

\*\*\*\*\*

# THE SINC TIMES

## NATIONAL SINCLAIR TIMEX NEWSLETTER

July/August 1985; vol. 4, no. 4  
\$2.00/issue, \$12.00/year

Editor: Joseph Dell'Orfano  
122 Weaver St.  
Greenwich, Ct. 06830  
(203) 531-7677

Supporting the Sinclair and Timex Personal Computers

Writers: Brian Bauer	John Bloxham	Narti Kitiyakara
3327 G Rd.	18 Lea Close	2917 Ursulines Ave
Clifton, Co. 81520	Stratford-upon-Avon	New Orleans, La. 70119
	Warwickshire	
	England CV37 9JS	

\*\*\*\*\*

### EDITORIAL

Uh oh. It seems that Uncle (oops, I mean Sir) Clive is in trouble. Sinclair Research is on the verge of bankruptcy. According to my sources they are being sued by the Hoover company for debts due to the manufacture of the Sinclair electric car. John Bloxham says in the JB Corner that Robert Maxwell, a newspaper publisher, has offered to buy out Clive's share, but I understand that he has backed out on the deal. (Can you verify this or add to it, John?) I don't know how this will effect us here in the states, but just remember that despite Sinclair's financial woes due to his electric moped, the Spectrum is still the most popular computer in England.

Speaking of finances, many of you owe dues to the club (has it really been a year now? Time flies...) Anyway, if you find an asterisk on you mailing label, then please send money so that I don't have to suspend any memberships. The group has grown some in the past year, and I expect it to grow even more next year. Just as a quick sneak preview of what's ahead, it looks like we may have the opportunity to expand through the New York area and hold monthly meetings. If there's enough interest here, I'll go ahead with my plans. That would mean a lot more communication, and a whole bunch of fresh ideas that I can put out to you folks through this newsletter. So stick with it, we may have something good here!

Incidentally, there are "heaps and gobs" (thanks, Tom!) of 2068's still available in the New York area. It seems that Timex told a few fibs when they said that they sold 70,000 units and produced only 100,000! If that's true, then where are all these new 2068's coming from? C'mon, Timex, give us a break!

On the lighter side of the news, I have received my Aerco disk system, and have a full review of it within. There's also news from the source in the JB Corner, and some machine code by our friend Narti Kitiyakara. And to round out this issue I have included some info on software available from 21st Century Electronics. Until next time then, keep on ZX'in!

## MEMBERSHIP ROSTER

Brian Bauer  
3327 G Road  
Clifton, Col. 81520

Joanne H Choly  
62 Glenwood Ave.  
Stratford, Ct. 06497

Sue Currier/Softsync  
14 E. 34th St.  
New York, N.Y. 10016

John Kemeny (BCS/TSUG)  
284 Great Rd. Apt. D-5  
Acton, Mass. 01720

Jim McKenzie  
20401 Coulson St.  
Woodland Hills, Ca. 91367

Sam Speed, Jr.  
P.O. Box 1264  
San Diego, Ca. 92118

Syntax  
Attn.: Judy Lorimer  
RD 2, Box 45/  
Harvard, Ma. 01451

TSUG  
c/o Dick Scoville  
2315 W. Club Blvd.  
Durham, N.C. 27705

Kevin Debray  
36 Arch St.  
Greenwich, Ct. 06830

John Bloxham  
18 LeaClose  
Stratford-Upon-Avon  
Warwickshire  
England CV37 9JS

Ralph Coletti  
869 Levitt Pkwy  
Rockledge, Fl. 32955

Jack Deuber  
P.O. Box 305  
Casselberry, Fl. 32707

Richard A. Kircher  
437 Penguin Dr.  
Satelite Beach, Fl. 32937

Russell C. Parker  
7814 Keeler Ave.  
Skokie, Ill. 60076

James Stanta  
8625 Shady Glen Dr.  
Orlando, Fl. 32819

Van S. Vangor  
Bethlehem Tool  
Box 346 C  
Retreat Rd.  
Island Falls, Me. 04747

