MICRO-80

P.O. BOX 213, GOODWOOD, S.A. 5034 AUSTRALIA
TELEPHONE: (08) 272 0966

ISSUE 8
JULY 1980

PRICE AUS. $2.50
NZ. $3.00

***** CONTENTS *****

		PAGE
ABOUT MICRO-80		1
EDITORIAL		2
TANDY LEVEL II & SYSTEM 80 COMPATIBILITY		6
'80 USERS GROUPS		7
ASSEMBLY LANGUAGE PROGRAMMING - PART 5		8
BETTER BYTES		14
INPUT/OUTPUT (Letters to the Editor)		17
RADIO SHACK CASINO GAMES PACK - A Review		19
HARDWARE SECTION		
- TANDY XRX III MODIFICATION		20
SOFTWARE SECTION:		
- CATCH	L1/4K	22
- FUEL ECONOMY	L1/4K	25
- INTEREST & LOAN REPAYMENT CALCULATOR	L2/16K	26
- EDITORS/ASSEMBLERS, MONITORS etc. (a brief explanation)		27
- SYMBOL/BAS	L2/16K	27
- SOUND GENERATION WITH THE '80		35
* SOUND EFFECTS	L2/16K/4K	36
* SOUND	L2/16K	42
NEXT MONTH'S ISSUE		44
CASSETTE EDITION INDEX		45
ORDER FORM		45

MICRO-80 is registered for posting as a publication - CATEGORY B

Printed by:- The Shovel and Bull Printers,
312A Unley Rd., Hyde Park, SA5061.

All enquiries should be directed to:-
MICRO-80 P.O. BOX 231, GOODWOOD, SA 5034.

Published by MICRO-80, 284 Goodwood Rd., Clarence Park, ADELAIDE
Telephone (08) 272 0966

MICRO-80

** ABOUT MICRO-8Ø **

EDITOR: Ian Vagg

ASSOCIATE EDITORS:

SOFTWARE: Peter Hartley

HARDWARE: Edwin Paay

MICRO-8Ø is the only Australian monthly magazine devoted entirely to the Tandy TRS-8Ø microcomputer and the Dick Smith System 8Ø. It is available by subscription, $24.00 for 12 months or by mail order at $2.50 per copy. A cassette containing all the programs in each month's issue is available for an additional $3.50 or a combined annual subscription to both magazine and cassette is available for $60.00. Special bulk purchase rates are also available to computer shops etc. Please use the form in this issue to order your copy or subscription.

The purpose of MICRO-8Ø is to publish software and other information to help you get the most from your TRS-8Ø or System 8Ø and their peripherals. MICRO-8Ø is in no way connected with either the Tandy or Dick Smith organisations.

** WE WILL PAY YOU TO PUBLISH YOUR PROGRAMS **

Most of the information we publish is provided by our readers, to whom we pay royalties. An application form containing full details of how you can use your TRS-8Ø or System 8Ø to earn someextra income is included in every issue.

** CONTENT **

Each month we publish at least one applications program in Level I BASIC, one in Level 2 BASIC and one in DISK BASIC (or disk compatible Level 2). We also publish Utility programs in Level 2 BASIC and Machine Language. At least every second issue has an article on hardware modifications or a constructional article for a useful peripheral. In addition, we run articles on programming techniques both in Assembly Language and BASIC and we print letters to the Editor and new product reviews.

** COPYRIGHT **

** LIABILITY **

***** EDITORIAL *****

Many of our recent subscribers are owners of new System 80's. We welcome you to our circle and assure you that we will support the System 80 just as enthusiastically as we have supported the TRS-80 for the past seven months. Elsewhere in this issue, you will see evidence of our support - for example, Peter Hartley has written some programming guidelines for our readers and authors to help ensure that our published programs are fully compatible between the two machines.

Some of our System 80 owners have complained that they cannot load the programs on side 1 of our Free Software tape. These programs are written for TRS-80 Level I machines which use a different BASIC interpreter and record programs at only half the rate of Level II machines (the System 80 is a Level II machine). Therefore, you cannot load Level I tapes directly into Level II machines.

If you wish to do so, there are three alternatives:

- obtain a copy of the Tandy Level I to Level II conversion tape (this, however, has some bugs in it and can entail a lot of program rewriting).
- buy one of the Level I in Level II programs available on the American market. These programs temporarily convert your machine to Level I. MICRO-80 Products has some on order, so they will shortly be available here, too.
- instal a Level I ROM in the Level II machine with a switch, as described in last month's issue. This alternative is really not available to System 80 owners as it would be a major job involving an additional specially designed, PC board.

** PRESENTATION AND PUBLICATION DATES **

No doubt you will have observed the changed presentation of MICRO-80 which commenced in June. This was rather forced upon us when the editorial CPU ceased to function with disk drives as we were half way through preparing the June issue. We have fixed it up now but, in the meantime, decided to revert to an old-fashioned electric typewriter to produce the magazine. The text is certainly more readable, judging from the comments we have had from a number of readers, despite the lack of right justification. We have therefore decided to stick with this format for the foreseeable future, but look for a further improvement in a month or two when one of the hush-hush projects currently being worked on by MICRO-80 PRODUCTS, comes to fruition.

The main casualty of the CPU failure was the June publication date which actually stepped back into July. This, the July issue is being mailed out in early August. We hope you will understand and we assure you that we are doing our best to recover the lost time.

** WIDER DISTRIBUTION OF MICRO-80 **

Those readers who prefer to buy single copies of MICRO-80 will find the July issue in the following outlets:

Melbourne:	McGill's Newsagency 187 Elizabeth St.
Sydney:	C.I.S.A. 159 Kent St.
Brisbane:	Software-80 Shop 11 - 198 Moggill Rd. Taringa. 4068.
All States:	Dick Smith Stores.

** ABOUT LETTERS TO THE EDITOR **

Two of the more important functions of MICRO-80 are to act as a forum for readers' ideas and to assist readers who have specific problems. These we do by publishing and answering letters to the Editor. Unfortunately, we are becoming snowed under with letters. Before we get to the stage of one of the American magazines which recently announced that it had a backlog of 7 months of readers' letters, we will try to improve the situation by taking the following steps.

- we will maximise the use of the space available for publishing replies to letters by abbreviating the letter itself as much as possible.
- those letters which entail requests for articles or programs for future issues of MICRO-80 will not be published separately but the request will go straight into the Readers' request column (to be reintroduced next month).
- when we get a considerable number of letters about one subject we will prepare an article specifically on that subject and will not acknowledge the individual letters.

However, all that still leaves us with a problem area. A proportion of letters are from readers with specific problems which require detailed answers. Usually, these problems are annoying to the individual and may be preventing him from doing something important, so their solution is required urgently.

The only way we can think of handling this situation is to farm the replies out to people we know who have the technical background and experience to solve such problems. But such people usually earn part or all of their living solving such problems so they need to be paid. If you have a problem, and are willing to pay for its solution, then tell us so when you write in and we will find the appropriate person to deal with it. Send a cheque for $4.00 with your letter for which you can expect to obtain a full half hour of attention to your specific request. If you require more assistance than that, such as the rewriting of significant parts of a program, then whoever it is that is dealing with your request will contact you directly to negotiate a reasonable price. To ease administration, you should make your initial cheque payable to MICRO-80 but we are simply acting as a clearing house in this matter. The whole amount will be passed on to the person selected to solve your problem.

We hope these actions will result in a better and faster response to your letters.

** LEVEL II + I MODIFICATION **

Last month, in our article describing how to instal a Level I ROM in a Level II machine, we did not make it clear that this modification only applied to CPU's containing either A or D boards. To ascertain whether your CPU is one of these, look at Fig. 3 on Page 25 of the June issue. At the top-centre of the PC board, you will see printed "TRS-80" "1700069.D". The letter at the end of the number, in this case D, indicates which version PC board you have. Unless this is an A or a D, you should not carry out the conversion described. We have now successfully converted a "G" board and will describe the conversion for that, next month.

If you have a "G" board and have tried to carry out the conversion, this is what you should do while you are waiting for next month's issue:

- leave the switch wiring in place - that is the same for the G board as for the A and D boards.
- Repair the PC track you cut in accordance with step 5 on Page 28 of the June issue.
- remove the Level I ROM from Z34 socket and put it somewhere safe.

Your machine will now be back to Level II only but you will have carried out 80% of the work required to convert to Level I & II.

In the meantime, if any of our readers have carried out conversions on other than A, D or G boards, we would be interested in publishing the results of their work.

** THE EXATRON STRINGY FLOPPY (ESF) **

Many MICRO-80 readers have bought ESF's and, from all accounts, are very happy with them. MICRO-80 intend to support this peripheral and Charlie Bartlett has agreed to become the Editor of an ESF Forum, to be published in each issue of MICRO-80. Charlie will give hints and tips on getting the most from your ESF. Next month, for instance, he will explain how to increase the effective size of your memory by chaining BASIC programs. Charlie will be happy to publish letters, programs, hints, tips, etc. from ESF owners so, as you discover new ways to use this excellent product, write in to us and we will pass your letter on to Charlie.

** PROGRAMS SUBMITTED FOR PUBLICATION **

As some readers who have submitted programs for publication will know, we have sometimes been a bit slow in contacting them concerning the fate of their programs. This has proven to be a difficult area administratively, and we are busy streamlining our procedures. With the number of programs being submitted it is still going to take us at least a month, perhaps longer, to really put a program through its paces, but what we will do within 10 days of receiving your program is to make a backup copy and return your original cassette or disk so you know that we have received your correspondence. Thereafter, we will consider it in detail just as quickly as we can.

** THE TRS-80 AND TI's SPEAK & SPELL **

Several readers have contacted us regarding items they have seen in the American press announcing a board which interfaces the TRS-80 to a Texas Instruments Speak & Spell machine. These boards, called "You Speak to me, Too" are produced by Percom Data Corporation and we have ordered a small quantity for evaluation, assuming the editorial offspring allow me to use their Speak & Spell ook for a review of this product in a month or two's time.

We have also ordered some data separator boards the U.S. press has said are a must for any business application involving disk drives. Apparently, the manufacturers' specification for the FD 1771 Floppy Disk Controller chip used in the expansion interfacs recommends the use of a data separator but Tandy did not follow this recommendation - hence the relatively high incidence of data I/O errors. A data separator board which could sell in this country for around $39.00 is supposed to virtually eliminate such errors. Look for a review on this product too, in the near future.

** SCOOP - NEW PRODUCTS FROM TANDY **

We have it on good authority that, about October, Tandy will release two new products in Australia. The first is a hand- held, wallet-sized microcomputer with an alpha-numeric display, able to be programmed in BASIC!! It will have 1.9K of user RAM and will sell for less than $250. There will be an optional cassette interface available for about $90 or a compatible cassette recorder for about $140.

The other product is called "classroom controller" (shades of 1984?). It will enable a number of "slave" TRS-80's to be connected to one "master" TRS-80, controlled by the teacher. Software in the "master" unit will allow the teacher to select which of the "slave units he (she) wishes to communicate with at any time. Price $699.

** MARKET PLACE **

Don't forget our Market Place announced in the June issue if you have any hardware items to dispose of. Simply send in an advertisement up to 50 words in length. On successful sale, you pay us a commission of $5 or 5% of the advertised price, whichever is the higher, up to a maximum of $30.

** SYSTEM 80 DEVELOPMENTS **

Jamieson Rowe, Dick Smith's Technical Director, has prepared Technical Bulletin No. 12 which explains how System 80 owners may fit a "Forward Arrow" key to their machines and also how to fit a switch which will allow optional transfer of all cassette functions to an external recorder. This would enable machine language programs as well as BASIC programs to be loaded from an external recorder having an adjustable volume control. We understand that this technical bulletin is available for the asking from any Dick Smith store.

- ooooo -

** TANDY LEVEL II & SYSTEM 80 COMPATIBILITY **
** NOTICE to all prospective authors **

The policy of MICRO-80 is to support both the Tandy and the Dick Smith machines. There are areas where these two Level II systems are not compatible, and authors will have to observe certain rules if their software is to be considered for publication by MICRO-80.

1. If the CHR$(23) function is used, the screen display must look correct in both TANDY LARGE (32 CHARACTER) MODE, and in SYSTEM 80 NORMAL SIZE BUT DOUBLE SPACED MODE.

2. The right arrow key is not to be used.

3. The CLEAR key is not to be used.

4. In games, for left and right movement INPUTS, the "<" and ">" will be adopted for on-screen instructions, and the characters "," and "." will be used in-program.

5. In games, for up and down movement INPUTS, the up arrow (CNTRL on the System 80) and the down arrow (ESCP) will be used in-program and, for on-screen instructions, these keys will need to be described in words, as the symbols for the up and down arrows are absent from the character set of the System 80.

6. In more serious applications, the right TAB function is available on the System 80, by using the <SHIFT><CNTRL> and <I> keys simultaneously. Any on-screen instructions should clarify this if the function is required. It would be better to try to write around this.

7. Programs using lower-case modified displays are unsuited to the System 80, which uses a different, smaller character generator.

8. Because of uncertainties regarding future proposed modifications to the System 80 (if any) we reserve the right to amend these requirements in the future.

- 00000 -

***** '80 USERS GROUPS *****

The following is a list of '80 Users groups. If you have a group that is not included here, please let us know about it so we can publish details. Owners of System 80's are welcome at all the groups.

BRISBANE:
This is a new group which is just getting started. Enquiries to:
Mr. Lance Lawes, 21 Rodney St. Lindum, 4178, Qld.
Telephone;- home (07) 396 2998; bus. (07) 268 1191 Ext.15

MELBOURNE EASTERN-SUBURBS:

1st Wednesday of the month	Tandy Store, 96 Koornang Road, Carnegie.
3rd Wednesday of the month	Kingswood College 355 Station St., Box Hill
CONTACT:-	John Fletcher, 89 0677 between 9 - 4.

