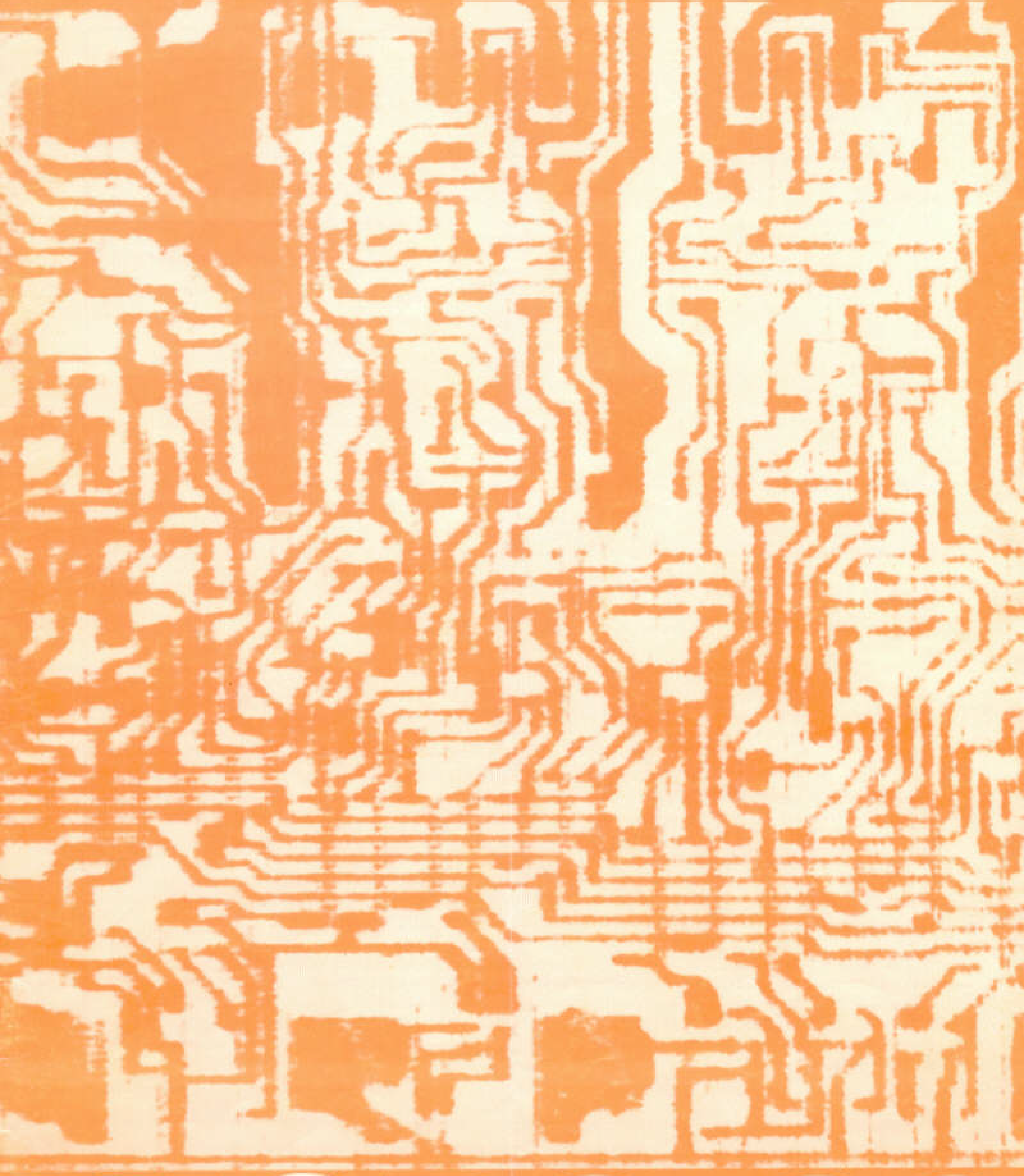
TRS-80 SYSTEM 80 VIDEO GENIE



MICRO-80

P.O. BOX 213. GOODWOOD. S.A.,5034, AUSTRALIA. TELEPHONE(08) 272 0966. PRICE: AUS. $2.50, NZ. 3.00, U.K. £1.50

** ABOUT MICRO-80 **

EDITOR: IAN VAGG
ASSOCIATE EDITORS:
SOFTWARE LEVEL I : MICHAEL SVENSDOTTER
SOFTWARE LEVEL II: CHARLIE BARTLETT
HARDWARE : EDWIN PAAY

MICRO-80 is an international magazine devoted entirely to the Tandy TRS-80 microcomputer and the Dick Smith System 80/Video Genie. It is available at the following prices (all prices shown in Aus.$ except for U.K. prices which are in pounds Sterling).