CA13  
c/o Jules Gesang  
P.O. Box 725  
Bladensburg, Md. 20710

Gregory C. Harder  
F.O. Box 6493  
Denver Co. 80206

Tom Carroll  
Vassar College  
Box 1673  
Foughkeepsie, N.Y. 12601

Jeff Craddock  
333 N. 5th St.  
Burbank, Ca. 91501

E.D. Feldman  
400 S. Burnside Ave. #MJ  
Los Angeles, Ca. 90036

Narti Kitiyakara  
2917 Ursulines Ave.  
New Orleans, La. 70119

Silicon Valley S/T Users  
Attn: Rita L. Carr  
Box 4133  
Santa Clara, Ca. 95054

Edward Stone  
Rt. 11, Box 93  
Gainesville, Ga. 30501

Richard Lieberman, M.D.  
9535 Reseda Blvd., #204  
Northridge, Ca. 91324

E.K. Lindsay  
c/o Mitre Corp.  
Box 208  
Bedford, Ma. 01730

Enoch La Touche  
710 Azure Ave. NE  
Palm Bay, Fl. 32905

Lawrence Holmes Jr.  
P.O. Box 1921  
Ft. Walton Beach, Fl. 32549

Mike O'Kane  
2121 Plaza Del Amo  
Torrance, Ca. 90501

Mike Mahelski  
175 Elizabeth Blackwell St.  
Apt. 806  
Syracuse, NY. 13210

## HARDWARE REVIEW EXP/3000 DISK SYSTEM

by Joe Dell'Orfano

First, let me explain. The EXP/3000 is the system sold by 21st Century Electronics. It is completely encased, and includes one double sided double density drive, and is based around the AERCO FD-68 disk controller board. You can see what is included in the system by reading the information reproduced on the following pages, so let me try to give you a user's perspective of the drive system.

It should be noted that as of this writing, only BASIC, binary, and screen files are supported. However, AERCO promises that the system will be able to support five more types of files (see below). The system also offers four different types of disk formats: Timex, Spectrum, CP/M, or none at all. However, only the Timex format is available now.

That it's faster than tapes goes without saying. However, this setup is even faster than most disks that I've had experience with. Also, the DOS is on a separate chip, so it takes up no RAM space at all. And, finally, it includes an extra 64K bank which takes the place of the cartridge dock. This means that machine code programs (or even the Spectrum ROM) can be loaded in to this bank and run like a cartridge. I imagine that this will be the next type of Spectrum emulator we'll be seeing. Also, this 64K bank will house the CP/M system when it becomes available (I am told this will be a month or two down the road). The disadvantage to this setup, though, is that I can no longer use my cartridges. This is no big loss, however, since all I have on cartridge is Flight Simulator and States and Capitals!

Well, it seems to be a good system, but its best feature overall is its expandability. It is a full blown disk system as it stands now, and yet there is so much which has yet to be implemented on it. How much it will improve in the future remains to be seen. I think that we are on the doorsteps of a major advancement in TS and ZX computing.

## THE DISC COMMANDS

The **MOVE** command is used to write information from the computer to the disc. The information may be of any of the types supported by the cassette tape system as well as several others. The type of information is described by a period and three letters after the name you give the information. This addition to the name is commonly called an **EXTENSION**. These are the Extensions currently supported.

- .ARO AROS runs in the dock bank starting at 8000 Hex.
- .BAS BASIC program and its variables.
- .BIN BINARY data.
- .BUT BOOT program to be executed from cold start.
- .CHR CHARACTER array.
- .DAT NUMERIC array.
- .LRO LROS runs in the dock bank starting at 0.
- .SCR SCREEN is a copy of the video display.

The extension is always 3 letters, but the main part of the name may be any length from 1 to 10 characters. For example, to save the video display as **PIX**, you would enter the following command: **MOVE"PIX.SCR"**, The screen data will be saved onto the disc that was last used. If instead of using the currently selected drive you wish to specify drive C, enter **MOVE"C:PIX.SCR"**,

The **.ARO** extension: SAVES/LOADS memory in the Dock Bank, starting at address 8000H. The Chunks are selected and the program is started per the standard TIMEX rules for operating an AROS (Application ROM Orientated Software). Instead of running it from ROM, you are running it from RAM as loaded from the disc.

The **.BAS** extension: SAVES/LOADS BASIC program and data areas. An optional Decimal parameter may be specified as the starting Line Number. For example, to SAVE a program that is to commence running from Line 100 when loaded, enter **MOVE"NAME.BAS",100** To LOAD the program, enter **CAT"NAME.BAS"**, A BASIC program may be started at a different location when it is loaded by specifying a new Start Line numeral, as **CAT "Name.BAS", start line**.