DARWIN:	CONTACT:-
CONTACT:-	Tony Domigan, P.O. BOX 39086, Winnellie, NT. 5789

ADELAIDE:

1st Thursday of the month	TANDY City Store, 94 Gawler Place, Adelaide
CONTACT:-	Rod Stevens, 515241 between 9 - 4.

***** ASSEMBLY LANGUAGE PROGRAMMING - PART 5 *****

by EDWIN R. PAAY

This month we will deal with branching and subroutines. First we will discuss branching.

* * * BRANCHING * * *

There are two different ways that program execution can be made to branch to different locations. They are the jump (JP) instruction and the relative jump (JR) instruction. The jump instruction is equivalent to a GOTO statement in BASIC. It transfers program flow to whatever address is specified after the jump command, the instruction 1ØØ JP 7ØØØH for instance will simply transfer control to the code at 7ØØØ hex. This is the only type of jump available on some microprocessors, notably the 8Ø8Ø. The Z8Ø however can also perform relative jumps. The relative jump instruction is followed by a two's complement displacement value which is one byte long. This means that the range of a relative jump is from -126 to +129. The benefits of relative jumps are:

(1) A relative jump is only two bytes long as opposed to three bytes for a normal jump.

(2) A program containing relative jumps is relocatable. This means that the program can run in any memory area because the relative jump doesn't point to a memory location directly, but indirectly, through its displacement byte.

The actual format of the displacement byte needs some more discussion. The value of the displacement is actually not the displacement counting from the location of the address containing the relative jump instruction but is referenced to the address of the next instruction in memory following the relative jump. The relative jump is two bytes long,therefore, the displacement value is the displacement from the address of the relative jump minus two. This is because when an instruction is fetched from memory by the CPU the program counter (PC) is incremented to point to the next instruction before execution takes place, therefore, since the displacement is added to the current program counter value to compute the jump address, the displacement must be altered to allow for this. All that sounds very tedious doesn't it? Well, here is some good news. The assembler does all this for us and the actual displacement will never have to be calculated by the programmer. A relative jump might be used like this :

```
1ØØØ LOOP   ADD    A,C
11ØØ        LD     B,A
12ØØ        JR     LOOP
```

All that is required is to specify a label or an address and the assembler will calculate the correct displacement and insert it in the right place without us having to worry about it.

This brings us to conditional branching. In some instances we will want to make a jump only if certain conditions are met. In BASIC, for instance, we have the IF THEN statement and in FORTRAN we have an IF statement to do this. With machine language we use the flags to achieve exactly the same result. (If necessary review the discussion on flags in last month's issue to refresh your memory). Let me give an example.

```
1ØØØ     CALL   INPUT         ;INPUT A BYTE INTO THE "A" REG.
1Ø1Ø     CP     3             ;COMPARE WITH 3.
1Ø2Ø     JR     Z,EQUAL       ;JUMP IF A = 3.
1Ø3Ø     JR     C,LESS        ;JUMP IF A < 3.
1Ø4Ø     JR     NC,GREATR     ;JUMP IF EQUAL OR GREATER THAN.
```

The program above CALLs a routine labelled INPUT which inputs a byte into the A register. Line 1Ø1Ø will then compare the value with the number 3. If the A register contains 3 then line 1Ø2Ø will transfer control to a fictitious location labelled EQUAL, if the A register contains a number less than 3 then the carry flag will have been set (the carry flag is set if a borrow is generated) and therefore line 1Ø3Ø would force a jump to location "LESS" if that was the case. Line 1Ø4Ø is not really needed but it demonstrates what a line causing a jump on an equal or greater than basis looks like. Note that relative jumps were used in the example above. If normal jumps had been used the only difference in the source code would be to replace the JR's with JP's.

Register indirect addressing can be used for jumps too. For instance : 23ØØ JP (HL) will cause a jump to the location pointed to by the HL register pair. This can be put to good use in a program which inputs an address from the keyboard and converts it to Hex, then places it in the HL register pair and executes a JP (HL) , this would allow the user to jump to any location in memory.

***** SUBROUTINES *****

Subroutines are possible through the CALL instruction. A CALL is equivalent to a GOSUB statement in BASIC, the RET is the same as the RETURN statement in BASIC.

```
1ØØ          CALL       SUB
11Ø          JP         DONE
12Ø SUB      LD         HL, 3E3H
13Ø          XOR        A
14Ø          LD         (HL),A
15Ø          RET
```

The program above shows how a subroutine might be used. Apart from the unconditional CALLs there are also conditional CALLs and RETurns.

```
1100        CALL      C,INPUT
1234        RET       Z
0035        CALL      NZ,OUT
0010        RET       NC
```

All the lines above are valid, line 1100 will CALL a subroutine if the carry flag is set, line 1234 will RETurn from a subroutine if the Z flag is set etc.

This brings us to the stack. When a CALL is executed the return address will be pushed on the stack. The situation is reversed with a RETurn instruction, a value is POPped from the stack and placed in the Program Counter. It is important therefore to make sure that the stack is at the same level when a RET is executed, as it was when the CALL was executed.

Then there is the restart (RST) instruction, this is in effect a CALL, and a RETurn instruction can be used to return from it. The only difference is that a RST is only one byte long as compared to three for a normal CALL. The RST is only capable of going to eight different locations in low memory :

```
100      RST   0H
200      RST   8H
300      RST   10H
400      RST   18H
500      RST   20H
600      RST   28H
700      RST   30H
800      RST   38H
```

The restart in line 100 calls address 0000H, line 500 calls 0020H etc. The locations are in ROM of course and the TRS-80 makes good use of them. The ROM REFERENCE MANUAL gives a complete explanation of these and many other ROM routines.

***** ROTATE AND SHIFT GROUP *****

These instructions are used for some arithmetic functions and for bit testing. In the case of microprocessors with a limited instruction set, the rotate instruction can be used to test if a certain bit is set or reset, by rotating the required number of bits around and testing the appropriate flag to see if the bit was 0 or 1. The Z80 however, has complete bit test facilities and we do not have to resort to such tactics. As far as the shift function is concerned, each time a right shift takes place the number operated on is effectively divided by two while a left shift will multiply by two. This feature is used to avoid having to do many repeated additions when multiplying.

This is all I want to say about these functions at this point in time, my aim is to get you programming in assembly language and memorizing all the rotate and shift functions is futile. All that is required is to know that these functions exist then when the need arises simply look them up in your ZILOG assembly language manual.

***** BIT SET, RESET AND TEST GROUP *****

As I mentioned above, the Z8Ø has bit manipulation logic built in. This is one of the functions that sets the Z8Ø aside from other, less powerful microprocessors (like the 65Ø2 used by the Pet and the Apple micro computers).
The instructions in this group set, reset, or test one of the eight bits in A, B, C, D, E, H OR L registers in the CPU or in memory.

```
1ØØØ        BIT     3,A         ;TEST BIT 3 IN REG. "A"
1Ø1Ø        SET     5,(HL)      ;SET BIT 5 IN MEM. POINTED TO BY HL
1Ø2Ø        RES     Ø,H         ;RESET BIT Ø IN REG. "H"
```

Line 1ØØØ tests the fourth bit (bit number 3) for zero or one. If bit 3 is zero the Z flag will be set and if one the Z flag will be reset. Line 1Ø1Ø simply sets bit 5 in the memory location pointed to by the HL register pair to "1" and line 1Ø2Ø resets bit Ø in the H register.

***** INPUT/OUTPUT GROUP *****

Lastly, we have the I/O group. These instructions are used for I/O to the Z8Ø ports. There is not much that I will say about these because they are exactly the same as the INP(X) and OUT X,Y statements in BASIC.

```
34ØØ  OUT (Ø FEH),A
56ØØ  IN  A,(25H)
```

These are two typical I/O instructions line 34ØØ outputs the contents of the A register to port FE Hexadecimal (254) whilst line 56ØØ inputs a byte from port 25 hexadecimal (37) to register A.

This brings us to the end of our discussion of the Z8Ø instruction set. The material presented so far in this series should allow anyone to write assembly language programs, simple ones at first and more complicated ones later, as all that is needed now is practice. In the next part of this series I will present small programs involving common problems arising with assembly language programs. To prepare for this you could try to write an assembly language program to input hexadecimal addresses from the keyboard, display them, find the content of the address specified and display it to the right of the address on the screen. For those of you who find that easy, try to write an assembly language program to accept hexadecimal numbers, convert them to decimal and print the result to the screen.

As you will remember, last month I left you with a problem. The problem was to write an assembly language program which will allow us to store the image on the screen and to restore it at will. I will list the program below and then explain each line.

***** SAMPLE PROGRAM *****

```
ØØ2ØØ                ORG     7ØØØH
ØØ3ØØ   VIDEO        EQU     3CØØH
ØØ4ØØ   START:       CALL    ØA7FH            ;GET FUNCTION CODE
ØØ45Ø                LD      A,L              ;   PUT IT IN "A"
ØØ5ØØ                EXX                      ;SAVE REGISTERS
ØØ6ØØ                CP      2
ØØ7ØØ                JR      Z,USR2           ;IF USR(2) GOTO USR2
ØØ8ØØ                LD      HL,VIDEO
ØØ9ØØ                LD      DE,BUFR          ;INITIALIZE REGISTERS
Ø1ØØØ   CONT:        LD      BC,4ØØH
Ø11ØØ                LDIR                     ;STORE VIDEO IMAGE
Ø12ØØ                EXX                      ;RESTORE REGISTERS
Ø13ØØ                RET                      ;BACK TO BASIC
Ø14ØØ   USR2:        LD      HL,BUFR
Ø15ØØ                LD      DE,VIDEO
Ø16ØØ                JR      CONT             ;RESTORE VIDEO IMAGE
Ø17ØØ   BUFR:        DEFS    4ØØH             ;STORE IMAGE HERE
Ø18ØØ                END
```

This program has been written for use by a USR call in BASIC. If the command USR(Ø) is executed, the image on the screen is stored in a buffer, else if the command USR(2) is executed then the image is restored to the screen from the buffer.

LINE NUMBER	EXPLANATION
2ØØ	This tells the assembler that the object code is to be placed at 7ØØØ hexadecimal onward.
3ØØ	This sets label VIDEO to 3CØØH. It is good practice in assembly language to use labels whenever you can as it makes the program more readable and if it is necessary to change the value of a key address it is easier to change one EQU statement than it is to change many Hexadecimal values. 3CØØH is the start of video memory.
4ØØ	A CALL ØA7FH will pass the value from the USR statement and place it in the HL register pair. This is explained in the level II manual.
45Ø	This instruction loads the value passed from BASIC into the A register where it can be operated on.
5ØØ	By using the alternate register set we effectively save the registers for BASIC.
6ØØ	This line tests the value passed from BASIC to see if it is to store or recall the image.

7ØØ	If it is USR(2) then go and recall the previous image to the screen.
8ØØ	Load the HL register pair with the start of video memory.
9ØØ	Load the DE register pair with the BUFfeR address.
1ØØØ	Load the BC register pair with the length of video memory.
11ØØ	Move a block of memory from HL into DE of length BC.
12ØØ	Restores the original registers.
13ØØ	Returns to BASIC.
14ØØ	Entry for the recallimage function. (USR(2))
15ØØ	Same function as 8ØØ and 9ØØ above.
16ØØ	This jump simply makes use of some code that already exists earlier on in the program and avoids duplication of instructions already ther
17ØØ	This is DEFined to be Storage for the video image.
18ØØ	That's all folks.

If this is used as a subroutine in a game program then all that is required is tc execute a USR(Ø) after every move made and to provide the game with an extra command which would invoke a USR(2) to restore the picture to the screen.

The program below contains the machine language subroutine in DATA statements and shows how the subroutine is used.

```
2 FOR X=28672 TO 28702: READY: POKEX,Y: NEXT
4 DATA 205,127,10,125,217,254,2,40,13,33,0,60,17,30,112,1,0,4
6 DATA 237,176,217,201,33,30,112,17,0,60,24,241,0
10 CLS
15 FOR X=0 TO 127: SET(X,23): SET(X,0): SET(X,47): NEXT
20 FOR Y=0TO47: SET(64,Y): SET(0,Y): SET(127,Y): NEXT
25 Y=0: FORX=0TO127: SET(X,Y): Y=Y+.370079: NEXT
30 Y=47: FORX=1TO127: SET(X,Y): Y=Y-.370079: NEXT
100 POKE16526,0: POKE16527,112
200 X=USR(0) :'store display.
300 CLS: FORX=0TO100: NEXT
400 X=USR(2): FORX=0TO100: NEXT:'restore display and pause.
500 GOTO300
600 END
```

Lines 2 to 6 will put the machine language program in place.
Lines 15 to 30 will draw a picture on the screen.
Line 1ØØ initializes the USR vector.
Lines 3ØØ to 5ØØ will cause the picture to be flashed on and off at intervals determined by the FOR NEXT loops. This will demonstrate the speed of the program

** BETTER BYTES ** a wander through the jungle of bytes, nybbles and bits, with Peter Hartley.
(nybble = half a byte - No kidding!)

Some of the tapes I've been sent recently have been most pleasing in their content. But too often their recording quality has been greatly lacking. So...HOW TO GET THE BEST FROM YOUR CASSETTE DECK.

1. DO clean the deck at least once a week - ideally after every two hours of use. Computer usage is much more fussy than audio.
2. DON'T use so-called cleaning tapes - they'll wear out the heads faster than they'll clean 'em.
3. DO use cotton-buds for your cleaning.
4. DON'T use metho - save it for a nightcap. DO use ISOPROPYL ALCOHOL.
5. DO remember to clean the capstan and pinch roller.
6. DO demagnetise the unit at least once a month - residual fields in the heads can wipe as much as 3 to 5 db off the top of a recording the very first time the tape is played!