12 months subscription	Aus.	$24.00
	NZ.	$36.00 (Airmail)
	Hong Kong	$46.00 (Airmail)
	U.K.	16.00 pounds
Single Copy	Aus.	$2.50
	N.Z.	$3.50 (Airmail)
	Hong Kong	$4.25 (Airmail)
	U.K.	1.50 pounds
Months programs on cassette	Aus.	$3.50
	N.Z.	$4.00 (Airmail)
	Hong Kong	$4.50 (Airmail)
(at present available from Australia only)	U.K.	$4.75 (Airmail)
12 months subscription to magazine and cassette	Aus.	$60.00
	N.Z.	$78.00 (Airmail)
	Hong Kong	$88.00 (Airmail)
	U.K.	$41.00 (Airmail)

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-80 is to publish software and other information to help you get the most from your TRS-80, System 80 or Video Genie and their peripherals. MICRO-80 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-80 or System 80 to earn some extra income is included in every issue.

** CONTENT **
Each month we publish at least one applications program in Level I BASIC, one in Level II BASIC and one in DISK BASIC (or disk compatible Level II). We also publish Utility programs in Level II 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 **
The programs and other articles in MICRO-80 are published in good faith and we do our utmost to ensure that they function as described. However, no liability can be accepted for the failure of any program or other article to function satisfactorily or for any consequential damages arising from their use for any purpose whatsoever.

***** CONTENTS *****

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

AUSTRALIAN OFFICE AND EDITOR:-
MICRO-80 PO BOX 213, GOODWOOD, SA 5034

UK SUBSCRIPTION DEPT.:-
24 WOODHILL PARK, PEMBURY, TUNBRIDGE WELLS, KENT TN2 4NW.

Printed by:-
Shovel & Bull printers, 312 A Unley road, HYDE PARK, SA 5061.

Published by MICRO-80, 284 Goodwood Rd., Clarence Park, ADELAIDE.

***** EDITORIAL *****

** TRS-80 MODEL 1 DOOMED IN THE U.S.A. **

The F.C.C. has finally pulled the plug on the TRS-80 Model 1 in the U.S.A. Radio Shack will no longer be permitted to manufacture the Model 1 for the U.S. market after December 31 this year. The reason is that the TRS-80 Model 1 (in common with virtually all other U.S. personal computers except the T.I.) exceeds permissible levels of T.V. interference (TVI). The F.C.C. (Federal Communications Commission) is the American regulatory body responsible for radio transmissions etc. It had already granted Radio Shack a stay of execution but now has set a final deadline. No doubt, the new TRS-80 Colour and TRS-80 Model 3 are Radio Shack's answer to the F.C.C. ruling. The Model 3 in its basic L1/4K form is $200 dearer in the U.S.A. than the Model 1 ($699 compared with $499) but Tandy presumably believe that this price will not be a handicap.

What of Australia? We put the question to a Tandy spokesman and received the following answer. The Australian Government has no specific limits on TVI and would only respond to complaints from T.V. viewers. So far, these have been negligible. The Model 1, therefore, has an unlimited future in Australia. No Australian price is available for the Model 3 which is not due to be released here until May or June 1981.

** MICRO-80 IN THE U.K. **

At the time of writing, the first advertisement for MICRO-80 has been on the newstands for about two weeks. Already, we have received hundreds of enquiries. The response was a little overwhelming and there were a few delays in sending out magazines as our U.K. stocks were exhausted before reinforcements could arrive from Australia. We welcome our new readers in the U.K. For the moment, our news is mainly about the Tandy scene in Australia and the U.S.A., but we hope we will soon be receiving letters from our U.K. readers and are eager to appoint a U.K. correspondent to keep us all abreast of happenings there. If you would like to assume that role (for which we are prepared to pay) please write directly to us in Australia.