The **.BIN** Extension: SAVES/LOADS binary data. It is the equivalent of CODE in the tape system. Two Decimal Parameters are required when saving: the Start address and the number of bytes. No parameters are required to LOAD the CODE back into the system. For example: to SAVE 850 bytes starting at 26440, enter **MOVE "NAME.BIN", 26440, 850**. To LOAD

it back, enter CAT "NAME.BIN", A Binary program may be loaded into a different location by specifying the new location, as CAT" Name.BIN",newnumber .

The .BUT extension: An assembly language program written at 3400H (0D40H in the SPECTRUM mode) and saved with the command MOVE"0.BUT", will be executed when the system is first turned on. Be sure that any programs that your boot program calls are on the disc. CAT"0.BUT", will load Track 0, Sector 1 of the currently selected disc into the Boot memory area. 128 Bytes are available for the Boot program.

MANCHESTER GUARDIAN WEEKLY, June 23, 1985

## *Rescue for Sinclair*

By Paul Brown

MR ROBERT MAXWELL, the publisher, has taken a controlling interest in Sir Clive Sinclair's home computer company, in a £12 million rescue campaign.

After nine hours of talks between the two men at the weekend, Mr Maxwell agreed to take a controlling share in Sinclair Research for a nominal sum in return for sinking £12 million into the company by buying the majority of a new share issue.

The rescue comes after several weeks of attempts to save the company, which has been left with £30 million worth of unsold home computer stock after a collapse in the market.

The Bank of England stepped in to help Sir Clive after Mrs Thatcher was said to be taking a "close personal interest" in his company's fate. Her Government knighted Sir Clive two years ago.

Mr Maxwell, the publisher of Mirror Group Newspapers, is taking control of Sinclair Research through another company, Hollis, which is a subsidiary of Mr Maxwell's Pergamon Press.

Sir Clive, who pioneered Britain's first pocket calculator, home computers, and pocket television, was hit by a drop in sales last Christmas which began to pile up unsold stocks.

It followed a four-year sales boom which has put computers into 18 out of every 100 British homes. Sinclair had 40% of the British market.

Mr Maxwell will become chairman of Sinclair Research, with Mr Sinclair life president.

**FOR FUN  
LIFE - IN MACHINE CODE**

**by Narti Kitiyakara**

I don't normally go in for writing my own machine code programs, but I do like to keep in practice, and this is more or less the result of that keeping in practice. You've probably already seen 'life' programs, and this one offers nothing new, except speed. It is really an inefficient program, but it is also pretty easy to understand, it reads like a program written in Pascal, but that's reasonable enough because it was first written in Pascal. The comments in each of the subroutines should make their purpose clear enough, and I think that they are small enough so that the code for them is fairly clear as well. The BASIC program need only set up the original colony, and keep calling the main routine, machine code does all the work, but I wanted it to return to BASIC so that it would be easy to break out of. I used a very simple method for setting up the colony, but it is easy enough to add a more sophisticated routine to do so (you may want to actually set them up on the screen, with a joystick, and then have the program determine where they are, or what-ever). If you want to add routines to save and load colonies it's just SAVE "???"CODE 53248,704, and you can add what-ever frills you want to that as well. As far as entering the code the first time goes, I've included the decimal, hex, and mnemonics for it, start POKEing it in at 53956. If you use an assembler and move it around (I had to put it pretty low in memory to avoid my DOS), be sure to note where the first bytes of LIFE, CLER, and ARAY are for use in the BASIC program, and also change the value of DE in the routine UNDR to the last byte of ARAY. Well that's all for now, by next time I'll have my computer back, a friend is using it for typing something right now, and will have something new, I've gotten ZIP BASIC, a BASIC compiler, and I hope to be soon getting Pascal and C for my disk drive, and I'll review at least the BASIC compiler for the next issue.

```

ARRAY 02C0h bytes for the growing surface.
DAT1 1 byte, temporary storage.
DAT2 2 bytes, temporary storage.

FRNT ; given the proper data in A print a space or "0" for the
      absence or presence of a cell.
      CP 00 ; 00 = no cell
      JR NZ,+4
JMP1 LD A,20 ; space
      JR +6
      CP 03 ; 03 = a coming birth
      JR Z,JMP1
      LD A,4F ; "0"
      RST 10
      RET

DISP ; display the entire growing surface.
      LD HL,ARRAY
      LD BC,03C0
LOP1 LD A,(HL)
      CALL FRNT
      INC HL
      DEC C
      JR NZ,LOP1
      DJNZ,LOP1
      RET

UNDR ; a new index for the array has gone under the limit.
      LD DE,D2C0 ; the end of the array for anyone moving the
                  code

EXIT ADD HL,DE
      POP DE
      POP DE
      RET

OVER ; same as under but the value is over the upper bound.
      LD DE,ARRAY
      JR EXIT

BCHK ; check to make sure the index is within the bounds of the
      array.
      PUSH DE
      PUSH HL
      LD DE,ARRAY
      SBC HL,DE
      JP M,UNDR
      LD DE,02C0
      SBC HL,DE
      JP P,OVER
      POP HL
      POP DE
      RET