Now comes the really nasty business of HEAD ALIGNMENT. The head should be aligned in numerous planes, but the decks supplied with the '8Ø are only capable of alignment in one plane - the most critical.
If you have really good hearing, you can align a head by ear, playing a known good tape - such as those supplied by MICRO-8Ø - and adjusting the head for the best treble response. However, it is far better to use a multimeter... Connect the multimeter across the loudspeaker terminals, set the volume to MINIMUM, load the good tape, press PLAY, sit the deck right way up and locate the tiny hole or rebate above the head that lines up with the little adjusting screw beside the head.

Now go find a screwdriver that fits!

Turn the volume up just enough for some '8Ø type noises to issue forth, and select a meter scale that gives you a vaguely mid-scale reading. Now, gently rotate the adjusting screw back and forth until you locate the one position that gives the highest possible reading on the meter.

*** WARNING *** NEVER PRESS THE PLAY OR STOP BUTTONS WHILE THE SCREWDRIVER IS INSERTED - I'll instal new heads for a fee!

Now for some general advice on tape use.
Even with a leaderless tape, always leave a few inches at the start before the dump starts. It isn't possible to erase that first couple of inches without using a bulk eraser.

Don't use the tape that had last week's Big Band Concert on it without bulk erasing it first.
Don't record on batteries. You were given a mains lead - so use it. There are some very nasty tapes around. Pay a few cents extra and get a good one.
Don't use C9Ø or C12Ø tapes - they are too heavy for some decks, and are very prone to damage while running.
For some strange reason, the first dump of the day always takes a long time to come out at the cassette end, so waste a dump, with no tape in the deck. (Press the little douvralackie at the back while you press PLAY and RECORD).
In Level II, use the CLOAD? facility to check the dump, rather than making three or four, but always make two saves just in case.
Level I is more reliable than Level II, so two dumps should be adequate if you follow all the above guidelines.

Well, that's the sum of my contribution in this arena, but any tips from readers will be appreciated and passed on.

WARP 8 YOUR DOS!

An unlikely title for this month's most cheery topic. It's an amazing thing, having been tied to a cassette system for so very long, that having now graduated to Disk, I still get frustrated with the delays of I/O.

Herewith - a fix, which is proven on my own Dick Smith Drives, and which will probably work with all but the very oldest of worn drives, except that I'm advised that it only works with very new Shugarts, so please, no angry letters if it doesn't work on YOUR drive!
Gloriously, itis a software fix, which can be organised by anyone with access to a copy of NEWDOS PLUS or a presumably pirated copy of SUPERZAP.

The fix involves changing the values stuffed into the memory-mapped I/O location 37ECH. This location carries the delay values used by the Disc Controller to "slow down the system" while the drives move from one track to another.

First make a back-up "System" disk. If you are unlucky enough to find that the fix doesn't work for you, the result will be garbage written all over the disk in the wrong places. Better to test it before your only good disk gets zapped.
Now load and run SUPERZAP. Where the change is to be made will depend on the DOS you are using, as will the values to be altered.

NEWDOS Track Ø, sector 8, byte FE
or sector 9, byte ØF
You are looking for a 4 byte sequence IBF5.
Change this to 1AF5

2.1 TRSDOS Track Ø, sector 7, byte 52
Change 1BF5 to 1AF5

2.2 TRSDOS Track Ø, sector 7, byte 52
Change the 1F to 1E **not a misprint **

2.3 TRSDOS same as for 2.2

Not only does this fix actually work, but I have found it psychologically pleasing too! The read/write is noticeably reduced, as is the "suspense-time" at those moments when the read/write head goes careering back and forth like a lunatic sheepdog. In fact, I have experienced no losses of data since making these changes, and whenever a parity error has been detected by DOS I haven't even had time to get my podgy pinkies crossed before the sector was reprocessed and smooth operation resumed.

One word of warning. After making these changes, run some SAVE and LOAD tests. There are two situations which will tell you that this modification isn't for YOUR drives.

1. Faulty I/O operation.
2. Metallic clanking noises from the drive.

In these situations there isn't much that you can do except re-format the test disk and pretend that you never read this piece of MICRO-8Ø.

If you don't have a copy of NEWDOS PLUS, do yourself a favour and buy one for yourself. Absolutely BEWDY! If you don't know how to drive SUPERZAP, that now famous TRS-8Ø DOS & OTHER MYSTERIES will do you another favour.

** INPUT#A$?????????

Now, it doesn't need much of an expert to see that the title instruction wouldn't work. Only one problem - it does! In Level I, anyway. To save one entire screen of text, simply load MICRO-MUSIC, and type. Save to cassette as required. The pages of data can then be recovered from Level I BASIC with "I.#A$". Absolutely amazing. Even more puzzling, if you BREAK after the load is completed, try PRINTA$. You'll be greeted with the line number of THAT instruction.

***** INPUT/OUTPUT *****

From: Ron Kehn, 27 Guys Rd., Korumburra, Vic. 3950.

Will the lower case modification kit have any effect on storing programs on disk?

I would like to connect my TRS-80 to an Apple 2 via an RS 232 interface. If anyone has attempted this or has any comments to make about the feasibility of exchanging programs between the two (given differences in graphics and prints, etc) I would really appreciate hearing from them.

(No, Ron, the lower-case mod kit will not affect the storing of programs on disk in any way whatsoever.

It should certainly be possible to transfer information between the two computers using RS 232 interfaces. Whether or not you could get useful programs to run on both, I cannot say. Perhaps some of our readers could help you, as you requested. Ed.)

From: George Glendinning, P.O. Box 1004, Mackay, Qld. 4740.

I am an amateur radio operator (call sign VK4vcz). I would like to hear from any amateur who operates a computer net. I am usually on air on 3.620 MHz on Sundays at 9 p.m.

(Hope someone out there shares similar interests, George - Ed.)

From: Norm Partridge (by telephone)

Lines 1640 and 1645 of the WORDPROC program in the December issue don't seem to make sense. Also, line 550 in the AMAZIN program in the February issue seems to be a bad list.

(How right you are Norm, on both counts. We are amazed that no-one has queried us on the WORDPROC program before. The correct listings for these lines are shown below.

```
    1640  IF LEN (A$ (N3) ) = 0 THEN 1650
    1645 PRINT "LINE"; N3; "ALREADY OCCUPIED." : GOTO 200
```

We have been queried on the AMAZIN program before but have answered the queries to individuals rather than to readers in general. The last statement on line 550 was cut off in its prime in some (all?) copies of the magazine. Instead of GOTO 5 it should have read GOTO 57Ø

From: Frank Ellett, Palm Beach, Qld.

I have a Level II 32K machine. The 16K version of FILES published in the February issue works well but the 32K version does not. The program returns a "CHECKSUM ERROR" and if I type "PRINT B" and enter that number, the computer freezes up. Also, why do you protect high memory when the program is at 42EF?

(The 32K version of FILES was written for Disk BASIC users. Disk BASIC starts at 6A24H and that is about where the machine language program expects to be. If you enter it into a Level II machine, it is at the wrong address and causes the program to crash. A number of other readers have had the same problem so we will publish a 32K and 48K Level II version in a future issue. As far as protecting high memory is concerned, the first thing the machine language routine does is to move itself up to the top of memory, hence the need to protect it - Ed.)

From: E.B. Lindsay, 6 Hillcrest Avenue, Faulconbridge, N.S.W. 2776.

I have an IBM Model 72 input/output writer that has been converted to work with a Monroe 300 calculator. Does anyone have information on this conversion so that I can convert it to work with my System 80?

(If any reader has such information, perhaps they would contact Mr. Lindsay directly - Ed.)

From: George Skarbeck, Glen Waverley, Vic.

I have discovered a small bug in BMON's RENUMBER routine. If there is a line Ø in the program, then the renumbering will not work properly. It inserts your first line number after every ELSE & THEN. I would like to see a program that will print the contents of the screen, just by pressing several keys, but does not need DOS.

(thank you George, we have tried this out and you are correct about BMON's inability to renumber programs starting with a line Ø. On occasion we have noticed that the BASIC interpreter itself has problems with line Ø and usually avoid using it. We do not consider this bug is serious enough to warrant reworking BMON so advise all BMON users not to use line Ø in their programs. We have put your suggestion into Readers' Requests - Ed.)

From: Mr. B. Bussenschutt, Highbury, S.A.

Mr. Bussenschutt made a number of constructive suggestions concerning layout and presentation of the magazine, most of which have now been implemented. He then asked "Re your 'On Board Cassette Monitor' in the May issue. The term, a "low value tantalum capaciter" is a bit vague. In your next issue, could you suggest a value?"

(Any value from 0.1 micro F up would do - Ed.)

From: Chris Griffin, Doubleview, N.S.W.

A feature exists on the Sorcerer whereby the entire contents of an array can be dumped to, or read in from, tape. Could you publish a machine language sub-routine to do this on the '80?

(It's now in the Readers' Requests column, Chris - Ed.)

From: Graham Malcolm, Thomastown, Vic.

With reference to a conversation I had with you by phone about your Touchtype program. I tried your program from Cassette #-2 and had complete success. Perhaps you could publish modifications for the System 80 on-board cassette to allow it to run tapes such as your Touchtype program.

(Yes Graham, just as soon as we have a modification worked out, we will publish it - Ed.)

- 00000 -

** RADIO SHACK CASINO GAMES PROGRAM PACKAGE ** a review by Uno Hoo.

This three cassette package sells for $29.95. If you're looking for fun, then it's O.K. If you're looking to test your latest "foolproof" system - then forget it!

You get programs (no listings) for Craps, Keno, Slot Machine, Roulette, Wheel of Fortune, and Baccarat/Chemin-de-fer. As games they work well, and quite correctly to the rules and mechanics of the games. The graphics are first-rate, with a redraw routine in all the programs (the pretties are re-drawn each go, in case you've pressed break, clear, or hit the keyboard with a baseball bat), although some are a little slow, or odd, like the roulette wheel, which you might stare at for a few moments, until you realise that you're looking at the wheel from side-on! You'll also get a reasonable instruction book, with a cover that obviously came from their advertising agency!

I quote from the Introduction...

"These programs are so accurate that you can even learn and devise betting systems for the real games in Reno..."

Bulldust! There isn't a "system" in the world that doesn't depend to some extent on being able to use a staking plan. The roulette game, for example, restricts each player to betting one chip on each spin of the wheel. While you can be several players at once, so to speak, you'll find that you soon run out of "seats". Remember that the longest recorded run of one colour at Monte was 28 consecutive spins!

Keno only allows for one player, betting single chips. The Craps game suffers from similar problems. That's the gambler's point of view.

From the fun aspect - well, my missus doesn't exactly love "the beast", but she spent five hours just playing with the slot machine, so this package clearly has some great saving graces. However, the package price is wickedly high. $29.95 is a rip-off, when you consider the size of the market that TANDY dominates. It's a bit different for the small programmer who's trying to make a crust, but an organisation like TANDY/RADIO SHACK should be able to retail a package of this relatively mediocre calibre for about $12.50.

- 00000 -

***** XRX-III MODIFICATION - by Charlie Bartlett *****

We have all experienced the pure hell of trying to load our favourite program, (or a new one), from tape, only to have it not load at all or to find the memory full of nothing but garbage. So you sit there for six hours trying umpteen different volume settings and finally load the program, unfortunately by this time it's three o'clock in the morning and time to pack it in. At this point you have two choices, see your local doctor for some tranquilisers or instal this simple XRX-III Modification. I suggest the modification, it's cheaper!!!

Remove the top cover of the TRS-80. Place the pcb just below the " ? " key on the keyboard pcb, where it is held in place by the double-sided tape. BE SURE that its position will not interfere with the closing of the case.

Using a sharp knife CAREFULLY cut the foil pattern between Z23, pin 9 and Z4, pin 10 (see figure 3.)

Connect the red wire to VD = +5 VDC	(Z45, pin 10, figure 1.)
Connect the black wire to VS = GROUND	(Z44, pin 8, figure 1.)
Connect the yellow wire to	(Z43, pin 9, figure 1.)
Connect the green wire to	(Z4, pin 10, figure 1.)
Connect the blue wire to	(Z44, pin 12, figure 1.)
Connect the violet wire to	(Z24, pin 9, figure 1.)

Put the case back together, you've finished.