** LEVEL II ROM REFERENCE MANUAL IN THE U.S.A. **

We at MICRO-80 are pretty proud of Eddy Paay's Level II ROM Reference Manual which we first published in May of this year. Not long ago, we sent a copy to one of our U.S. suppliers who had expressed interest in it. His verdict, it is the best book of its kind he has read and he has seen virtually all of those on sale in the U.S.A. Well done Eddy - there are several hundred happy owners of the manual in Australia who heartily endorse that sentiment. Incidentally, Eddy continues to uncover useful routines in the Level II ROM. The latest allows you to reset MEMORY SIZE during execution of a program. This will be fully explained in the BLINKING CURSOR program he has written to be published next month. Finally, we are happy to announce that Eddy has joined MICRO-80 as our first full-time staff member. Already, he has cleared MICRO-80 PRODUCTS backlog of orders which built up whilst we were at the Melbourne show, and we are now able to offer the 48 hour turnaround on all in-stock items which has been our aim for so long.

** TANDY ROAD SHOW **

We had heard a number of good things about the Tandy Road Show which has been working its way around Australia for several months, now. So, when it came to Adelaide, we made sure we were one of the visitors.

The main show was in a large room which generally does service as a disco. Surrounding the main room were three or four smaller rooms with specialist displays such as educational programs, word processing, books, software, etc. In the main room there were long tables with chairs and a large number of TRS-80's. We counted 20 and there could have been more divided fairly evenly between Model 1 and Model 2 machines each with a line-printer attached whilst the Model 1 machines had a single disk drive to bring them ore or less in line with the Model 2 units. Visitors were invited to sit at the tables, near the microcomputers (generally there were no more than 2 or 3 people per computer).

The whole show lasted for 45 minutes or so. It started with a film, obviously an American sales film but not objectively so. The film explained the background to the development of the microcomputer and had a number of users giving testimonials which sounded pretty genuine. After the film, visitors were invited to participate in a demonstration routine, using the computers in front of them. It was obvious at the session I attended, that very few visitors had any experience with microcomputers and were delighted if a little aprehensive, at the opportunity to use them. The programs on disk took the users through a number of simple business type routines such as a mailing list and some accounting. Instructions were given verbally by a presenter visually on slides projected onto a screen, and there was a veritable army of Tandy personnel to assist anyone having difficulty. The computer nearest me was a Model 1 and suffered from losing power every time someone kicked the labrynth of plugs and sockets on the floor under the table. Willing Tandy staff quickly put things to rights although the sequential nature of the demonstration program made it difficult for our group to catch up with the rest of the flock.

After the demonstration in the main room had finished, we were invited to visit the "sideshows" in the smaller rooms. I paid brief visits to the word processing demonstration and also the educational demonstration. Unfortunately, I did not have as much time to spend in the latter as I would have liked.

In summary, the Tandy Road Show was well run and organised. There has probably never been its like in Australia before and I am sure it gave many people a worthwhile introduction to microcomputing. Well done Tandy.

** TANDY'S PRICING POLICY **

Last month we were pretty critical of Tandy's pricing policy, at least in the area of the Lower Case Mod. We have had no response to those comments but wonder if a change might be on the way. Having the interest in word processing that we do, we had a good look at the new daisy-wheel printer being sold by Tandy for $2280. We thought it looked familiar and so it proved to be. It is a RICOH printer until recently sold by D.S. Electronics for $3300. That is an amazing turnaround. Usually we expect D.S. Electronics to undersell Tandy by a comfortable margin.

Just in case that was an accident, we also looked at the new Line Printer VI priced at $1495. It is a nice printer and we were not surprised to hear that one Tandy manager left the meeting at which the printer was announced shaking his head and muttering that it is $1000 too cheap. Let's hope this is just the beginning and that all Tandy's prices will soon follow suit.

** 8TH INTERNATIONAL COMPUTER CONFERENCE **

The 8th International Computer Conference was undoubtedly the greatest collection of computer equipment Australian has ever seen. Exhibitors covered the whole gamut from very large to very small. MICRO-80 was there, of course, and we were pleased to meet many of our interstate readers and many new readers too. We also forged a number of new contacts with suppliers which will be to the benefit of all of us over the next few months.

** ON PRESENTATION OF MICRO-80 **

This month we have changed to the new form of presentation which we promised last month. Typesetting is being carried out on the Olivetti ET-121 typewriter with MICRO-80 conversion and copy is photographically reduced by 80% As a result, we fit more editorial matter into fewer pages. This helps us to contain costs and we believe it makes for a more readable magazine. As an extra bonus, we have a more interesting cover. John Edwards, one of our readers in Adelaide who is studying art, was set a project to design a new cover for a low-budget magazine. He chose MICRO-80. John called with his design on the same day that we were looking for someone to redesign our voer! The result you can see for yourself. We hope you like John's efforts and our new format.