```

```

GENR ; mark all new births and deaths.
      LD BC,03C0
      LD HL,ARRAY
LOP2  CALL CHCK
      LD A,(HL)
      CP 00
      JR NZ,+5
      CALL BRTH
      JR +3
      CALL DETH
      INC HL
      DEC C
      JR NZ,LOP2
      DJNZ,LOP2
      RET

CNGE ; change all markers to a life cell or no cell.
      LD HL,ARRAY
      LD BC,03C0
LOP3  LD A,(HL)
      CP 02
      JR Z,+4
      CP 03
      JR NZ,+2
      DEC (HL)
      DEC (HL)
      INC HL
      DEC C
      JR NZ,LOP3
      DJNZ,LOP3
      RET

LIFE ; display the last generation, and produce the next one.
      CALL DISP
      CALL GENR
      CALL CNGE
      RET

CLER ; clear the growing surface.
      LD HL,ARRAY
      LD BC,03C0
      SUB A
LOP4  LD (HL),A
      INC HL
      DEC C
      JR NZ,LOP4
      DJNZ,LOP4
      RET

```

Decimal code:

254	0	32	4	62	32	24	6	254	3	40	248	62	79	215
201	33	0	208	1	192	3	126	205	196	210	35	13	32	248
16	246	201	17	192	210	25	209	209	201	17	0	208	24	247
213	229	17	0	208	198	0	237	82	250	229	210	17	192	2
198	0	237	82	242	236	210	225	209	201	205	241	210	126	254
0	200	254	3	200	33	193	210	52	201	25	205	10	211	42
194	210	198	0	237	82	205	10	211	42	194	210	201	34	194
210	151	50	193	210	17	31	0	205	25	211	19	205	25	211
19	205	25	211	30	1	205	25	211	33	193	210	94	42	194
210	201	62	3	187	192	54	3	201	62	2	187	200	60	187
200	54	2	201	1	192	3	33	0	208	205	43	211	126	254
0	32	5	205	77	211	24	3	205	84	211	35	13	32	236
16	234	201	33	0	208	1	192	3	126	254	2	40	4	254
3	32	2	53	53	35	13	32	241	16	239	201	205	212	210
205	94	211	205	123	211	201	33	0	208	1	192	3	151	119
35	13	32	251	16	249	201								

Hex code:

FE	00	20	04	3E	20	18	06	FE	03	28	F8	3E	4F	D7
C9	21	00	D0	01	C0	03	7E	CD	C4	D2	23	0D	20	F8
10	F6	C9	11	C0	D2	19	D1	D1	C9	11	00	D0	18	F7
D5	E5	11	00	D0	C6	00	ED	52	FA	E5	D2	11	C0	02
C6	00	ED	52	F2	EC	D2	E1	D1	C9	CD	F1	D2	7E	FE
00	08	FE	03	C8	21	C1	D2	34	C9	19	CD	0A	D3	2A
C2	D2	C6	00	ED	52	CD	0A	D3	2A	C2	D2	C9	22	C2
D2	97	32	C1	D2	11	1F	00	CD	19	D3	13	CD	19	D3
13	CD	19	D3	1E	01	CD	19	D3	21	C1	D2	5E	2A	C2
D2	C9	3E	03	BB	C0	36	03	C9	3E	02	BB	C8	3C	BB
C8	36	02	C9	01	C0	03	21	00	D0	CD	2B	D3	7E	FE
00	20	05	CD	4D	D3	18	03	CD	54	D3	23	0D	20	EC
10	EA	C9	21	00	D0	01	C0	03	7E	FE	02	28	04	FE
03	20	02	35	35	23	0D	20	F1	10	EF	C9	CD	D4	D2
CD	5E	D3	CD	7B	D3	C9	21	00	D0	01	C0	03	97	77
23	0D	20	FB	10	F9	C9								

```

10 GO TO 40
20 CLEAR 53000
30 PRINT #4: LOAD "LIFE.mc" CODE
40 RANDOMIZE USR 54173: REM clear growing surface
50 INPUT "How many cells? ";n
60 FOR i=1 TO n
70 PRINT "Cell #";i;"? ";
80 INPUT "Line? ";l,"Column? ";c
90 PRINT l;"",c
100 IF PEEK (53248+l*32+c) THEN GO TO 70
110 POKE 53248+l*32+c,1
120 PRINT AT 0,0;
130 RANDOMIZE USR 54163: REM display & generate
140 IF INKEY#="" THEN GO TO 120

