

NAKAMOZU TINY BASIC

H68/TR, TV



超高速BASICインタプリタ中百舌鳥TINY BASICをH68/TR+H68/TVで走らすためのI/Oルーチンと移植テクニック。ダンプリストのチェックがスピーディに行なえるチェックサム付

日下田 恵一

H68/TR+TV用 I/Oルーチン

```

0236      *
0236      * I/O ROUTINE
0236      * FOR H68TV01
0236      * CONNECT TO
0236      * TV MONITOR
0236      *
0E92      *
0E92      * OUTH
0E92      *
0E92 81 05      CNPA #5
0E94 26 03      BNE L01
0E96 7E 1E56    JMP $1E56
0E99 36         L01 PSHA
0E9A 81 20      CNPA ##20
0E9C 2A 0A      BPL L02
0E9E 81 0D      CNPA ##D
0EA0 27 06      BEQ L02
0EA2 81 0F      CNPA ##F
0EA4 26 05      BNE L03
0EA6 86 7E      LDAA ##7E
0EA8 BD 1EFE L02 JSR $1EFE
0EAB 32         L03 PULA
0EAC 7E 0000    JMP $0000
0EAF          *
0EAF          * 0.5SEC DELAY
0EAF          *
0EAF 7F 0C63    CLR $0C63
0EB2 7E 0000    JMP $0000
0EB5          *
                                END
*** LABELS ***

L01 0E99      L02 0EA8
L03 0EAB

RESEMBLE END.
    
```

NTBをH68TV01に移植する最も簡単な方法として、I/OルーチンをTV01付属のTVモニタ内のサブルーチンに接続する方法を紹介し、入力ルーチンは山下氏のをそのまま使うことにし、出力ルーチンにバッチを入れることにしました。

ところで、NTBを動かすためには\$0C00から\$0FFFまでのH68TRとH68TVの間のメモリの不連続領域にRAMが実装されていることが必要です。もし、実装されていないければ、インタプリタの変更やI/Oルーチンのリロケートが必要となります。ここではRAMは実装されているものとします。また、キーボードにH68KBが使われている場合は今回は対象外です。ごめんなさい。

まず、インタプリタ及びI/Oルーチンをオリジナルのアドレスでメモリにロードしてください。正しくロードされたかのチェックにはサムチェックプログラムを使うと便利です。チェックサムが本誌に載っているものと一致していればOKです。(サムチェックプログラムはTVモニタの管理下で動かしてください。1ページごとにアドレスとチェックサムを出力し、CRを入力するとモニタにもどります。他のキーを押すと次のページを表示します。)次に表のようにインタプリタとI/Oルーチンを変更し、TVモニタと接続するためのプログラムをロードしてください。コールドスタートは\$0100、ホットスタートは\$0103です。スタートすると画面左上にREADYの文字が表示され入力待ちとなります。

以上の操作により、NTBを擬似TTYベースで動かすことができます。I/Oルーチンのコントロール・コードはオリジナルと同様に使えます。ただし、出力CASET解除ではリレーOFFの前に0.5秒のディレイを置いてあります。(テープレコーダによっては

