screen.

Game

Here is a sample of the “UFO Game”. To play, press up and down to fire the beam. Enjoy the speed that you can't get with PRINT. Please reset all CLEAR statements, RAMFILE, etc. to their initial states before loading.

P.S. May E.S.C. prosper to the end!

Keys used:

@ /   up down?

:

```
10 WIDTH 20,4:POKE &H7E,&H80
20 CLS:LOCATE 0,0,0
30 V=&H3EAB:X=2:X1=1:S=0
40 Y0=18:S1=0:Y=0
50 PRINT CHR$(11):POKE V+X*20+Y,32
60 X=X+INT(RND(1)*3)-1
70 Y=Y+INT(RND(1)*3)-1
80 GOSUB 360
90 IF X<0 THEN X=0
100 IF X>3 THEN X=3
110 IF Y<0 THEN Y=0
120 IF Y>13 THEN Y=13
130 POKE V+X*20+Y,64
140 GOSUB 160
150 GOTO 50
160 POKE V+X1*20+19,32
170 IN$=INKEY$
180 IF IN$="" THEN 200
190 IM$=INKEY$:IF IM$<>"" THEN 190
200 IF IN$="@" THEN IF X1>0 THEN X1=X1-1
210 IF IN$="/" THEN IF X1<3 THEN X1=X1+1
220 POKE V+X1*20+19,154
230 IF S=1 THEN GOSUB 260
240 IF S=0 THEN IF IN$=":" THEN S=1:X0=X1
250 RETURN
260 POKE V+X0*20+Y0,32:IF Y0=0 THEN S=0:Y0=18:RETURN
270 Y0=Y0-1
280 IF PEEK(V+X0*20+Y0)=64 THEN 310
290 POKE V+X0*20+Y,133
300 RETURN
310 POKE V+X*20+Y,42:PRINT CHR$(11)
320 SOUND 15,4
330 POKE V+X*20+Y,32
340 S=0
350 Y0=18:RETURN
360 IF RND(1)>.5 THEN IF S1=0 THEN S1=1:Y2=Y+1:X2=X
370 IF S1=0 THEN RETURN
380 POKE V+X2*20+Y2,32
390 Y2=Y2+1:X2=X2+INT(RND(1)*3)-1
400 IF X2<0 THEN X2=0
410 IF X2>3 THEN X2=3
420 IF Y2=20 THEN S1=0:RETURN
430 IF PEEK(V+X2*20+Y2)=154 THEN 460
440 IF PEEK(V+X2*20+Y2)=133 THEN 510
450 POKE V+X2*20+Y2,43:RETURN
460 SOUND 19,4
470 POKE V+X1*20+19,42:PRINT CHR$(11)
480 POKE V+X1*20+19,32:PRINT CHR$(11)
490 S1=0
```

```
500 RETURN
510 IF RND(1)>.5 THEN S1=0:RETURN
520 POKE V+X0*20+Y0,32:S=0:Y0=18:RETURN
```