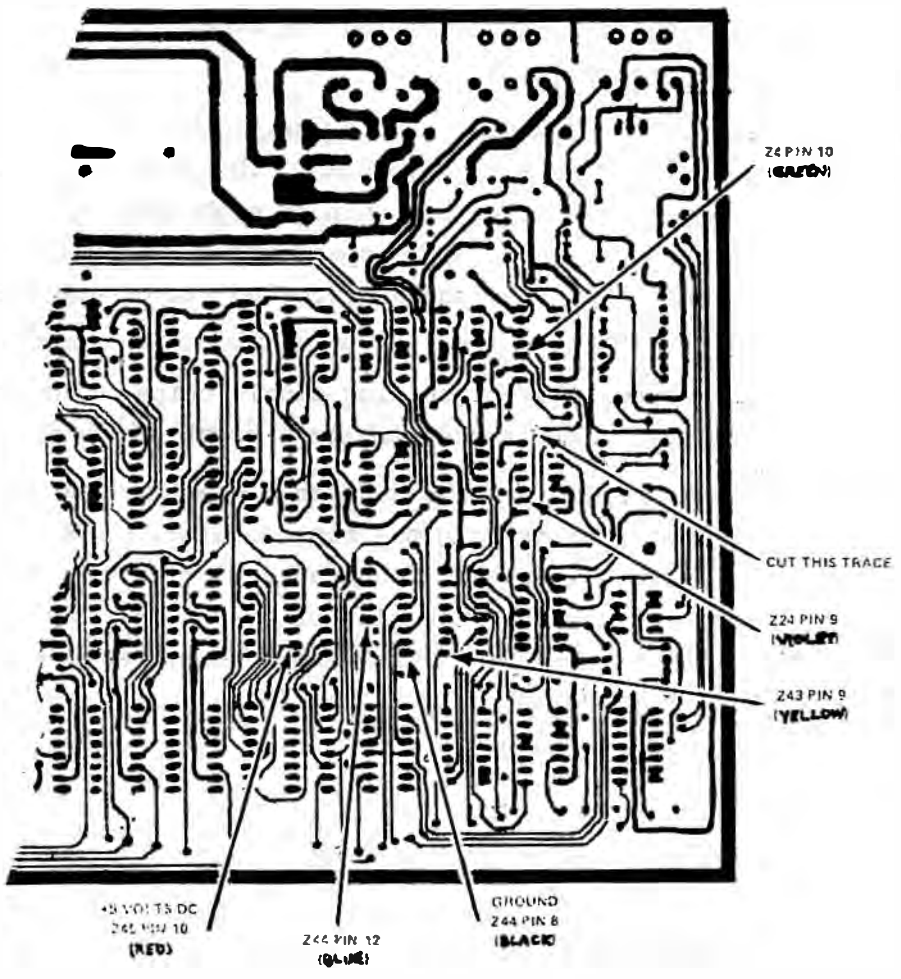

Figure 1. TRS-80 Microcomputer PCB, Bottom Side (Partial View).

For those of you that are interested in the hairy theory, what this little piece of hardware does is to reset the R/S Latch, Z24, just as the OUTSIG* line would do, but BEFORE THE OUTSIG* line does. In this way we allow for a stretched CASSIN signal (due to poor cassette motor speed control and high-volume stretching) and that is about all there is to it, but it works. I can personally recommend that you make this modification, as before installing it in my machine I could not load the program that is helping to write this article, (ELECTRIC PENCIL) at all, after the modification it loaded first time, EVERY TIME. The XRX-III is available from Tandy for about $3.50 (they charge for installation), hence this article. (Last month we said that the XRX-III is supplied free by Tandy. That is only partially correct. If you upgrade to Level II now, you will get an XRX-III included in the price - if you already have a Level II machine you will have to pay at least $3.50 (plus fitting if they do it for you). Leaving aside the principle of the thing, that's pretty good value.).

***** SOFTWARE SECTION *****

In reading the listings in this section, it should be noted that the longer Level II programs have been processed to make them easier to read. Either a space has been inserted after each colon or multi-statement lines have been split at each colon and occupy several lines on the page. There is no need to include these additional spaces or line breaks when typing these programs into your own '80.
The MICRO-80 printer shows either a "squiggle" or a square bracket instead of a line feed in the original listing- sadly it has no downward arrow symbol.

** CATCH ** Level I 4K by Michael Svensdotter.

The first of two level one programs by Michael that we are publishing this month, and you can look forward to seeing a lot more from this Level I wizard. This game is a sort of reversed "pong", where you trap the moving ball using a gap in the court wall.

Simulation of the Level II INKEY function is undertaken by conventional means, and all the player has to do is to press the backspace key at the right moment and the gap in the boundary will start moving. After every successful "trapping" the computer will change the speed and angle of the "ball".

The published listing doesn't have any on-screen instructions, but there is some memory left in 4K to add these if you wish.

```
10 REM ** CATCH ** COPYRIGHT (C) 1980 M. J. SVENSDOTTER
12 REM             4 HORTON AVENUE, SALISBURY NORTH, SA
15 REM LI/4K
20 REM ** MICRO-80 ISSUE 8 JULY 1980
60 A(1)=0:A(2)=0
70 F=RND(2):IFF=1F=2:A=960:B=1:G.90
80 F=1:A=1023:B=126
90 CLS:C=41:G=0:P.ATA,:R.(B,47)
100 F.X=1TO126
110  S.(X,0):S.(X,47)
120 N.X:R.(1,0):R.(126,0)
130 F.Y=1TO46S.2
140  S.(64,Y)
150 N.Y
160 F.H=1TO41
170  S.(1,H):S.(126,H)
180 N.H
190 IFG=1G=0:G.300
250 E=RND(0)+1:D=RND(0)+RND(0)+RND(0)+1
260 IFA=960X=125:E=-E:D=-D:G.290
270 X=2
290 Y=RND(46)
300 IFP.(B,47)=0R.(B,47):GOS.1100
310 S.(X,Y):IFP.(9,47)=0G.70
320 X=X+D:Y=Y+E:IF(C<40)*(P.(B,46)=0)S.(B,46):S.(B,45)
330 IFP.(B,47)=1F.Z=1TO20:N.Z
340 R.(X-D,Y-E):IFINT(X)=64X=X+1:Y=Y+E
350 IFY<1Y=1:E=-E
```

** GREEN SCREEN SIMULATOR **

The Editorial monitor has been a target for derision from the assocate Editors for some time now. It is an early, 115V, American model. The display shivers and shakes and the controls have to be set critically to avoid annoying hum bars running up the screen. We had heard of Green Screen simulators before but memories of bits of coloured plastic stuck on black and white TV's in the old days to simulate colour sets (that shows our age!) had prevented us from taking the idea seriously. An enterprising young man from Melbourne, however, sent us a sample of a Green Screen Simulator which attaches to the front of the monitor cabinet via sticky pads. We fitted it to the Editorial monitor and, lo and behold, it is now the envy of the neighbourhood! This product really works. the green display is much more pleasing to the eye than the white, contrast is much better and even the shivers and shakes don't seem so noticeable but that is probably psychological. We are now converted to the virtue of this device and have no hesitation in offerring it to our customers. The GREEN SCREEN SIMULATOR is made from green perspex, cut and curved to fit your monitor. Please make sure to specify whether you have an old (squarish) or a new (rounded) style monitor when you send in your order. PRICE $19.95 incl. P&P.

***** S KEY *****

Edwin paay has written a super new machine language utility to assist BASIC programmers. SKEY:

* is compatible with BMON and other high memory utilities
* stores BASIC expressions in the machine and allows you to recall them with a single shifted key entry
  Eg. you could designate (SHIFT) S to be STRING$(. There are 25 keys which store up to 15 characters each and one which stores 64 characters!
* prints graphics symbols on the screen as you type in or list programs. This enables you to create moving graphics as easily as text

PRICE $15.95 +50¢ p&p

Each cassette has TRS80,disk & Sys 80 versions

***** MICRO-80 PRODUCTS *****

The advert. on the opposite page will appear in the September editions of Electronics Australia and ETI. Since we have paid all that money for typesetting we thought we would give you a sneak preview. Not all the products are in stock yet. Both the Micropolis disk drives and the MPI drives are due in about the end of August but we will take orders immediately on $30 deposit. The MPI drive represents excellent value. Others in Australia are selling this same drive for over $400. It is the fastest mini-floppy around, having a track to track access time of only 5ms.- that's at least 5 times faster than most other makes. It is also capable of running in double-density mode although we are not yet sure if it can be used that way on the TRS-80.

****

***** NEWDOS 80 *****

We have had some supply problems with this exciting new product. Latest word is that our shipment is held up pending the arrival of documentation and will be shipped by air on 5/8. We apologise to those customers who have already ordered NEWDOS 80 from us, when we advertised it in the June issue we genuinely believed that we would have it in stock before we received any orders.

****

***** Tandy's TEAC Drives *****

We have been informed by a reader of MICRO-80 that the TEAC disk drives sold by TANDY are actually 40 track units but, since TANDY doesn't have a 40 track operating system, they are not telling anyone about it. If you have one of these drives why not buy a 40 track NEWDOS+ operating system from us, if it doesn't work we will change it over to 35 track for you and refund the difference. NEWDOS is much better than TRSDOS anyway so either way, you gain.

****

***** LOWER PRICES FOR NEWDOS *****

You will note that the prices for NEWDOS systems are lower than in our catalogue. This has been made possible by bulk buying. Anyone who has paid the higher price can subtract the difference from his next order.

****

MICRO-80 PRODUCTS FOR TRS-80 AND SYSTEM '80 MICROCOMPUTERS

77 TRACK DISK DRIVES

DOUBLE YOUR CAPACITY

DD-7S **$775**
Micropolis Floppy Disk, 77 Track, 100% larger capacity than most mini-floppy drives, complete with cable, power supply, chassis, and includes NEWDOS '80.

DD-7 ... ... **$649**
Same as above but no cable or Newdos '80.

DC-4 ... ... **$45**
4 drive connector cable.

★ ★ SPECIAL ★ ★ MPI DISK DRIVES ONLY $339!!

40 track bare drive for TRS-80. Only requires readily available 5 volt 0.7 amp and 12 volt 1 amp power supply to be up and running. Can be mounted in simple cabinet or used bare.

DISKETTES FOR TRS-80

GREENCORP ... ... **$4.50 ea**
VERBATIM 77 TRACK ... **$7.99 ea**
Minimum Order 10

EXATRON STRINGY FLOPPY

$352.50 incl p&p

15 times faster than cassette, infinitely more reliable. Completely under computer control, the stringy floppy is easier to use then disks and is a very much cheaper alternative. Will save and load any L2/16K software. Special software also available.

Wafers for Stringy Floppy $3.50 ea. Any Size

MICRO-80 MAGAZINE

ANNUAL SUBSCRIPTION ... **$24.00**
Monthly Magazine dedicated to TRS-80 and System '80 users. Every issue contains at least 6 new programs, plus problem solving columns, hardware articles, readers' letters, hints, etc., etc.

FREE SOFTWARE OFFER

$40 WORTH OF SOFTWARE FOR TRS-80 AND SYSTEM '80 WITH EVERY NEW SUBSCRIPTION TO MICRO-80!
Every new subscriber will receive on cassette, ready to load, 3 x Level I and 3 x Level II programs (includes our fabulous household budget program), with a regular retail value of at least $40.

16K MEMORY EXPANSION KIT

ONLY $97 incl. p&p

These are prime, branded, 200 ns (yes, 200 ns!) chips. You will pay much more elsewhere for slow, 350 ns. chips. Ours are guaranteed for 12 months. A pair of DIP shunts is also required to upgrade the CPU memory — these cost an additional $4.00. All kits come complete with full, step-by-step instructions, no soldering is required. You don't have to be an electronic type to instal them.

THE FABULOUS NEWDOS80 *IN STOCK NOW!*

ND-80 ... ... **$149**
The disk operating system that gives:

- New basic commands that support variable record lengths up to 4095 bytes long.
- Mix or match disk drives — supports any number of tracks from 18 to 80. Use 35, 40 or 77 track 5" mini disk drives or 8" disk drives, or any combination.
- A security boot-up for basic or machine code programs. User never sees "Dos-ready" or "Ready" and cannot "break" clear screen or issue any direct basic statement including "List" . . .

. . . and much, much more

ND-35+ **$99**
Newdos+ for 35 track drives

ND-40+ **$110**
Newdos+ for 40 track drives

BOOKS

LEVEL II ROM REFERENCE MANUAL
$24.95 plus $1.20 p&p
Produced in Australia by MICRO-80, this book is a must for the machine language programmer or for the BASIC programmer who wants to understand how the BASIC interpreter works. Over 70 pages packed full of useful information and sample programs. Applies to both TRS-80 and SYSTEM 80.

TRS-80 DISK AND OTHER MYSTERIES
$24.95 plus $1.20 p&p
The hottest selling TRS-80 book in the USA. Disk file structures revealed, DOS's compared and explained, how to recover lost files, how to rebuild crashed directories — this is a MUST for the serious Disk user.

INTRODUCTION TO T-BUG
$12.50 plus $1.20 p&p
Describes in detail how to use Tandy's T-bug. Each command explained and discussed in detail with examples. A must for the T-Bug user.

AUSTRALIAN SOFTWARE

BMON (L2/16K) $19.95 plus 50c p&p
The ultimate program to assist BASIC programmers.

TOUCHTYPE (L2/4K) $19.95 plus 50c p&p
22 lessons to teach you to type on your own keyboard and screen!

RPN CALCULATOR (L2/16K & 32K)
$14.95 plus 50c p&p
Turns your computer into a $600 calculator. Ideal for Architects, Surveyors, Engineers, Teachers, Scientists, etc.

U-BOAT (L2/16K) $7.50 plus 50c p&p
You're the Commander, read all the guages, fire the torpedoes, crash dive, get the enemy before his depth charges get you!!

MMM-1 GAMES PACK (L2/4K)
$7.50 plus 50c p&p
3 fast moving games — INDY 500, SUB-HUNT, KNIEVEL

MMM-2 GAMES PACK (L2/4K)
$7.50 plus 50c p&p
3 more games with fast moving graphics — TANK, THIEF, SHOOTOUT

PROGRAMS BY MICROSOFT

BASIC COMPILER **$210**
Converts Level II basic programs to machine code, automatically. A compiled program runs, on average, 3-10 times faster than the original basic program.
(Requires 48K One Disk)

DISK EDITOR ASSEMBLER
Save $30 ... ... **ONLY $109**
Supports macros, linking loader, editor, cross reference.
(Requires 32K One Disk)

ADVENTURE ON DISK ... **$37.15**
This game fills an entire diskette. Endless variety and challenge as you seek to rise to the level of Grand Master (until you gain skill, there are whole areas of the cave that you cannot enter).
(Requires 32K One Disk)

EDITOR ASSEMBLER-PLUS ... **$41.15**
A much improved editor assembler and debug/monitor for L2/16K machines. Assembles directly into memory, supports macros and conditional assembly, includes new commands — substitute, move, copy and extend.