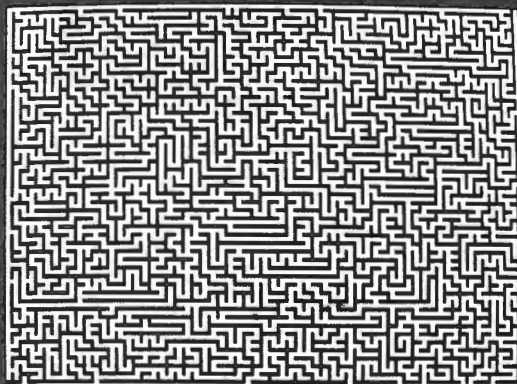
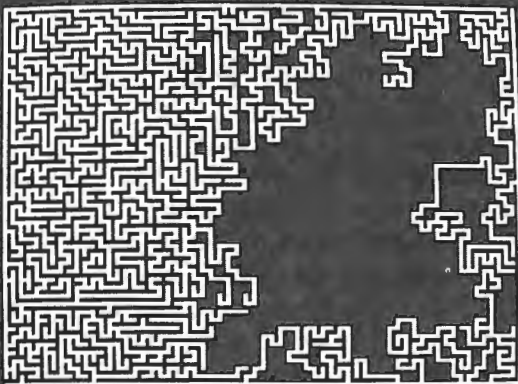
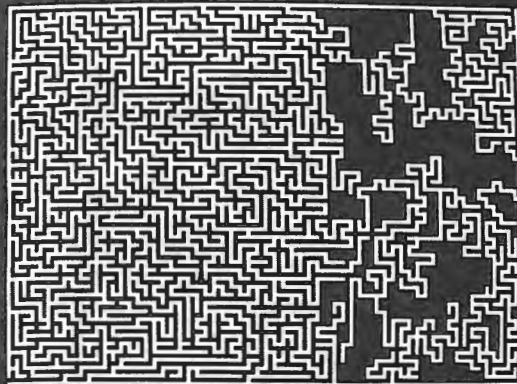
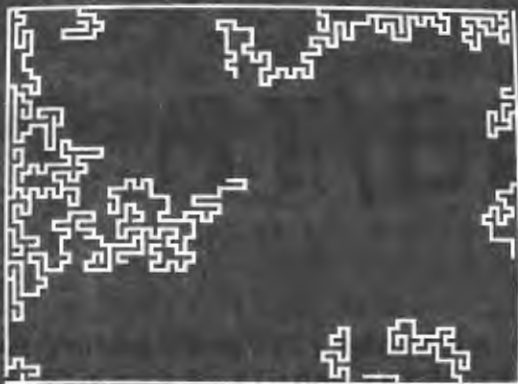
SAVEを行なったときに、最後のコードが記録されないことがあるため)。また、LFのコードは無効にしているため、CTキーでプログラムを一時停止させたときの再開にはLFキーを用いるとよいでしょう。このI/OルーチンではCLR、CURSは無効です。また、グラフィック命令を行なうとERR9となります。NTBの場合、システムに依存

表1 インタプリタ変更箇所

アドレス	変更前	変更後
0106	0E92	0F80
0108	33FF	17FF
01C2	A047	0F47
01D4	A047	0F47
02DB	A047	0F47
091C	A047	0F47
025D	A080	0F80
06F7	A04A	0F4A
0709	A080	0F80
098D	3CB6	0A2E
0992	0E76	0A2E
099B	0E67	0A2E
09AD	0E47	0A2E
09C4	3BFO	0A2E

表2 I/Oルーチン変更箇所

アドレス	変更前	変更後
0C74	0CD1	1E56
0C89	8D11	0101
0C8C	0D00	0E92
0DOF	7F0C63	BDOEAF



コンピュータ・メイズ プログラム・リスト

```

10 REM COMPUTER MEIRO
20 REM NAKAMOZU TINY BASIC
30 REM BY H.YAMASHITA
40 G.300
100 IF Y>92 RET
110 IF !P(X,Y+2)=1 RET
120 !W(X,Y):!W(X,Y+1):!W(X,Y+2)
130 Y=Y+2:K=6:RET
150 IF X<4 RET
160 IF !P(X-2,Y)=1 RET
170 !W(X,Y):!W(X-1,Y):!W(X-2,Y)
180 X=X-2:K=6:RET
200 IF Y<4 RET
210 IF !P(X,Y-2)=1 RET
220 !W(X,Y):!W(X,Y-1):!W(X,Y-2)
230 Y=Y-2:K=6:RET
250 IF X>124 RET
260 IF !P(X+2,Y)=1 RET
270 !W(X,Y):!W(X+1,Y):!W(X+2,Y)
280 X=X+2:K=6:RET
300 REM ** MAIN PROGRAM **
310 P.C.3:
340 F.X=1T0127:!W(X,1):!W(X,95):
N.X
350 F.Y=2T094:!W(1,Y):!W(127,Y):
N.Y