** DEADLINES **

Try as we will, we have been unable to keep to our production deadlines, as most readers will be aware. We know that this has caused some displeasure and we are doing everything possible to correct the situation. The problem is, of course, lack of resources. We have 1200 subscribers at present and it is not yet possible to appoint a full-time Editor, although that day is not too far away. (If you fancy a job as Editor of MICRO-80 you could write to us in the near future so we will have a list of interest, qualified people to choose from when the time comes). In the meantime, we are finally resolved that we will put quality ahead of deadlines and that is why your magazine comes later. Have no fears though, it will arrive and every issue will be produced!

** BMON AND THE NEW 2 CHIP ROMS **

Tandy's new 2 chip ROM's have some "improvements" over their earlier counterparts. One is the use of the SHIFT/DOWN ARROW to give a CONTROL function. Unfortunately, access to BMON is via SHIFT/DOWN ARROW. Those with the new, 2 chip ROM's will find that they can access BMON by typing SHIFT/DOWN ARROW/Z at the same time. This gives the same ASCII code as SHIFT/DOWN ARROW did with the earlier ROM's.

** SUBSCRIPTION RENEWALS **

We are nearing the end of our first year of publication and many subscriptions will be due for renewal after next month's issue. If your subscription has almost expired, why not send in for a renewal now and avoid the risk of missing any issues?

- 0000000000 -

***** TIP OF THE MONTH *****

For situations requiring password protection to determine access levels.
Open a disk file, as OPEN,"O",1,"PASSWORD,XYZZY" or any other password you may care to name and then CLOSE.
In your main program, use the following....

```
10  CLS
20  INPUT"PASSWORD";D$
30  ONERRORGOTO70
40  D$="PASSWORD,"+D$
50  OPEN"I",1,D$
60  GOTO NNNN1
70  CLS
80  PRINT"PASSWORD REJECTED"
90  PRINT"RESTRICTED ACCESS ONLY"
100 RESUME NNNN2
NNNN1 CLOSE.....
NNNN2 CLOSE.....
```

- 0000000000 -

***** G.T. BASIC - PART THREE of a continuing series - by Peter Hartley. *****

Before we kick off this month's session, allow me to thank you all for your letters of praise, criticism, etc. I cannot emphasize enough just how important these are in ensuring the continuing success of this series, which is intended to take the raw beginner by the hand and lead him through the normally messy business of mastering BASIC.

Some of you have asked for a full listing of the Microsoft BASIC commands and functions, together with their syntax and appropriate explanations. Microsoft did this in one of their recent publications (almost $200!!!) and it would fill no less than 17 complete issues of MICRO-80!!!

For Level I users, there really isn't any better guide than the Tandy Level I handbook, which is also sold as a separate publication under the title "BASIC Computer Language - it's easier than you think."

For Level II and System 80 users, Learning Level II, by David Lien (who also authored the Level I manual) is highly recommended. If you use both these books together, you will find that the Level II book has a whole heap of pages for cutting out and pasting into the Level I book - and this makes a great deal of sense.

I'm not prepared to compare the Tandy/System 80 versions of BASIC with other brands, except to say that I am still unable to find better. I find the Model II to be somewhat inflexible, by comparison with the System 80/TRS-80 Model I, and the soon-to-be-released Model III is alleged to be compatible with Model I, but I'll know more about that before the November Edition (hint!!!).

Now to business...

Last month I threw you all in at the deep end with the FOR/NEXT loop, showed you how to tidy up your PRINT statements, and just introduced the DATA STATEMENT with this little program.

```
 10 POKE 16553,255 (Level I users leave this line out.)
 20 DATA 1, 3, 5, 7, 9, 2, 6, 8, 10
 30 CLS
 40 PRINT "ODD NUMBERS BETWEEN 1 AND 10 ARE ";
 50 FOR K=1 TO 5
 60 READA
 70 PRINTA;
 80 NEXT            (Level I users use   N.K    )
 90 PRINT "EVEN NUMBERS ARE ";
100 FORK=1 TO 5
110 READA
120 PRINTA;
130 NEXT            (Level I users use   N.K    )
140 END
```

Those of you who tried this will have discovered that it didn't look quite as tidy as we would have liked - running the entire output into one continuous sentence.
To tidy it up, now add two new lines.

```
 85 PRINT "."
```

and

```
135 PRINT "."
```

** Before any of the old lags in the back row start putting pen to paper to send in their complaining letters - "YES, I KNOW that there are 27 better ways to do this, without using DATA statements, but I'm teaching about DATA statements, so save your stamps for later."