LEVEL III BASIC **$59.95**
Loads on top of Level II Basic and gives advanced graphics, automatic renumbering, single stroke instructions (shift-key entries), keyboard debounce. Suitable for L2/16K and up. (Not Disk Basic).

SCOTCH BRAND PERSONAL COMPUTING CASSETTES

C-10 pack of 10 . . . $26.00 incl. p&p
C-30 pack of 10 . . . $28.00 incl. p&p

To: MICRO-80
P.O. Box 213, Goodwood, S.A. 5034

Please rush me the items checked below:

☐ 12 month subscription to MICRO-80 and my free software cassette $24.00

☐ 12 month subscription to MICRO-80 and the cassette edition, plus my free software cassette $60.00

☐ The lastest issue of MICRO-80 $2.50

PLUS THE ITEMS LISTED BELOW

DESCRIPTION	PRICE
TOTAL ENCLOSED	

Name ....

Address ....

.... Post Code ....

bankcard welcome here No ....

Please debit my Bankcard $ ....

Expiry date ....

Signature ....

```
360 IFY>46Y=46:E=-E
370 IF(X>2)*(X<125)G.300
380 G=0:IFX<2X=2
390 IFX>125X=125
400 IF(P.(X+1,Y)=1)+(P.(X-1,Y)=1)G=1:D=-D:S.(X,Y)
420 IFG=1G.450
430 IFF=1A(1)=A(1)+1:G.450
440 A(2)=A(2)+1
450 P.AT90,A(2);AT99,A(1);
460 F.N=1TO5:S.(B,C+N):N.N:F.N=1TO5:R.(B,41+N):N.N
470 IFG=0F.Z=1TO1000:N.Z
480 IFF=1F=2:A=960:B=1:G.500
490 F=1:A=1023:B=126
500 C=41:P.ATA,:S.(B,47):G.190
1100 IFC<1RET.
1110 IFC=42R.(B,C+1)
1150 S.(B,C+5):R.(B,C)
1160 C=C-1
1200 RET.
```

** FUEL ECONOMY ** LI/4K

Michael's second program this month is, like all his work that we've see
very compact. What more can we say - except that it works and works well, a
it's useful.

```
1 REM X FUEL ECONOMY BY M J SVENSDOTTER
2 REM X 4 HORTON AVE. SALISBURY NTH.   2589756
3 REM X 10/5/79
4 REM COPYRIGHT (C) 1979, 1980
5 REM MICRO-80 ISSUE 8 - JULY 1980
90 CLS
100 P."FUEL ECONOMY"
110 P."%%%%%%%%%%%%":P.
120 I."HOW MANY LITRES DID YOU USE";L
130 I."WHAT WAS THE FIRST ODOMETER READING";N
140 I."ENTER THE SECOND ODOMETER READING";O
150 G=INT(L/.0454609)/100:M=O-N:P=INT((M/G)*100)/100
160 K=INT(M*160.9)/100:J=INT((L/K)*10000)/100
170 P.AT128,"GALLONS";T.(20);"MILES";T.(40);"M/G
180 P.G;T.(20);M;T.(40);P
200 P.:P.:P."LITRES";T.(20),"KILOMETERS";T.(40);"L/100 KM
210 P.L;T.(20);K;T.(40);J
```

** INTEREST & LOAN REPAYMENT CALCULATOR ** LII/16K by Bernie Simpson.

MICRO-80's software editor is not a financial genius (more of a financial disaster) and he doesn't understand how Bernie's copyrighted formualae work! However, he is advised that (a) the formulae DO work, and (b) the program follows the formulae. He can tell you that the program logic is A1, therefore the program does what Bernie claims. Start typing!

```
10 ' COPYRIGHT (C) 1979, 1980
     AUTHOR:  BERNIE SIMSON    18 BULLER TCE,
20 '                           CHELTENHAM   SA.   PH 477528
30 'MICRO-80 ISSUE 8, JULY 1980
50 '  PROGRAM TO CALCULATE LOAN REPAYMENTS FOR ADJUSTABLE
60 '  RATE LOANS, USING FORMULA:
70 '     R=P*F[N*(F-1)/(F[N-1)   FOR INTEREST CHARGED ON MAXIMUM
75 '                             MONTHLY BALANCE.
80 '     R=P*F[(N-1)*(F-1)/(F[N-1)    FOR INTEREST CHARGED ON
82 '                                  MINIMUM MONTHLY BALANCE.
85 '   WHERE R=REPAYMENT PER PERIOD, P=PRINCIPAL,
87 '         F=INTEREST INCREASING FACTOR PER PERIOD,
89 '           (IE. F=ANNUAL INT RATE/1200 + 1)
90 '         N=NUMBER OF REPAYMENT PERIODS.
92 '
94 '   ***   THIS FORMULA IS COPYRIGHT OF B SIMSON.   ***
96 '
100 CLS:PRINT@19,"LOAN REPAYMENT CALCULATIONS"
120 PRINT@83,"============================="
140 PRINT:T=0:P1=0:P2=0:PI=0:N1=0:N2=0:NI=0:F=0:M$="":A$="$$#,###.##"
160 INPUT"MAX OR MIN MTHLY BALANCE";M$
180 IFMID$(M$,1,3)="MAX"THENMM=1:GOTO220
200 IFMID$(M$,1,3)<>"MIN"THENPRINT,,"WHAT ??":GOTO160
210 MM=-1
220 INPUT"STARTING PRINCIPAL";P1:IFP1<=0THENPRINT,,"WHAT ??":GOTO220
225 INPUT"ENDING PRINCIPAL";P2:IFP2<P1THENPRINT,"CAN'T BE LESS THAN STARTING PRI
NCIPAL":GOTO225
230 INPUT"INCREMENT";PI:IFPI<0THENPRINT,"CAN'T BE NEGATIVE":GOTO230
240 INPUT"STARTING TERM IN MTHS (1-480)";N1:IFN1<1ORN1>480ORN1<>INT(N1)THENPRINT
,"WHAT THE...??":GOTO240
245 INPUT"ENDING TERM IN MTHS (ALLOW MAX 4 COLUMNS)";N2:IFN2<N1ORN2>480ORN2<>INT
(N2)THENPRINT,"INVALID ENDING TERM":GOTO245
250 INPUT"INCREMENT (IN MTHS)";NI:IFNI<0ORNI<>INT(NI)PRINT,"INVALID INCREMENT":G
OTO250
255 IFN1+(NI*3)<N2THENPRINT,"INCREMENT TOO SMALL FOR 4 COLUMNS":GOTO250
260 INPUT"INTEREST RATE (% PER MTH)";F:IFF<=0ORF>10THENPRINT,,"WHAT THE...??":GO
TO260
262 CLS:PRINT"MONTHLY REPAYMENTS -    ";F;"% ON ";:IFMM=1THENPRINT"MAXIMUM";:ELSE
PRINT"MINIMUM";
263 PRINT" MONTHLY BALANCE"
265 F=F/100+1:PRINT"PRINCIPAL","     T E R M -----------":PRINT"  ($)",
268 PRINT"       ";:FORL=N1TON2STEPNI:PRINTL;"      ";:NEXT:PRINT
270 PRINT,"  ";:FORL=N1TON2STEPNI:Y=INT(L/12*10)/10:PRINT"(";Y;"YRS)";"  ";:NEXT
:PRINT
275 FORP=P1TOP2STEPPI
277 PRINTP,
280 FORN=N1TON2STEPNI
300 IFMM=1THENGOSUB500ELSEIFMM=-1THENGOSUB600ELSEPRINT"MM TEST ERROR":STOP
310 PRINTUSINGA$;R;:NEXTN:PRINT:NEXTP
399 END
490 ' **  MAX METHOD **
500 T=F[N:R=P*T*(F-1)/(T-1):RETURN
590 ' **  MIN METHOD **
600 T=F[N:R=P*T*(F-1)/(F*(T-1)):RETURN
```

***** EDITORS/ASSEMBLERS, MONITORS etc. *****

(a brief explanation)

Last month we promised that we would give a brief, simple explanation of some of the programs which machine language programmers use to help speed up their work. such programs are called utilities.

EDITOR/ASSEMBLER

The most obvious way to write machine language programs is to arm yourself with a card containing the 150 odd Z80 instructions, a flow chart of the program you want to write and then list down each instruction, all the relevant addresses, registers etc.. This is called "hand assembling" a program and, if you are writing a program of more than about 500 bytes in length, would drive you mad!

An alternative is to use an Editor/Assembler utility program. First, you load it into the computer then put it into EDIT mode, which allows you to enter text and edit it, rather like typing in a BASIC program. You use line numbers and, instead of BASIC statements, commands etc., you use mnemonics which are a convenient shorthand for the Z80 instructions (eg. LD=LOAD). When you have finished typing in the program you can use ASSEMBLY mode to assemble the program, which is the process whereby the mnemonics are converted to the actual machine code used by the Z80. You may write this out to a SYSTEM tape or, in some cases, put it directly into memory for execution Using an Editor/Assembler makes machine language programming very much faster than doinng it by "hand".

(continued on page 34)

** SYMBOL/BAS ** 16K DISK BASIC AND LINE PRINTER by Spencer George.

Those of you who have subscribed to Micro-80 from the early days will recall our stand on Educational Programs. This, then, is our first Educational Program, which we present with a great deal of pleasure.

The psychological basis for the approaches adopted here are detailed in Spencer's article, elsewhere in this edition, and the listing is well endowed with REMs, which include notes on altering the program for non-disk systems.

The program seeks to teach the student 14 chemical symbols, but can be easily altered to cope with any educational area where there is a one to one correspondence with two alphanumeric data sets.

Use is made of a "cassette-full of praise", complementary screen displays, etc., while the line-printer outputs a revision sheet for the student to take away after the learning session.

It doesn't matter if you don't give a damn about chemical symbols (like me) - this program still succeeds in teaching them to you, and, most importantly, it doesn't leave you feeling like an idiot when you make a mistake. (It even demonstrates how to take care of possible varient spellings of the same data).