```

```

360 !B(126,1):!B(1,94):Z=0
400 D0:K=5
410 D0:X=R.(64)*2+1:Y=R.(48)*2+1
420 U.!P(X,Y)=1
430 D0:60S.R.(4)*50+100:K=K-1
440 U.K=0
450 Z=Z+1
460 U.Z=200
500 F.I=1T0127 S.2:F.J=1T095 S.2
510 IF !P(I,J)=0 T.560
520 X=I:Y=J:K=5
530 D0:60S.R.(4)*50+100:K=K-1
540 U.K=0
560 N.J:N.I
580 P.C.7:C.6:A=GET$
590 P.C.5:

```

```

READY
>_

```

する命令が多いため、各自のシステムにあわせてルーチンを作る必要があります。なお、プログラム領域は\$0F80から\$17FFまで、また\$0F00から\$0F47までがマシスタック、\$0F48から\$0F7FがIXスタックとなります。

NTBの特徴には、高速、命令数が多いなどがあるのですが、H68派の人にとってはグラフィック記号を直接入れられることも魅力でしょう。またリアルタイム入力やCRTベースの命令を持っているので、CRTディスプレイならではのゲームを作ることも可能です。グラフィック命令も使える（イニシャライズとプロットルーチンが必要）ので、グラフィックディスプレイを使ったプログラムの開発も容易になると思います。

注意

① IX配列は使用する前にIX変数を定義

しておかないと、インタプリタまたはTVモニタを壊して暴走を起すことがあります。

② NTBのSAVE及びLOADルーチンは、まだ多少安定性に欠けるところがある

ようです。

(H68/TVの場合)

テストプログラム1(実行時間4.8秒)

テストプログラム2(実行時間14.2秒)

```
10 P.*START*
100 REM AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
110 REM BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB
120 REM CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
130 REM DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD
140 REM EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE
150 REM FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
160 REM GGGGGGGGGGGGGGGGGGGGGGGGGGGGGGG
170 REM HHHHHHHHHHHHHHHHHHHHHHHHHHHHHH
180 REM IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII
190 REM JJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJ
500 K=0
520 K=K+1
530 IF K<1000 THEN520
540 P.*END*
```

```
>LIST
10 P.*START*
100 REM AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
110 REM BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB
120 REM CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
130 REM DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD
140 REM EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE
150 REM FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
160 REM GGGGGGGGGGGGGGGGGGGGGGGGGGGGGGG
170 REM HHHHHHHHHHHHHHHHHHHHHHHHHHHHHH
180 REM IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII
190 REM JJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJ
500 K=0
510 DO
520 K=K+1
530 UNTIL K=1000
540 P.*END*

READY
_
```

1000までの素数を出力するプログラム(実行時間3.6秒) NAKAMOZU TINY BASICチェックサム・リスト

```
>LIST
10 REM 1000 MADE NO SOSU
20 F.A=5 T0995 S.6:20(A)=0:20(A+2)=0:N.A
30 F.A=5 T031 S.2:IF20(A) N.A
40 F.B=24A T0100 S.A:20(B)=1:N.B
50 N.A
80 P.C.75
90 P.*# 1000 MADE NO SOSU #*
100 P.U.*2,U.*3,
110 F.A=5 T0995 S.6
115 IF20(A)=0 P.U.*A,
120 IF20(A+2)=0 P.U.*A+2,
130 N.A

## 1000 MADE NO SOSU ##
###2 ###3 ###5 ###7 ###11 ###13 ###17 ###19
###23 ###29 ###31 ###37 ###41 ###43 ###47 ###53
###59 ###61 ###67 ###71 ###73 ###79 ###83 ###89
###97 ###101 ###103 ###107 ###109 ###113 ###127 ###131
###137 ###139 ###149 ###151 ###157 ###163 ###167 ###173
###179 ###181 ###191 ###193 ###197 ###199 ###211 ###223
###227 ###229 ###233 ###239 ###241 ###251 ###257 ###263
###269 ###271 ###277 ###281 ###283 ###293 ###307 ###311
###313 ###317 ###331 ###337 ###347 ###349 ###353 ###359
###367 ###373 ###379 ###383 ###389 ###397 ###401 ###409
###419 ###421 ###431 ###433 ###439 ###443 ###449 ###457
###461 ###463 ###467 ###479 ###487 ###491 ###499 ###503
###509 ###521 ###523 ###541 ###547 ###557 ###563 ###569
###571 ###577 ###587 ###593 ###599 ###601 ###607 ###613
###617 ###619 ###631 ###641 ###643 ###647 ###653 ###659
###661 ###667 ###677 ###683 ###691 ###697 ###709 ###719
###727 ###733 ###739 ###743 ###751 ###757 ###761 ###769
###773 ###787 ###797 ###809 ###811 ###821 ###823 ###827
###829 ###839 ###853 ###857 ###859 ###863 ###877 ###881
###883 ###887 ###907 ###911 ###919 ###929 ###937 ###941
###947 ###953 ###967 ###971 ###977 ###983 ###991 ###997
```