Now we'll take it to pieces...

Line 10 is a DATA statement debugger, explained in September.
20 is a DATA statement - a list of DATA to be used by the program.
30 Make with the eraser...
40 PRINT exactly what is in between the quotes, and leave the cursor at the end - more to come.
50 (See September) Do the next bit 5 times.
60 READ the next item of data, and store it in variable A.
70 PRINT what's stored in variable A
80 (See September) Back to 50.
85 PRINT it and start a new line (No comma, semicolon or other PRINT MODIFIERS).
90 These lines have the same
to effect as lines 50 through
135 85 so carry on as before..
140 Finished with engines...

Because this little routine is so annoyingly inefficient, this would seem as good a time as any to introduce the concept of SUBROUTINES.

As every instruction line has a number, and because the well-trained '80 is obedient and long-suffering, we do not have to waste either our time, or the computer's valuable memory space, writing out identical statements over and over. In the example above, for instance, lines 50 to 85 are doing exactly the same jobs as lines 90 to 135. To save space and effort, then, we need to understand the instructions...

GOTO GOSUB and RETURN

Without typing NEW (Because we are going to come back to the last program in a moment) type in the following muddle, ending each line with (ENTER) or (NEW LINE).

```
200     CLS
210     GOTO 260
220     GOSUB 320
225     GOSUB 320
230     GOTO 280
240     PRINT "E";
250     GOTO 220
260     PRINT "H";
270     GOTO 240
280     PRINT "O"
290     END
300     PRINT "YOUR COMPUTER IS R/S."
310     PRINT "TAKE IT BACK AND SPOIL SOMEONE'S DAY."
315     GOTO 300
320     PRINT "L";
330     RETURN
```

Now type RUN 200 and press (ENTER) or (NEW LINE) (Another surprise... you can have more than one program in memory at one time, just tell ol' '80 where to start...)

If you didn't make any mistakes with your typing, the screen should read

```
HELLO

READY
)_
```

If you follow the listing through, you'll see that the computer jumped about quite a bit, and visited the SUBROUTINE starting at 320 twice. When the computer encounters a GOTO, it jumps to the new location and continues as if nothing had happened. When it encounters a GOSUB it makes a note of where it is prior to the jump. Then it jumps and carries on there until it encounters a RETURN. This makes it look up the previously noted address, and it jumps back TO THE FIRST STATEMENT or INSTRUCTION FOLLOWING THE GOSUB.

** ALWAYS end a SUBROUTINE with RETURN **

```
##### ########## ########## ########## ########## #####
##### G.T. BASIC RULE #3                          #####
##### IF YOU NEED TO EXECUTE THE SAME SERIES OF   #####
##### INSTRUCTIONS FROM MORE THAN ONE PLACE IN    #####
##### YOUR PROGRAM - ** ALWAYS USE A SUBROUTINE   #####
##### ########## ########## ########## ########## #####
```

**** One of the three known bugs in MICROSOFT '80 type BASIC relates to the use of subroutines. As this particular bug only rears its ugly head within the most convoluted of programs (with multiple nested - inside one another - SUBROUTINES and FOR/NEXT LOOPS) we'll not need to worry about that for quite a few months.

Now, we already know that to clear an entire computer-full of program we use the command NEW. We also know that to delete a single line we have only to type in the line number and hit (ENTER) or (NEW LINE). Now try...

```
DELETE 200-330
(ENTER) or (NEW LINE)
LIST
(ENTER) etc
```

Now let's make the remaining program a bit more efficient... Type in the following, again ending each line with (ENTER) or (NEW LINE).

```
        50 GOSUB 100
        60 PRINT "."
        70 PRINT "EVEN NUMBERS ARE ";
        80 GOSUB 100
        90 END
       135        (This is simply to delete the old line 135)
       140 RETURN
```

RUN (ENTER)

O.K?

* STRING VARIABLES
In the first of these sessions I explained just how many NUMERIC VARIABLES there were in both Level I and Level II. Now its time to learn about STRING VARIABLES. These are used to store STRINGS. You'll recall from last month's piece that a STRING is something to be printed - not something upon which mathematical miracles are to be performed. So while the number 123 would normally be stored in a NUMERIC VARIABLE, "123 Goodwood Road" could only be stored in a STRING VARIABLE. Note, though, that you could still store "123" in a STRING VARIABLE, but you could not perform any math on it without doing a few other things to it first - and we'll discuss these sometime in the future.
Level I has only 2 STRING VARIABLES - called A$ and B$, and each of these can store a string of up to 16 characters and spaces.
Level II has over 950 STRING VARIABLES - A$ through to Z$, and then AA$ through to AZ$, to A0$ and on to A9$, then to BA$ and on and on and on until you eventually get to Z9$. STRING VARIABLES in Level II can hold up to 255 characters and spaces, but for practical purposes we should try to limit ourselves to 248 characters.
The '80 normally knows that you are referring to a STRING VARIABLE only by the $ symbol in its name - again, more on this aspect later.