Study the REMs carefully, because they clearly explain what each routine is doing, and what changes may be needed for your system. Just the "noddy" routine makes this program worthwhile!

```
14 '   SYMBOL/BAS  16k DISK BASIC, LINE PRINTER AND CASSETTE~

          MODIFICATIONS FOR LESSER SYSTEMS INCLUDED IN LISTING~
   ~
          MICRO-80 ISSUE 7, JUNE 1980
20 '         COPYRIGHT      C       SPENCER GEORGE    1980
100 '     S  Y  M  B  O  L  S  /  B  A  S
120 '                     S C G          13/3/80
150 '     THIS PROGRAM IS WRITTEN IN DISK BASIC, BUT MAY BE CHANGED TO
   LEVEL II BY ALTERING LINES    970,  1000,  1030   TO   INPUT
   INSTEAD OF LINEINPUT.~
      THIS CHANGE MEANS THAT A COMMA MAY NOT BE USED IN A DATE.
160 '     SO    3 MARCH     OR    3/3/80   WILL BE ACCEPTED~
                  BUT NOT   3 MARCH, 1980
180 '     IF A PRINTER IS NOT ATTACHED THEN LINES   300  ,  480  ,  2490
      AND              2580   SHOULD BE DELETED
240 CLEAR 2000
250 DIM ON$(14)
260 RANDOM
270 GOSUB 630           : '    DISPLAYS THE HEADING
280 GOSUB 820         : '     PLACES CHEMICALS AND SYMBOLS IN NAMED
   MEMORY
290 H = 1
300 GOSUB 950   :       '    COLLECTS  NAME, DATE, TIME
310 GOSUB 1130        : '     PLACES NODDY PICTURE IN NAMED MEMORY
320 GOSUB 1380        :  '   DISPLAYS SYMBOLS
330 C = 0
340 GOSUB 1460       : '      DISPLAYS
   " TODAY WE WILL USE THIS INFORMATION"
350 GOSUB 1230       : '      PLACES A LARGE  C    O    R    R    E    T
          IN NAMED MEMORY
360 C= 0
370 K = 1
380 GOSUB 1580       : '     PRODUCES FIRST SET OF  10  QUESTIONS ~
                               (WHAT IS SYMBOL FOR  ?      TABLE
   DISPLAYED)
390 CLS
400 K = 2
410 GOSUB 1580        : '     PRODUCES SECOND SET OF   10    QUESTIONS
    ~

                  (WHAT  IS THE SYMBOL FOR   ?   TABLE NOT DISPLAYED)
420 GOSUB 1760
430 K = 3
440 GOSUB 1580        : '     PRODUCES THIRD SET OF   10   QUESTIONS
   ~

                  (WHAT IS NAME FOR   ?      TABLE OF SYMBOLS ON
   DISPLAY)
450 CLS
460 K = 4
470 GOSUB 1580         : '     PRODUCES FOURTH SET OF   10   QUESTIONS
    ~

                 (WHAT IS THE NAME FOR  ?  TABLE NOT DISPLAYED)
480 GOSUB 1810        :      '     PRINTS NUMBER OF QUESTIONS CORRECTLY
   ANSWERED
490 GOSUB 3580        : '  DO YOU WANT A REPEAT    ??
590 '    ****************
610 '          THIS ROUTINE GIVES THE HEADING  C H E M I C A L   S Y M
   B O L S                      ON A FRAMED SCREEN
630 V = 15360
640 CLS
650 PRINT CHR$(23)
660 PRINT@330,"C H E M I C A L " :    PRINT @ 588,
    " S Y M B O L S"
```

```
670 FOR J2 = 64 TO 896 STEP 64
680 POKE V + J2, 191
690 POKE V + J2 + 62, 191
700 NEXT J2
710 FOR J2 = 0 TO 63
720 POKE V +J2,188
730 POKE V + J2 +960,143
740 NEXT J2
750 FOR J2 = 1 TO 1100
760 NEXT J2
770 RETURN
780 '    ***************
800 '      THIS ROUTINE SETS UP THE LIST OF FOURTEEN CHEMICAL SYMBOLS
    AND NAMES
820 DIM S$(14,2)
830 FOR J = 0 TO 13
840 FOR K = 0 TO 1
850 READ S$(J,K)
860 NEXT K
870 NEXT J
880 DATA AL, ALUMINIUM,~
    C, CARBON,~
    CA, CALCIUM,~
    CL, CHLORINE,~
    FE, IRON,~
    F,FLUORINE,~
    K. POTASSIUM,~
    H, HYDROGEN,~
    MG, MAGNESIUM
890 DATA HE, HELIUM,~
    NA,SODIUM,~
    O, OXYGEN,~
    ZN. ZINC,~
    S.SULFUR
900 RETURN
910 '    *****************
950 CLS
960 PRINT@128,"PLEASE TYPE YOUR NAME"
970 LINEINPUT Z$
980 LPRINT"~
    ~

    "; Z$
990 PRINT"PLEASE TYPE TODAY'S DATE
1000 LINEINPUT Z$
1010 LPRINT~
      Z$
1020 PRINT"PLEASE TYPE THE TIME
1030 LINEINPUT Z$
1040 LPRINT~
     Z$
1050 RETURN
1060 '    ****************
1080 '            THIS ROUTINE SETS UP THE NODDY PICTURE
1100 DATA 3,168,~
     5,148,~
     66,160,~
     67,158,~
     68,188,~
     69,173,~
      130,178,~
      131,187,~
      132,188,~
      133,183
```

```
1110 DATA 193.174,~
     194.145,~
     195.191,~
     196.189,~
     197,191,~
     198.162,~
     199,157
1120 DATA 258,131,~
     259,171,~
     261,151,~
     262,131,~
     322,176,~
     323,186,~
     325.181,~
     326,176.~
     70,144,~
     134,177
1130 CLS
1140 DIM A1(27),B1(27)
1150 FOR J = 1 TO 27
1160 READ A1(J), B1(J)
1170 NEXT J
1180 RETURN
1190 '    ****************
1210 '            THIS ROUTINE PLACES A LARGE  C  O  R  R  E  C  T   ~
                        IN NAMED MEMORY
1230 X$(1) =" ....     ....    ......   ......   ......   ....
     ...........".
1240 X$(2) =".. ..   .. ..   .. ..   .. ..   ..       .. ..       ..
     "
1250 X$(3) =".. 	   .. ..   .....    .....    ....     ..          ..
     "
1260 X$(4) =".. ..   .. ...  .. ..   .. ..   ..       .. ..       ..
     "
1270 X$(5) =" ....     ....  ..  ..  ..  ..  ......   ....        ..
     "
1280 FOR TY = 1 TO 5
1290 FOR TH = 1 TO LEN (       X$(TY)          )
1300 IF MID$( X$ (TY), TH,1) ="." THEN MID$( X$ (TY), TH, 1) = CHR$(191)
1310 NEXT TH
1320 NEXT TY
1330 RETURN
1340 '    **********************
1360 '            THIS ROUTINE DISPLAYS THE FOURTEEN CHEMICAL SYMBOLS AND
     NAMES
1380 CLS
1390 PRINT@127," ";
1400 FOR J =0 TO 6
1410 PRINT S$(J*2, 0); TAB(4); S$(J*2, 1),,    S$(J*2+1, 0); TAB(36);
      S$(J*2+1, 1)
1420 NEXT J
1430 PRINT
1440 PRINT
1450 RETURN
1460 PRINT@0,"TODAY YOU WILL USE THESE NAMES AND SYMBOLS";
1470 RETURN
1480 '    ***************
1520 '             THIS ROUTINE CHOSES A RANDOM NUMBER AND DISPLAYS ONE
     OF FOUR~
                           TYPES OF QUESTION :
1530 '   GIVE A NAME FOR THE SYMBOL   (WITH OR WITHOUT SYMBOL TABLE
     DISPLAYED)
1540 '   GIVE A SYMBOL FOR THE NAME   (WITH OR WITHOUT SYMBOL TABLE
     DISPLAYED)
```

```
1580 FOR RT = 0 TO 13
1590 QN$(RT) = " "
1600 NEXT RT
1610 FOR L = 1 TO 10
1620 IF K = 1 GOSUB 1380
1630 IF K = 3 GOSUB 1380
1640 Q = RND(14) -1
1650 IF QN$(Q) = "*"  THEN 1640
1660 QN$(Q) = "*"
1670 ON K GOSUB 2020 , 2020 , 2180 , 2180
1680 INPUT Z$
1690 GOSUB 2110
1700 NEXT L
1710 RETURN
1720 '    ***************
1740 '            THIS ROUTINE GIVES A DISPLAY HEADING FOR THE SECOND TYPE
     OF ~
                                                  QUESTION
1760 CLS
1770 PRINTCHR$(23);
1780 PRINT@522, "PART  TWO "
1790 FOR WW = 1 TO 800 :  NEXT
1800 RETURN
1801 '    ***************
1810 LPRINT"~
     ~
     YOU HAVE CORRECTLY ANSWERED   " ;   C   ;   "   QUESTIONS FROM A
     SET OF    40. ~
     ~
     "
1815 PRINT"~
     YOU HAVE CORRECTLY ANSWERED    ";  C  ;  "   QUESTIONS FROM THIS
     SET OF   40.
1816 FOR H3 = 1 TO 500 :  NEXT
1820 RETURN
1830 '    **************
1850 '            THIS ROUTINE FLASHES A STAR IN THE BOTTOM RIGHT HAND
     CORNER~
                         WHILE THE CPU IS WAITING FOR A KEY TO BE DEPRESSED.
     IT IS USED~
                          AS A DELAY WHILE THE DISPLAY IS BEING READ.~

1890 POKE 16382,42
1900 Z$ = INKEY$
1910 FOR JJ = 1 TO 100
1920 NEXT JJ
1930 POKE 16382,32
1940 FOR JJ = 1 TO 100
1950 NEXT JJ
1960 IF Z$ = "" THEN 1890
1970 RETURN
1980 '    **************
2020 PRINT@780,"PLEASE TYPE THE SYMBOL FOR ";  S$(Q,1)
2030 R$ = S$(Q,0)
2040 RETURN
2041 '    ***************
2060 'THIS ROUTINE CHECKS IF AN ANSWER IS CORRECT
2080 C = C + 1
2090 GOSUB 2250
2100 RETURN
2110 IF Z$ = R$ THEN  2080
2120 IF Z$ = "SULPHUR" AND R$ = "SULFUR" THEN   2080  ~
                                                   ELSE   GOSUB 2440
```

```
2130 RETURN
2140 '    **************
2180 PRINT@704,"PLEASE TYPE THE CHEMICAL NAME FOR  "; S$(Q,0)
2190 R$ = S$(Q,1)
2200 RETURN
2210 '    ****************
2230 '             THIS ROUTINE CHOSES THE NEXT REWARD DISPLAY
2250  H = H + 1
2260 IF H > 6 THEN H = 1
2270 ON H  GOTO 2320 , 2670 , 2790 , 3030 , 3330 , 3170
2280 '    **************
2300 '             THIS ROUTINE DISPLAYS  11  CORRECTS  DIAGONALLY
     ACROSS THE ~
                                            SCREEN
2320 CLS
2330 PRINT CHR$(23);
2340 FOR J = 1 TO 11
2350 PRINT TAB(J*2)"CORRECT"
2360 NEXT J
2370 GOSUB 3556
2380 CLS
2390 RETURN
2400 '    **************
2420 '             THIS ROUTINE CHOSES THE RELEARNING RESPONSE WHEN A
     STUDENT ERROR~
                                            OCCURS
2440 ON K GOTO  2480,  2480,  2570,  2570
2460 '    ****************
2470 '             THIS ROUTINE PRINTS AND DISPLAYS THE CORRECT SYMBOL
2500 LPRINT"~
     THE SYMBOL FOR   "; S$(Q,1) ;  "  IS  " ; S$(Q,0)
2505 PRINT"~
     THE SYMBOL FOR   "; S$(Q,1) ; "  IS  "; S$(Q,0)
2510 GOSUB 1890
2520 CLS
2530 RETURN
2540 '    **************
2560 '             THIS ROUTINE PRINTS AND DISPLAYS THE CORRECT NAME
2580 LPRINT"~
     THE CHEMICAL NAME FOR   "; S$(Q,0) ; "  IS  "; S$(Q,1)
2590 PRINT"~
     THE CHEMICAL NAME FOR   "; S$(Q,0) ; "  IS  "; S$(Q,1)
2600 GOSUB 1890
2610 CLS
2620 RETURN
2630 '    **************
2650 '             THIS ROUTINE FILLS THE SCREEN WITH  LARGE  Y E S  MANY
     TIMES
2670 CLS
2680 PRINT CHR$(23);
2690 FOR K2 = 1 TO 85
2700 PRINT"YES  ";
2710 NEXT K2
2720 GOSUB 3556
2730 CLS
2740 RETURN
2750 '    **************
2770 '             THIS ROUTINE DISPLAYS    R I G H T    IN A
     RECTANGULAR BOX
2790 CLS
2800 V = 15360
2810 PRINTCHR$(23);
2820 FOR HY = 404 TO 420
2830 POKE V + HY, 128 + 15
```

```
2840 POKE V + HY +128. 128 + 60
2850 NEXT HY
2860 FOR HY = 404 TO 404 +128 STEP 64
2870 POKE HY + V, 191
2880 POKE HY + V + 16. 191
2890 NEXT HY
2900 FOR HY = 1 TO 5
2910 Z$ = "RIGHT"
2920 N = ASC (  MID$ ( Z$,HY,1) )
2930 POKE V + 470 + HY * 2, N
2940 NEXT HY
2950 GOSUB 3556
2960 CLS
2970 RETURN
2990 '    *************
3010 '            THIS ROUTINE DISPLAYS STARS RANDOMLY
3030 CLS
3040 V = 15360
3050 PRINTCHR$(23)
3060 FOR SS = 1 TO 25
3070 D = RND(500)
3080 POKE V + 2 * D, 42
3090 FOR JH = 1 TO 25
3100 NEXT JH
3110 NEXT SS
3120 PRINT@512,"CORRECT"
3130 GOSUB 3556
3145 CLS
3150 RETURN
3152 '    ******************
3153 '     THIS ROUTINE DISPLAYS A VERY LARGE    C    O    R    R    E
     C    T
3170 CLS
3180 PRINT@512, " "
3190 PRINTX$(01);
3200 FOR HG = 2 TO 5
3210 PRINT X$(HG)
3220 NEXT HG
3230 FOR HG = 1 TO 500
3240 NEXT HG
3250 CLS
3260 RETURN
3290 '    ***************
3310 '           THIS ROUTINE DISPLAYS THE NODDY PICTURE
3330 CLS:   FOR SS = 1 TO 27
3340 POKE 15880 + A1(SS), B1(SS)
3350 NEXT SS
3360 FOR TY = 1 TO 3
3370 POKE 16012,179
3380 POKE 15883,128
3390 POKE15885,128
3400 POKE 15947,174
3410 POKE 15949,157
3420 FOR S = 1 TO 100
3430 NEXT
3440 POKE 15883,168
3450 POKE 15885,148
3460 POKE 16012,188
3470 POKE 15947,142
3480 POKE 15949.141
3490 FOR S = 1 TO 100
3500 NEXT S
3510 NEXT
3520 CLS
```

```
3530 RETURN
3551 '    ***************
3553 '            THIS ROUTINE TURNS ON THE CASSETTE PLAYER TO PLAY A
     COMMENT SUCH AS RIGHT,  WELL DONE,  CORRECT,  BEAUTY,  GOOD WORK,
     EXCELLENT,   YES
3556 FOR K4 = 1 TO 400
3557 OUT 255,12
3558 NEXT
3559 OUT 255,4
3560 RETURN
3562 '    ****************
3580 CLS
3590 PRINT"WOULD YOU LIKE ANOTHER SET OF QUESTIONS";
3600 INPUTZ$
3610 IF LEFT$( Z$,1 ) = "Y" GOTO 360
3620 CLS
3630 PRINT@524,CHR$(23); "GOODBYE"
3640 PRINT@1023, " ";
3650 END
3740 '  TO CHANGE THE SET OF NAMES AND SYMBOLS CHANGE THE DATA LINES~
                               880   AND   890

     THERE IS A ONE TO ONE CORESPONDENCE BETWEEN TWO SHORT ALPHANUMERIC

     SETS
3780 '    LINES 1410,  1460,  2020,  2120,  2180,  2490,  2500,2580
     AND   2590 ~
                                 WILL NEED APPROPRIATE CHANGES
```

```
3740 '  TO CHANGE THE SET OF NAMES AND SYMBOLS CHANGE THE DATA LINES↑

                               880   AND   890
3770 '   THIS PROGRAM MAY BE ALTERED TO SUIT ANY EDUCATIONAL AREA WHERE
     THERE IS A ONE TO ONE CORESPONDENCE BETWEEN TWO SHORT ALPHANUMERIC
     SETS
3780 '    LINES 1410,  1460.  2020,  2120,  2180,  2490,  2500.2580
     AND   2590 ↑
                                 WILL NEED APPROPRIATE CHANGES
```

(Continued from page 27)

MONITOR/DEBUGGER

A monitor is a program which enables you to control the activities of the Z80.
A debugger is a program which helps you to trace errors (bugs) in your machine language programs. With a monitor you can usually:- examine and change memory, either byte by byte or in blocks, punch and read SYSTEM tapes, GO to an address in memory and execute a program there, move blocks of data around memory, relocate machine language programs to operate in different parts of memory, etc..
A debugger will usually allow you to break theoperation of machine language programs at specified points, examine and change registers, sometimes to step through a program instruction by instruction. Most programs do not work first time and a monitor/ debugger is a useful tool to use in fixing them up.