```

# JB CORNER JB CORNER JB CORNER JB CORNER JB

## Spectrum Update by John Bloxham

### NEWS.

Uncle Clive seems to have obtained a financial respite in the guise of Robert Maxwell, a newspaper publisher, who has taken over Sinclair Research for a sum reported to be 12 million pounds. Clive will no longer be the chief executive but head of research, which could be a good move as he seems to be a better inventor than a 'market person'. The Spectrum has been around for three years now and I am sure that the customers are waiting for something new and innovative to appear. Sinclair means brilliant technical novelty, something we have been short of for a while.

### MACHINE CODE BUMPER BUNDLE.

As we seem to have considerable Spectrum/2068 compatability here are a few machine code routines for you to try. This first one is probably the shortest machine code program you will ever see, but it does a big job. The program will help you to make a back up copy of any program of any length. You need two cassette machines, one to load the program in via the ear socket, the other to record the new copy via the mike socket. The program simply outputs to the mike socket whatever comes in the ear socket. It does produce better results than simply connecting two machines together, anyway - try it and see.

```
1 REM Tape to tape copier
10 FOR n=USR "a" TO USR "a"+8
20 READ c: POKE n,c: NEXT n
30 DATA 243,14,254,237,120,237,121,24,250
40 RANDOMIZE USR USR "a"
```

The program is relocatable but for convenience it is loaded into the UDG area. Note that the border colour does not change, and you will have to pull the plug to get out of the program when the copy is made.

The next program is a versatile sideways scrolling routine, I tell a lie, it's two routines in one - left and right (neat, huh?).

```
10 LET addr=23296: REM see text
20 FOR a=addr TO addr+49
30 READ b: POKE a,b: NEXT a
40 DATA 6,192,17,0,64,213,225,35,197,1,
31,0,26,237,176,43,119,0,35,35,19,
193,16,241,201,6,192,17,255,87,213,
225,43,197,1,31,0,26,237,184,35,119,
0,43,43,27,193,16,240,201
```

The routines are 25 bytes long each, the left scroll starting at address `addr` and the right scroll at `addr+25`. After you have carefully typed in the program save a copy to tape then RUN it. To test the routine you need something on the screen so why not LIST the program then enter as a direct command - FOR n=1 TO 100: RANDOMIZE USR `addr`: NEXT n and the display should whip off smartly to the left. Change `addr` to `addr+25` and it should move to the right. Calling the USR scrolls the screen by one character, so you have to call it repeatedly to get this effect. Now the address in line 10 (23296) is the start of the printer buffer, a spare 256 bytes of memory which is very often not used - unless your program contains an LPRINT instruction. Fortunately the routine is fully relocatable. You could change line 10 to

```
10 CLEAR 64999: LET addr=65000
```

and the routine will be safe above RAMTOP. Or you could use the graphics area again by using

```
10 LET addr=USR "a"
```

Just remember to call `addr` to go left, `addr+25` to go right.