So, now let's make a few more changes to our ever-changing progam... again, remember to hit (ENTER) or (NEW LINE) after each line...

```
        15 A$="ODD NUMBERS "
        17 B$="EVEN NUMBERS "
        40 PRINT A$;
        70 PRINT B$;
        RUN
```

O.K?

Nearly finished for this month - just a few more changes...

```
        15
        17
        20 DATA ODD NUMBERS ,1,3,5,7,9,EVEN NUMBERS ,2,4,6,8,10
        40
        70
       100 FOR K=1 TO 6
       110 READ A$
       120 PRINT A$;
RUN (ENTER) etc.
```

The lines with nothing except for the line number are just to delete the old lines. What we've done is to get the computer to treat all the DATA as STRINGS, by using A$ instead of plain old A in lines 110 and 120.

Now we really are in the home stretch....

```
        90 GOTO 50
RUN (ENTER) etc.
```

Level I produced HOW?
Level II produced a mysterious message... O/D ERROR IN 110????

What happened was that there just wasn't any more DATA to read. One solution is to add more DATA to line 20, however, try...

```
        90 RESTORE: GOTO 50
RUN (ENTER) etc.
```

The RESTORE statement makes the computer reset its internal records so that it thinks it hasn't read any DATA at all, yet.

- Continued on page 14.

```
 MEMORY SIZE?
RADIO SHACK LEVEL II BASIC
 READY
 )PRINT MEM
  9452
```

by E. R. Paay

You have just plugged in your 16K expansion kit to convert your 4K machine to 16K. Eagerly, you switch the machine on, the familiar MEMORY SIZE? question pops up. You type ENTER and wait for the machine to reply with RADIO SHACK etc. Now you type PRINT MEM and instead of 15572 coming up it says 9452. However if the MEMORY SIZE? question is answered with, 32000, for example, everything appears to be okay. Does this sound familiar ? It must do to some because several people have written in to us relaying similar problems after installing their 16K RAM's. One reader rang the dealer from which he bought the chips and was told that this occasionally happens with TRS-80's, and not to worry about it. This is, of course, a load of rubbish.

Provided the dip shunts are programmed properly, then the wrong answer to PRINT MEM means that one of the RAM's is faulty. Let us once and for all clear this matter up so that everyone knows why this happens.

The computer during initialization checks how much memory is available to the system. Knowing the algorithm it uses to determine this will reveal how the above problem arises. Depending on the answer given to the MEMORY SIZE ? question, the computer will do one of two things :

1. If MEMORY SIZE ? is answered with ENTER, the computer will start checking memory from about 434DH. It takes a byte from memory and stores it in a register for safe keeping, it then places a value in this location and reads it back. If the value read back is the same as the value placed in that location then it will RESTORE the original value and moves on to the next byte. This continues until a memory location turns up that returns a different value than that written into it. The computer assumes that this is the highest memory location available and sets the memory size accordingly.

2. When the MEMORY SIZE ? question is answered with a value, all that the computer does is test that one location to see if it actually exists.

This means that if you have a bit in one of the RAM's which is faulty, the computer will use this as the memory size value. It only takes one bit in one location in memory to be faulty for this to happen. However if a memory size value is specified by the user then this bad memory location will not be recognised and the computer will set the memory size as requested, providing that the address specified exists. So if this happens to you after installing a new ram kit, insist on replacement RAM's, don't let any one tell you not to worry about it! (MICRO-80 PRODUCTS points out that they have never had this problem with any of their RAM chips because they sell only PRIME spec. 200 nanosecond chips - Ed.)

- 0000000000 -

!!!!! HELP !!!!! From the new Software Editor - Charlie Bartlett

It's all our fault, but the magazine has grown so quickly that some things have got out of hand. First the good news, EVERY contributer who ever submitted software for publication has now had an individual file opened containing information such as name, address and phone number (where supplied), the programs he has sent in, whether they were accepted or rejected and whether he has been paid,etc ect.. In fact everything related to that contributor. This information is also stored in a computer information storage system enabling information about a contributor to be found in seconds.