ADDRESS	CSUM	ADDRESS	CSUM	ADDRESS	CSUM	ADDRESS	CSUM
0100-011F	51	0120-013F	18	0080-001F	9E	0020-003F	76
0140-015F	6C	0160-017F	DF	0040-005F	53	0060-007F	E1
0180-019F	CA	01A0-01BF	C4	0080-009F	CA	00A0-00BF	F4
01C0-01DF	76	01E0-01FF	5B	00C0-00DF	6B	00E0-00FF	32
0200-021F	F8	0220-023F	28	0000-001F	6C	0020-003F	92
0240-025F	F3	0260-027F	A9	0040-005F	48	0060-007F	E4
0280-029F	93	02A0-02BF	E3	0080-009F	FF	00A0-00BF	56
02C0-02DF	D6	02E0-02FF	11	00C0-00DF	75	00E0-00FF	EC
0300-031F	6A	0320-033F	AC	0000-001F	8D	0020-003F	72
0340-035F	24	0360-037F	51	0040-005F	FC	0060-007F	26
0380-039F	A5	03A0-03BF	25	0080-009F	F4	00A0-00BF	DD
03C0-03DF	EB	03E0-03FF	7C	00C0-00DF	3F	00E0-00FF	5D
0400-041F	AE	0420-043F	C7	0000-001F	4D	0020-003F	20
0440-045F	FE	0460-047F	D5	0040-005F	6B	0060-007F	6F
0480-049F	C6	04A0-04BF	93	0080-009F	87	00A0-00BF	51
04C0-04DF	34	04E0-04FF	A2	00C0-00DF	65	00E0-00FF	F3
0500-051F	B2	0520-053F	EB	0000-001F	60	0020-003F	AF
0540-055F	86	0560-057F	13	0040-005F	BB	0060-007F	C8
0580-059F	8E	05A0-05BF	AF	0080-009F	CF	00A0-00BF	1F
05C0-05DF	21	05E0-05FF	55	00C0-00DF	A9	00E0-00FF	9B
0600-061F	F7	0620-063F	35	0000-001F	37	0020-003F	38
0640-065F	86	0660-067F	1C	0040-005F	EC	0060-007F	51
0680-069F	F7	06A0-06BF	EF	0080-009F	18	00A0-00BF	6C
06C0-06DF	09	06E0-06FF	D3	00C0-00DF	69	00E0-00FF	E9
0700-071F	35	0720-073F	50	0000-001F	2C	0020-003F	19
0740-075F	53	0760-077F	A3	0040-005F	74	0060-007F	6E
0780-079F	94	07A0-07BF	3E	0080-009F	A4		
07C0-07DF	C6	07E0-07FF	91	/			

チェック・サム表示プログラム(このプログラムを使用してチェック・サムを表示することができる)