Finally, you will have noticed that as the screen scrolls it 'wraps round' - characters that move off one side of the display reappear on the other. You can switch this effect off by doing

```
POKE addr+16,54: POKE addr+41,54. You can now produce a novel sideways CLS by calling the routine 32 times.
```

This next routine is a very useful and powerful programming aid - well worth the few minutes it will take you to type in. It provides the Speccy with a feature found on many other machines - programmable function keys. Very useful during programming, the routine allows you to program any of the alphanumeric keys with lines of BASIC that can be input - or executed - with a single keypress. I normally use the keys 1 to 0 across the top of the keyboard, but see later for how to do it. Firstly enter and RUN the following listing which lowers RAMTOP and pokes the 150 bytes of machine code into place (note that this routine is NOT relocatable).

```
10 CLEAR 65128: LET tot=0
```

```
20 FOR n=65129 TO 65278: READ a: POKE n,a
```

```
30 LET tot=tot+a: NEXT n
```

```
40 DATA 255,243,229,213,197,245,205,120,254,241,193,209,225,251,201,253,203,1,110,200,33,0,0,57,235,237,123,61,92,225,1,127,16,167,237,66,235,249,192,42,83,92,24,2,235,9,35,35,78,35,70,35,84,93,126,254,234,192,35,58,8,92,190,32,235,35,126,254,58,32,229,35,126,254,13,40,223,11,11,11,11,197,229,42,91,92,205,85,22,19,237,83,91,92,35,235,225,193,237,176,235,43,126,254,35,40,8,205,29,17,253,203,1,174,201,1,1,0,205,232,25,62,13,50,8,92,253,203,1,238,201,62,9,237,71,237,94,201,62,62,237,71,237,86,201,0,0,0,0,0,
```

```
50 IF tot<>18140 THEN PRINT "DATA ERROR": BEEP .5,0
```

If the above program runs okay you can delete the BASIC with NEW - the machine code will be safe above RAMTOP. Then type in the loader/instruction screen program.

```
5 CLEAR 65128: CLS : PRINT "LEAVE TAPE RUNNING": BEEP .4,20: LOAD ""CODE
```

```
10 CLS : PRINT "PROGRAMMABLE FUNCTION KEYS"
```

```
20 PRINT "" Any key can be programmed to produce other characters or routines."" Set up the functions with REMsat the start of the program."" E.g.: 1 REM (key):(function)"
```

```
30 PRINT ""1 REM !: BORDER 7: PAPER 7: INK 0: BRIGHT 0#"
```

```

40 PRINT " This will reset the colours on pressing ! (symbol shift/1). The# i
ndicates a command to be carried out i.e. an 'ENTER'."
50 PRINT " The routine is switched:" "ON - RANDOMIZE USR 65260" "OFF - RANDOM
IZE USR 65267" "Press any key to delete BASIC."
60 RANDOMIZE USR 65260
70 PAUSE 0
80 NEW
9999 SAVE "Func keys" LINE 1: SAVE "mc"CODE 65129,150

```

Note that the contents of the PRINT lines are spaced to produce a neat output on the 32 column screen. When you are satisfied that all is well save both BASIC and CODE to tape by doing GO TO 9999.

Your function keys are set up in REM lines at the start of the program you are typing in. Clear out the BASIC then do RANDOMISE USR 65260 to switch the routine on. The basic format of the function is (line number) REM (key):(function) So, type in

```

1 REM !: BORDER 7: PAPER 7: INK 0: BRIGHT 0 After you have
entered this line whenever you press ! (symbol shift 1) those
words will appear in your input line. Remember that whatever
comes after the first colon is the function programmed onto the
key before the colon. EDIT down line 1 and add a "#" at the end
after BRIGHT 0. The"#" acts like a ENTER command and causes the
function to be carried out. Now when you press ! the words do
not appear but the colours are reset immediately. Now enter this
line

```

```

2 REM ,: Note that there is a space before the colon and a
comma after it. Now, whenever you press the space bar a comma
appears, very useful when your are entering in lots of DATA - as
in the program above! Try

```

```

3 REM 0: FOR n=1 TO 20: BEEP .01,n: NEXT n# Now, whenever you
press the zero your computer gives a little trill! Note that
only REM lines at the start of your program will have this
effect. If you want to do a normal REM i.e. one that doesn't
program a key then just place a "*" after the REM, like 5 REM
*normal rem - and it will have no effect. You can switch off a
programming REM line by entering a "*", moving it further down
the listing, or doing RANDOMIZE USR 65267 which switches the
whole routine off. This routine is worth experimenting with,
remember that you can program almost any sort of routine -
renumber,screen save, memory printout, printer routine etc. -
onto any key.

```

I hope these programs work for you and that you find them worthwhile.