NOW THE BAD NEWS

Now that the files are sorted out, I find that, in some cases, the information I have is incomplete. Somewhere along the line some letters, tapes and forms have been lost. If you sent us software more than six weeks ago and have not recieved a response or if we have not sent your tape back, please write to me c/o MICRO-80, P.O. BOX 213, GOODWOOD, SA 5034.

May I also ask that if your writ ing is not very clear will you PLEASE - PLEASE - PLEASE , either make a tape of REM statements or print by hand clearly but best of all use a typewriter. Also, several contributors are sending their programs in on cheap and nasty tapes. Please don't, they will not load most of the time and only end up being sent back which wastes your time and mine. As I said at the start most of the problems are our fault, help us sort it all out quickly and write and tell me what it is that we have not done. Once the missing info is placed on file there will never need to be a letter like this again.

STICK WITH US GANG, WE ARE TRYING TO GET THE BUGS OUT

- 0000000000 -

** BETTER BYTES ** A potpourrie of bits of this and bits of that..

Conducted by Peter Hartley.

Readers comments, hints, suggestions, etc., are requested for this column.

* MICROSOFT BASIC COMPILER

As previously reported, the Microsoft Basic Compiler for the TRS-80 Model 1 has been withdrawn, rewritten, and some 30 odd copies of the Mark II version are currently out on test. However, some of our readers have copies of the original version, and are experiencing some problems.
The solutions are always to be found in the manual, but often buried away, so here are a few tips.

If you have two drives and can use a 40 track drive for Drive 1, do so. You can get all three parts of the Compiler (BASCOM/CMD, L80/CMD and BASLIB/REL) onto the one 40 track diskette, and then use the Compiler as though it is on a 3 drive system. Use a minimum system diskette in Drive 0, with just the .../SYS and BASIC/CMD files on it, together with your BASIC program that is to be compiled, as an ASCII file.

The Compiler doesn't see a downward line feed in the middle of a line - not even inside quotations! So, use separate PRINT statements.

The Compiler treats STOP and END both as END - and sometimes ENDs when it gets there. So put them AT THE END!

The Compiler will only process lines of up to 127 characters. The simplest way to run it is with separate statement lines.

If your BASIC program contains ... NNN INPUT"PRESS (ENTER) TO CONTINUE";D$ you will have problems - responding to an INPUT with an (ENTER) generates a null string, which will be rejected by the compiled software, generating a "REDO FROM BEGINNING" message. So, use a D$=INKEY$:IFD$=""THENnnnELSE... statement.

If you either don't use any switches when you compile, or if you use the -4 switch, the use of any blanks around reserved words may create syntax errors. Equally, using the -5 switch will generate errors if these blanks are NOT there!

Using the -S switch will produce more efficient code, by forcing all strings to be stored at the front of the generated software.

As the compiler does understand WHILE... WEND and SWAP, you may as well use them for efficiency.

* ON EARTHING YOUR SYSTEM *

The Hartley system recently started showing some rather unusual symptoms - massive increase in spontaneous re-booting, slight increase in the notorious TRS-80 video shimmer in the upper left of the display, A.C. type tingles from the Disk Drives when also touching a good earth point - and the oddest symptom of all (and the first I noticed) - a 100 cycle hum coming right across the top of one of our local radio stations on about 970 KHz.
A little investigating showed that the hum was being picked up from all the house mains wiring, and that it stopped when the Video Display was switched off or disconnected from the C.P.U.
Earthing the Disk Drives (Pertec) cured the A.C. tingles, but nothing else, so I started with the multi-meter and discovered about 26 volts A.C. with respect to earth coming from the Video Display Cable. Discussions with one of the local experts suggested that using a high-impedence FET type meter would show that the 26 volts was really 240 volt mains!.
Fortunately I had a copy of the official manual for the VDU (one of the original "square" types which use an auto-transformer to drop 240 volts down to 110 volts a la U.S.A. style). This unit has a small board designed to produce total electrical isolation between the CPU and the VDU. The board draws a 5 volt supply from the CPU, and has its own earth, quite separate from the floating earth of the VDU. The two systems are interfaced through an opto-isolator chip.
The reason for the isolation technique used is that the main T.V. circuit used (a) does not have an earthed chassis and (b) can demonstrate a live chassis.
Joining the two earths is a sizeable capacitor. The purpose of this appears to be to maintain a stable relationship between the RELATIVE levels of the two earths, with the aim of producing a more stable video picture.
Removing this capacitor cured all the problems except the video stability - which got a great deal worse.
Replacing it with a new one of the same value left the symptoms exactly as before.
Examining the circuit of both the VDU and the CPU showed that the VDU earth (computer side) was common with the CPU earth. Rather than hunt around for a new opto-isolator (the one used is somewhat rare!) I finally settled for using the mains earth within the VDU. This terminates by soldered joint on a massive tag-strip deep in the bowels of the VDU. Running a heavy and well-insulated cable from the earth tag to the computer-earth on the video-isolation board proved the perfect simple fix and earths the entire system. Result - only one spontaneous reboot in the last six weeks (from a system that is up and running some eighteen hours a day!), stable video, no more hum from the radio and a total loss of the A.C. tingles.
Be warned though - the design of the old VDU demands that you do not go poking around inside unless it is unplugged from the mains. Also be careful of the large capacitors and the tube which can carry a sizeable charge for a surprisingly long time.