0236	*	173D C6 02	LDAB #2	1793 27 03	BEQ L26
0236	* SUM CHECK	173F 8D 21	BSR L1E	1795 5A	DECB
0236	* H68TR+TV01	1741 09	DEX	1796 26 F6	BNE L25
0236	*	1742 8D 26	BSR L20	1798 DF 00	L26 STX L01
0000	*	1744 C6 03	LDAB #3	179A 08	INX
0000	* WORK	1746 8D 1A	BSR L1E	179B DF 05	STX L06
0000	*	1748 33	PULB	179D 97 02	STAA L03
0000	L01 RMB 1 ADRH	1749 DE 00	LDX L01	179F 39	RTS
0001	L02 RMB 1 ADRL	174B 9C 07	CPX L08 EAD?	17A0	*
0002	L03 RMB 1 SUM	174D 27 03	BEQ L13	17A0 20	L30 FCB \$20
0003	L04 RMB 1	174F 5A	DECB	17A1 41	FCB \$41 A
0004	L05 RMB 1	1750 26 D1	BNE L12	17A2 44	FCB \$44 D
0005	L06 RMB 1	1752 BD F60D L13	JSR \$F60D	17A3 44	FCB \$44 D
0006	L07 RMB 1	1755 DE 00	LDX L01	17A4 52	FCB \$52 R
0007	L08 RMB 2 EADR	1757 9C 07	CPX L08 EAD?	17A5 45	FCB \$45 E
1700	*	1759 27 04	BEQ L14	17A6 53	FCB \$53 S
1700	* INITIALIZE	175B 81 0D	CMPA #\$D CR?	17A7 53	FCB \$53 S
1700	*	175D 26 B4	BNE L10	17A8 20	FCB \$20
1700 CE 00FF	LDX #\$FF	175F 7E F107 L14	JMP \$F107	17A9 20	FCB \$20
1703 DF 00	STX L01	1762	*	17AA 43	FCB \$43 C
1705 08	INX	1762 86 20	L1E LDAA #\$20	17AB 53	FCB \$53 S
1706 DF 05	STX L06	1764 8D 20	L1F BSR L23	17AC 55	FCB \$55 U
1708 CE 0E91	LDX #\$E91	1766 5A	DECB	17AD 4D	FCB \$4D M
1708 DF 07	STX L08	1767 26 FB	BNE L1F	17AE 20	FCB \$20
170D 86 05	LDAA #5	1769 39	RTS	17AF 20	FCB \$20
170F B7 E007	STAA \$E007	176A	*	17B0 0D	FCB \$0D EOL
1712 0E	CLI	176A	* DISP TO CRT	17B1	*
1713	*	176A A6 00	L20 LDAA 0,X		END
1713	* MAIN	176C CE 0003	LDX #L04		
1713	*	176F 6F 00	CLR 0,X		
1713 BD 1E56 L10	JSR \$1E56	1771 A7 01	STAA 1,X		
1716 C6 02	LDAB #2	1773 36	PSHA	*** LABELS ***	
1718 CE 17A0 L11	LDX #L30	1774 BD FBC7	JSR \$FBC7		
1718 BD 1A32	JSR \$1A32	1777 A6 00	LDAA 0,X	L01 0000	L02 0001
171E 5A	DECB	1779 8D 03	BSR L21	L03 0002	L04 0003
171F 26 F7	BNE L11	177B 32	PULA	L05 0004	L06 0005
1721 C6 1C	LDAB #28	177C 84 0F	ANDA #\$F	L07 0006	L08 0007
1723 37	L12 PSHB	177E 81 09	L21 CMPA #9	L10 1713	L11 1718
1724 CE 0005	LDX #L06	1780 23 02	BLS L22	L12 1723	L13 1752
1727 8D 41	BSR L20	1782 88 07	ADDA #7	L14 175F	L1E 1762
1729 CE 0006	LDX #L07	1784 8B 30	L22 ADDA #\$30	L1F 1764	L20 176A
172C 8D 3C	BSR L20	1786 7E 1EFE	L23 JMP \$1EFE	L21 177E	L22 1784
172E 8D 59	BSR L24	1789	*	L23 1786	L24 1789
1730 86 2D	LDAA #\$2D	1789	* 32 BYTE SUM	L25 178E	L26 1798
1732 8D 52	BSR L23	1789 DE 00	L24 LDX L01	L30 17A0	
1734 CE 0000	LDX #L01	178B 4F	CLRA		
1737 8D 31	BSR L20	178C C6 20	LDAB #32	ASSEMBLE END.	
1739 09	DEX	178E 08	L25 INX		
173A 09	DEX	178F AB 00	ADDA 0,X		
173B 8D 2D	BSR L20	1791 9C 07	CPX L08 EAD?		