(continued on page 35)

** SOUND GENERATION WITH THE TRS-80 ** a general introduction by Peter Hartley.

There are three main approaches to sound generation with a TRS-8Ø.
(a) Modifying the radio frequency interference that the '8Ø generates so that the "music" can be picked up on an ordinary A.M. Radio.
(b) Taking charge of the cassette port, and delivering the appropriate square-waves to the cassette jack.
(c) Interfacing the '8Ø to an external sound generator through one or more I/O ports.

Articles on the first and third approaches are in course of preparation. The second approach is the basis of two programs in this issue...and readers are urged to study the appropriate sections of "TRS-8Ø ASSEMBLY LANGUAGE PROGRAMMING" by William Barden, Jr., available from your local Tandy Store for the bargain price of $3.95.

Ron Sully has prepared copious notes on his effort, and these are reproduced in full. Note that line Ø of Ron's listing is a dummy line of 255 bytes.... used for the M/L routine.

A.F. West has adopted a very similar approach, but uses the two part method, a M/L dump, which is accessed by a BASIC master program, which allows the user to test various frequency/duration mixes, and responds with the access code required to generate the same sound from the BASIC program that you still haven't written. This M/L routine can then be used with your own BASIC program and you've got instant sound!

Use a good monitor to load the M/L routine (BMON perhaps) and save a copy to tape. Then write and save the BASIC program. Details of memory protect level, etc. are given in appropriate REMs.
*(HINT - To return to BASIC from a SYSTEM load, without accessing the M/L routine, just hit <ENTER>).

(continued from page 34)

DISASSEMBLER
A disassembler is a program which performs the opposite function to an assembler. It convert a machine language program back into mnemonics. This is useful for making minor changes to a program, discovering how it works etc..

SOUND EFFECTS (L2/16K/4K) - by Ron Sully.

This BASIC program demonstrates the possibilities of the use of sound effects in BASIC programs by having a M/L routine reside in a REM statement.

The M/L routine is essentially in two parts:

1. The routine which plays the notes, and
2. The data or table of values which determine the duration and frequency of the note.

(For further reading see RADIO SHACK publication "TRS-80 Assembly Language Programming" pp.179-183).

In this program the REM statement is at line Ø and part 1 of the M/L routine occupies MEM locations 17139 (42F3H) to 17187 (4323H). Part 2 occupies the remainder of the line, 17188 (4324H) to 17371 (43DBH).

To hear the sounds you will need either a cassette recorder/player with AUXILIARY INPUT facilities or an amplifier. (I use my CTR 41 which has been modified in accordance with the procedure published in MICRO-80 Issue 1, pp.16-17).

If you use a cassette, remove any tape that may be in it, hold down the record/protect lever and press the record/play buttons. Some better types of cassette have an AUX INPUT jack which allows the cassette to be used as an amplifier without having to press the record/play buttons. If you have one of those all the better. You can really turn the volume up. (Stand back!!) Simply connect the grey plug on the TRS-80, yellow lead on the System 80, to the cassette aux input or amplifier and you are in business.

To get the program going, CLOAD it - (if you are keying it in make sure you CSAVE it before RUNning it), prepare your cassette or amplifier then RUN...

(I don't profess to be a musician - in fact I'm tone deaf! So if the sounds I created aren't to your liking then read on!)

As mentioned previously the sound is created by a M/L routine which is in two parts. The first part is the DATA at line 1Ø and is POKEd into line Ø by the loop at line 2Ø ....FOR K = 17139 TO 17187:
READ Y:POKE K,Y:NEXT ..... so under no circumstances should you either;

1. Edit line Ø or shorten it.
2. Change the DATA in line 1Ø, or
3. Change the K loop in line 2Ø.

(unless you are an EDDY PAAY and know what you are doing).

The second part of the M/L routine determines the notes you hear. The values are again POKEd into memory location 17188 to 17371 by different routines. (If you exceed MEM location 17371 you will start POKEing into line 1Ø and will blow the program).

To create your own sounds you will need to POKE values into two memory locations for each note. The first value determines the duration of the note, the second determines the frequency. (And of course there's no need to tell you that the values must be in the range 1-255!)

After POKEing in the values for the notes, finish off by POKEing Ø into the next MEM location. (It should be an even number). (The Ø acts as a check for the M/L routine. It says ..
"No more notes to play...go back to BASIC"). Without it...well perhaps you'd like to find out!

To illustrate; to create the first BEEP you hear I POKEd the following:

```
POKE 17188, 9Ø : POKE 17189, 9Ø : POKE 1719Ø, Ø  (Line 3Ø).
    DURATION          FREQUENCY        RETURN
```

For the mathematical and musical buffs the following equations will aid in determining specific frequencies:

```
VAL = INT(1/FREQ*1Ø↑6/14.64)
```

where VAL = value to be POKEd and FREQ = frequency of the note. e.g. If you wish to play C_5 (frequency 523.3 Hz) then, using the above equation the value to be POKEd is 13Ø.

```
VAL = INT(1/523.3*1Ø↑6/14.64)
    = 13Ø
```

Similarly if you pick a value and you wish to know what frequency the note is, use the equation:

```
FREQ = INT(1/VAL*1Ø↑6/14.64)
```

Knowing we can only POKE the values 1 to 255 the above equation tells us the frequencies we can play are in the range of 267 Hz to 68,3Ø6 Hz. It also tells us that the higher the number we can use the lower the frequency. (VAL of 1 = 68,3Ø6 Hz, VAL of 255 = 267 Hz). Which means we can reproduce the notes from just above Middle C (261.6 Hz) to ear piercing.

For best results keep your values somewhere between 6Ø and 18Ø for both frequency and duration. You will need to experiment with the value for the duration. As long as you remember that the higher the number the longer the duration for each note.

To use this particular sound routine in your own BASIC programs you will need to include the following in your program:

1. Line Ø with a GOTO 2Ø followed by a REM and the rest of the line can be filled with spaces.
2. Line 1Ø the DATA list for part 1 of the M/L routine.
3. Line 2Ø a FOR..NEXT loop which reads the DATA and POKEs it into mem locations 17139 to 17187.
4. POKE 16526, 243 : POKE 16527, 66. The entry address of the M/L routine.
5. Different routines to POKE your sound effects into mem locations 17188 to 17371, and
6. X = USR(Ø) to jump to the M/L routine when you wish to bring the sounds into effect. (For general use suggested places are, before; PRINT statements, CLS, ENTER and INKEY$).

WARNING!! As all of us novices know, playing around with the M/L routines and POKEing around can get us into trouble. The following points are mentioned so you don't find the same pitfalls that I did:

1. DON'T TRY TO EDIT LINE Ø
2. DON'T POKE OUTSIDE THOSE MEM LOCATIONS
3. MAKE SURE YOU HAVE THE PROGRAM ON TAPE BEFORE RUNNING IT
4. IF YOU GET AN OD ERROR MESSAGE (AFTER BREAK THEN RUN) USE "GOTO 17Ø" TO KICK OFF AGAIN THEN SELECT OPTION 4. YOU CAN THEN BREAK AND RUN WITHOUT ANY HASSLE.
5. IF YOU IGNORE POINT 3 ABOVE THEN RE ENTER LINE Ø AS FOLLOWS:
   Ø GOTO 2Ø: REM....(FILL WITH SPACES)
   THEN CSAVE BEFORE RUNNING.

Last but not least. The DATA values in line 32Ø are the specific values for duration and frequency to play the tune. (What did you expect? TCHAIKOVSKY?)

4Kers will have to remove REM lines 4Ø, 5Ø, 6Ø, 7Ø, 33Ø and 34Ø to RUN the program

Hint? Use BMON to merge your program with lines Ø, 1Ø, 2Ø and 3Ø of this program.

I hope you have fun with your new found sound.

SOUND 'FECTS LII/16
=====================

```
0 GOTO20:
          REM  THIS LINE IS A DUMMY REMARK STATEMENT AND WIL
    L BEOVERWRITTEN BY THE MACHINE LANGUAGE ROUTINE.
                UNDER NO CIRCUMSTANCES ARE YOU TO EDIT THIS
    LINE OR REMOVE THIS LINE.    (FILL THE REMAINDER WITH SP
    ACES)
10 DATA221,33,36,67,221,78,0,121,183,200,221,70,1,62,1,211,25
    5,16,254,221,70,1,62,2,211,255,16,254,13,194,253,66,221,
    35,221,35,1,255,255,33,48,0,9,218,29,67,195,247,66
20 POKE16526,243:
         POKE16527,66:
         FORK=17139TO17187:
         READ Y:
         POKEK,Y:
         NEXT
30 POKE17188,90:
         POKE17189,90:
         POKE17190,0:
         X=USR(0)
40 ' COPYRIGHT MAY 1980
50 'RONALD J. SULLY
60 ' 117 BRYANT RD, LOGANHOLME, QLD. 4129
70 'PHONE (07) 2098370
80 DEFINTA-Z:
         DIMS(35):
         CLS:
         PRINTTAB(15)"DEMONSTRATION SOUND EFFECTS":
         PRINTTAB(15)"---------------------------"
90 PRINT"THIS  SIMPLE  PROGRAM  DEMONSTRATES  ONE OF THE WAYS
     THAT SOUND EFFECTS CAN BE ADDED TO  '80  PROGRAMS TO GI
    VE THEM THAT BIT OF EXTRA APPEAL.
100 PRINT"THE  M/L' ROUTINE  REQUIRED  TO  PUT THE SOUNDS INT
    O EFFECT WILLRESIDE  IN MEMORY  LOCATION  17139 TO 17187
    . (REM LINE 0)   THE VALUES WHICH DETERMINE  THE FREQUEN
    CY AND DURATION OF NOTES CAN OCCUPY MEMORY LOCATION 1718
    8 TO 17371.
110 PRINT"UNDER NO  CIRCUMSTANCES  ARE THE VALUES IN THE DATA
     STATEMENT ATLINE  10  TO  BE CHANGED.   TO ALTER THE  N
    OTES  PLAY AROUND  BYPOKING  DIFFERENT  VALUES  INTO MEM
    ORY LOCATIONS 17188 TO 17371.YOU MAY POKE  ANY NUMBER  O
    F VALUES  PROVIDING YOU ";
120 PRINT"DO NOT EXCEEDMEM LOCATION 17371 AND THE LAST VALUE
    YOU POKE MUST BE '0'.":
         PRINT"PRESS ANY KEY TO CONTINUE.
130 A$=INKEY$:IFA$=""THEN130ELSECLS:PRINT"MAKE READY YOUR CASSE
    TTE OR AMPLIFIER FOR SOUND EFFECTS . . .":FORX=1TO30:READS
    (X):NEXT
140 PRINT:PRINT"PRESS ANY KEY TO CONTINUE.   (WITH SOUND!)
150 A$=INKEY$:
         IFA$=""THEN150ELSECLS:
         X=USR(0)
```

```
160 PRINT"YOU SHOULD HAVE HEARD A BEEP THEN. IF NOT, CHECK TH
    E CONNECTIONSTO YOUR CASSETTE OR AMPLIFIER.":
         PRINT"IF SATISFIED THAT EVERYTHING IS AOK.  PRESS
    ANY  KEY  TO GET  ARECURRING BEEP. USE THIS BEEP AS A VO
    LUME CHECK.
170 PRINT"WHEN  HAPPY WITH THE VOLUME  PRESS ANY KEY  AGAIN T
    O CONTINUE TOFURTHER SAMPLES.
180 A$=INKEY$:
         IFA$=""THEN180
190 A$=INKEY$:
         IFA$=""THENX=USR(0):
         GOTO190
200 CLS:
         X=USR(0):
         PRINT:
         PRINT"THIS DEMONSTRATION PROGRAM OFFERS THE FOLLOWI
    NG VARIATIONS.":
         X=USR(0):
         PRINT"(ALL CREATED BY POKEING DIFFERENT VALUES  INT
    O MEMORY  LOCATIONS17188 TO 17371).":
         PRINT
210 X=USR(0):
         PRINT"1.   A SINGLE BEEP":
         X=USR(0):
         PRINT"2.   A SIREN (?)":
         X=USR(0):
         PRINT"3.   A WARNING SIGNAL":
         X=USR(0):
         PRINT"4.   LASER (?) SHOTS":
         X=USR(0):
         PRINT"5.   A SERIES OF RANDOM NOTES":
         X=USR(0):
         PRINT"6.   A TUNE
220 X=USR(0):PRINT:PRINT"SELECT 1, 2, 3, 4, 5 OR 6 . . . .
230 A$=INKEY$:
         IFA$=""THEN230ELSEA=VAL(A$):
         IFA<1ORA>6THEN220
240 FORT=1TOA:
         X=USR(0):
         FORZ=1TO30:
         NEXTZ,T:
         ONAGOTO260,270,280,290,300,310
250 POKE17188,120:
         POKE17189,120:
         POKE17190,0:
         PRINT:
         INPUT"PRESS ENTER TO RETURN TO OPTIONS";A$:
         GOTO200
260 CLS:
         PRINT:
         PRINT:
         PRINT:
         PRINT"THIS IS A SINGLE BEEP ....":
         POKE17188,100:
         POKE17189,100:
         FORT=1TO600:
         NEXT:
         X=USR(0):
         GOTO250
```

```
270 CLS:
         PRINT:
         PRINT:
         PRINT"A SIREN (?) . . . .":
         FORK=17188TO17251STEP2:
         POKEK,10:
         POKEK+1,K-17130:
         NEXT:
         POKE17252,0:
         FORT=1TO30:
         X=USR(0):
         NEXT:
         GOTO250
280 CLS:
         PRINT:
         PRINT:
         PRINT"A WARNING SIGNAL . . . .":
         Z=70:
         FORK=17188TO17357STEP2:
         POKEK,200:
         POKEK+1,Z:
         Z=-Z+120:
         NEXT:
         POKE17358,0:
         X=USR(0):
         GOTO250
290 CLS:
         PRINT:
         PRINT:
         PRINT"LASER (?) SHOTS. . . .":
         FORK=17188TO17247STEP2:
         POKEK,10:
         POKEK+1,K-17186:
         NEXT:
         POKE17248,0:
         FORT=1TO20:
         X=USR(0):
         NEXT:
         GOTO250
300 CLS:
         PRINT:
         PRINT:
         PRINT"A SERIES OF RANDOM NOTES . . . . .":
         FORT=17188TO17365:
         POKET,INT(60*RND(0)+60):
         NEXT:
         POKE17366,0:
         X=USR(0):
         GOTO250
310 CLS:
         PRINT:
         PRINT:
         PRINT"A TUNE . . . .":
         FORX=17188TO17218:
         POKEX,S(X-17187):
         NEXT:
         X=USR(0):
         GOTO250
320 DATA160,144,63,162,92,172,96,144,160,144,64,144.112,128,2
    40,144,93,162,91,173,96,144,224,107,72,95,255,84,0,0
330 ' THE MACHINE  LANGUAGE ROUTINE AND TUNE ARE FROM THE TAN
    DY RADIO SHACK PUBLICATION  'TRS-80 ASSEMBLY LANGUAGE PR
    OGRAMMING' PP 182-183.  THE M/L ROUTINE WAS RELOCATED AN
    D MODIFIED TO RUN  WITH BASIC.
340 ' MANY THANKS TO EDDY PAAY AND MICRO-80.  THIS PROGRAM WA
    S  THAT MUCH EASIER TO COMPILE USING BMON.  RON SULLY
```

```
  4 '  SOUND LEVEL II / 16k  - REQUIRES M.L. DUMP - NOT SUITED TO DISK BASIC

  5 'MEM 32737 - CIM = 7FE2-7FFF~
      A.F. WEST, 13 ROSE STREET, BRUNSWICK, 3506
  6 'MICRO-80 ISSUE 8 JULY 1980
 10 POKE16526,226: POKE16527,127
 20 CLS: PRINT
    "                       ** S O U N D  **
 30 PRINT: PRINT
    "THIS IS A PROGRAMME TO TEST FOR THE SOUNDS DESIRED BEFORE
    ENTERING  THE DATA INTO THE PROGRAMME BEING COMPILED.
 40 PRINT"IT USES A RELOCATABLE M/L SUBROUTINE. (DETAILS ARE GIVEN LATER)
    ~
    ~
    THE TONE OF THE SOUND AND ITS DURATION DEPENDS UPON THE NUMBER   SET
    (IN BASIC) IN THE USR(  ) CALL.
 50 PRINT"THIS NUMBER IS MADE UP FROM TWO VALUES : -~
            FV - WHICH DETERMINES THE FREQUENCY OF THE TONE~
            DU - WHICH DETERMINES ITS DURATION
 60 PRINT"A TABLE OF THE APPROX. FREQUENCIES AND DURATIONS OF THE IOUND
    TONES CORRESPONDING TO VALUES OF FV AND DU FOLLOW.
 65 GOSUB 135: GOSUB 70: GOSUB 135: GOTO140
 70 CLS: PRINT
    "FREQUENCY VALUES : - (FV NO. = FREQ. IN HERTZ)
 80 PRINT"1 = 14300"," 8 = 2299","19 = 1000"," 60 = 317
 90 PRINT"2 =  8200"," 9 = 2062","20 =  943"," 80 = 240
 95 PRINT"3 =  5750","10 = 1869","25 =  758","100 = 192
100 PRINT"4 =  4445","11 = 1695","30 =  633","125 = 153
105 PRINT"5 =  3571","13 = 1450","35 =  544","150 = 146
110 PRINT"6 =  3030","15 = 1250","40 =  476","200 =  96
115 PRINT"7 =  2632","17 = 1111","50 =  381","255 =  75
120 PRINT: PRINT
    "DURATION VALUES : - (DU NO. = TIME IN SECS.)
125 PRINT" 1 = .007",," 50 = .345
    ": PRINT
    " 5 = .035",,"100 = .690
130 PRINT"10 = .069",,"127 = .876
    ": PRINT
    "25 = .173",,"  0 = 1.753
    ": RETURN
135 PRINT@990,"";: INPUT
    "(PRESS <ENTER> TO CONTINUE) ";Z$: RETURN
140 CLS: PRINT
    "TO TEST IF THE : SOUND:  MEETS YOUR REOUIREMENTS, TRY DIFFERENT
    VALUES FOR FV & DU USING THE <T> COMMAND.~
    (THE TONE WILL BE REPEATED 5 TIMES FOLLOWED BY THE USR(  ) NUMBERTO BE
    ENTERED INTO YOUR OWN PROGRAMME)
145 PRINT: PRINT
    "CONNECT THE OUTPUT FROM THE CPU (NORMALLY PLUGGED INTO THE
    RECORDER AUDIO SOCKET) TO A SUITABLE MINI AMPLIFIER/SPEAKER.
150 PRINT: PRINT
    "         THE COMMAND LIST FOR THIS PROGRAMME IS : -~
    ~
                     T -  TO TEST THE SOUND~
                     L -  FOR APPROX. FREQUENCY & DURATION VALUES~
                     P -  FOR M/C LANGUAGE PROGRAMME FOR :      SOUND:
    "
```

```
155 PRINT"                    E -  TO EXIT THIS PROGRAMME"
                     H -  FOR COMMAND LIST
160 C$=INKEY$: IFC$=""THEN160
165 IFC$="H" THENCLS: GOTO150
170 IFC$="E"THENCLS: GOTO1000
175 IFC$="L"GOSUB 70: GOTO160
180 IFC$="T"THEN300ELSEIFC$="P"THEN200ELSE160
200 CLS: PRINT
    " * MACHINE LANGUAGE LISTING FOR RELOCATABLE : SOUND:  PROGRAMME *
205 PRINT: PRINT
    "B ####  CD","+0007  ED","+000E  03","+0015  10
210 PRINT" +0001  7F","+0008  5B","+000F  B3","+0016  F6
215 PRINT" +0002  0A","+0009  3D","+0010  D3","+0017  18
220 PRINT" +0003  3E","+000A  40","+0011  FF","+0018  F2
225 PRINT" +0004  01","+000B  45","+0012  0D","+0019  25
230 PRINT" +0005  0E","+000C  2F","+0013  28","+001A  20
235 PRINT" +0006  00","+000D  E6","+0014  04","+001B  F1
240 PRINT,,,"+001C  C9"
250 PRINT"NOTE:  IN THIS PROGRAMME THE BASE B FOR THIS M/C  SUBROUTINE
    IS 7FE2.  IF YOU CHANGE THE BASE THEN CHANGE THE VALUES (DEC)   POKED
    INTO ADDRESSES 16526 (LSB) AND 16527 (MSB) ACCORDINGLY."
255 PRINT"(E.G. IN THE : BASIC LISTING:  OF THIS PROGRAMME<0tHE NUMBERS
     226 & 127 IN LINE NO. 10 WOULD BE CORRESPONDINGLY ALTERED)
260 GOTO160
300 CLS: INPUT
    "TRIAL FREQUENCY (FV) NO.    (BETWEEN 1 & 255)   ";FV
305 IF FV<1 ORFV>255 THENPRINT"THAT NUMBER IS NOT VALIF": FORX=1TO1000:
    NEXT: GOTO300
310 INPUT"TRIAL DURATION  (DU) NO.    (BETWEEN 0 & 127)   ";DU
320 S=256*DU+FV: IFS>32767 THENPRINT
    "THAT NUMBER IS NOT VALID. TRY AGAIN": FORT=1TO1000: NEXT:
    GOTO300
330 FORX=1TO5: SS=USR(S): FORT=1TO200: NEXTT,X
340 PRINT: PRINT: PRINT
    "IF THAT FLAMING NOISE APPEALS TO YOU, ENTER INTO YOUR MAIN
    PROGRAMME WHERE YOU WANT THIS SOUND...."
    "
                         SS = USR(";S;")
350 GOTO160
1000 SS=USR(6500): SS=USR(200): PRINT
    "SORRY ABOUT THAT !!   MY YOUNG BROTHER ADDED THAT COMMENT.":
    FORX=1TO1500: NEXT: CLS
```

```
7FE0  00 00 CD 7F 0A 3E 01 0E 00 ED 5B 3D 40 45 2F E6
7FF0  03 B3 D3 FF 0D 28 04 10 F6 18 F2 25 20 F1 C9 00
```

32767
- 32
―――
32736 man

***** NEXT MONTH'S ISSUE *****

Still no sign of the Chess War article. If we haven't got it for next issue we will have to start all over again. Next month's software will include:

DEC BIN DEC (L1)
A level I program which teaches the relationship between Decimal, Hexadecimal and binary arithmetic

ADD HEX ADD (L2)
A Level II program similar to the above.

B GAME (L2)
In this business game you run your own business buying, processing and selling your product in competition with other companies doing the same. You can borrow money, pay dividends, etc. A full set of books presented at the end of each month shows you how well you have done.

JETBOAT (L1)
Try to steer your jetboat across the river against the vagaries of the current without being swept over the falls to join the Editor, who has never made it yet!

ANDROID SHOOTMAN (L2)
This is a 21st Century version of Hangman. Graphics are Androids being executed by laser beams rather than gallows. Quite grisly, really.

POET (L2)
This program generates poetry from an inbuilt data-base which is easily changed by people who do not like Edgar Alan Poe.

Plus all the usual articles, letters, news etc.

To MICRO-80
Please consider the enclosed program for... tick where appropriate

(i) Publication in MICRO-80

(ii) Publication on disk or cassette only

(iii) Both

Name..

Address...

....

.................................Post Code............

***** CHECK LIST *****

Please ensure that the cassette or disk is clearly marked with your name and address, program name(s), Memory size, Level I, II, System 1 or 2, Edtasm, System, etc. The use of REM statements with your name and address is suggested, in case the program becomes seperated from the accompanying literature.

Ensure that you supply adequate instructions, notes on what the program does and how it does it...etc

For system tapes, the start, end, and entry points, etc

The changes or improvements that you think may improve it.

Please package securely - padabags are suggested - and enclose stamps or postage if you want your cassette or disk returned.

***** CASSETTE EDITION INDEX *****

Level I programs are on Side 1 (not suitable for System 80 computers); Level II programs are on Side 2. Each program is recorded twice in succession.

			APPROX. START POSITION		
<u>SIDE 1</u>		I.D.	CTR-41	CTR-80	SYS 80
CATCH	L1/4K	-	16	10	-
			48	30	-
FUEL ECONOMY	L1/4K	-	80	50	-
			97	60	-
<u>SIDE 2</u>					
SYMBOL/BAS	L2/16K	C	16	10	7
			120	75	5
INTEREST	L2/4K	T	210	130	90
			233	145	100
SOUND EFFECTS			258	160	110
			290	180	125

Single copies of the Issue 8 cassette may be purchased by SUBSCRIBERS for $3.50. The combined magazine and cassette subscription rate is $60.00 for 12 issues.

To: <u>MICRO-80, P.O. Box 213, Goodwood, S.A. 5034.</u>

Please rush me the items checked below:

...12 month subscription to MICRO-80	$24.00
...12 month subscription to MICRO-80 plus the cassette edition	$60.00
...The latest issue of MICRO-80	$ 2.50

The MICRO-80 PRODUCTS listed below:

DESCRIPTION	PRICE
TOTAL ENCLOSED WITH ORDER	

☐ Cheque ☐ Bankcard ☐ Money Order

Bankcard Account Number

Signature........................Exp. End..............

NAME.......................................

ADDRESS....................................

......................POST CODE...........

Bankcard

A SPECIAL OFFER!

TO

SUBSCRIBERS OF **MICRO 80**

FOR THE FIRST TIME IN AUSTRALIA!
GENUINE TANDY TRS–80 MICRO-COMPUTERS, HARDWARE AND SOFTWARE AT

GREAT DISCOUNT PRICES!!!

HOW TO ORDER –

1. SELECT ITEMS FROM 1980 TANDY CATALOGUE
2. DEDUCT 10% FROM ADVERTISED PRICES
3. POST US YOUR ORDER STATING DESCRIPTION CAT. No AND A CHEQUE OR MONEY ORDER.

WE WILL –

1. ATTEND TO YOUR ORDER WITHIN 7 DAYS *
2. SUPPLY GOODS SELECTED FREIGHT FREE !
3. SEND ADVERTISING REGULARLY TO KEEP YOU INFORMED OF CURRENT SPECIALS !

** subject to availability*

to **CONQUEST ELECTRONICS** Pty. Ltd.
212 Katoomba St. KATOOMBA 2780

Please supply –

OTY.	AT NO	O S .	ADV. PRICE
		SUB TOTAL	
		LESS 10%	
		FIND CHEQUE FOR **TOTAL**	

S ND **FREIGHT FREE** TO

N ME ....

ADDRESS ....

.... P/Code ....

TANDY ELECTRONICS DEALER

CONQUEST ELECTRONICS PTY. LTD.

212 KATOOMBA ST KATOOMBA N.S.W. 2780 PHONE (047) 82 2491