- 0000000000 -

** PROBLEM CORNER **

Q. Can you show me a simple way to use a Disk based Menu?

A. Sure.

First file your programs with simple, alphabetical file-names and forget about using extensions. Eg.: A, B, C, etc..

Now, set your menu program up like this:-

```
10  CLS
20  PRINT"                MENU
                        -=::::=-"
30  PRINT"
                  A)      RING BELLS
                  B)      BLOW TRUMPETS
                  C)      FLASH LIGHTS
                  D)      FLY A KITE
40  PRINT"SELECT OPTION    ";
50  D$=INKEY$
60  IFD$("A"ORD$)"D"GOTO50
70  RUND$
```

Now, if one of your secondary programs is also a menu, you set it up the same way, options being single, key letters, starting from A. Give this and similar sub-programs two-letter names, eg.:- CA, CB, CD, CE, CF etc.

In this case, LINE 70 becomes:-

```
70  D$="C"+D$:RUND$
```

Q. I need a simple method off saving a screen-full of data to disk......

A. O.K.

Method 1

```
10  CLS:OPEN"O",1,"SCREEN"
20  D$=INKEY$
30  IFD$=""GOTO20
40  IFD$()"@"THEN PRINTD$;:GOTO20
50  M=15360
60  FOR K=-1 TO 8
70  D$=""
80  FOR L=M TO M+127
90  D$=D$+CHR$(PEEK(L))
100 NEXT
110 PRINT#1,D$
120 M=L
130 NEXT
140 CLOSE
```

And to get it back, just open the file and:-

```
   FOR K=1 TO 8:INPUT#1,D$:PRINTD$;:NEXT
```

METHOD 2 (for NEWDOS+)

Protect some high memory; format from DOS is BASIC 1,32767

```
10  CLS
20  D$=INKEY$
30  IFD$=""GOTO 20
40  IFD$()"@"THEN PRINTD$;:GOTO20
50  M=&H3C00
60  C=&H8000
70  FOR K=0 TO &H400
80  POKE C+K,PEEK(M+K)
90  NEXT
100 CMD"DUMP SCREEN/CIM (START=X'8000',END=X'8400)
110 END
```

And to recover it later, use:-

```
10  CLS
20  CMD"LOAD SCREEN/CIM"
30  M=&H3C00
40  C=&H8000
50  FOR K=0 TO &H400
60  POKEM+K,PEEK(C+K)
70  NEXT
80  END
```

Neither of these programs is all that fast - a ml. routine woould be much faster - something for someone out there to develop.

- 0000000000 -

COMPUTERISED RTTY - "MACROTRONICS AND THE SYSTEM 80" - by Ron Collins.....VK5RY

(Many of our readers are Amateur Radio Operators and we have had a number of requests for articles of particular interest to "Hams". We hope this article by Ron Collins will go some way towards satisfying those requests. In the meantime, if you have some experience in this field, why not write an article for us? - Ed.)

Let me begin by pointing out that I am not an "expert" on RTTY, or anything else for that matter. This article is to let others know of my observations with regard to the Dick Smith System 80 computer, being coupled with the Macrotronics M80 for RTTY operation.

I have been involved with RTTY for about seven years; my main interest being DX and Contest operating. Just recently it has become apparent that a lot of stations in the United States and Europe are using equipment that is obviously computer controlled. When asked what their equipment is, a large number answer, "such and such a micro-computer" with an "M800 RTTY interface". Having toyed with the idea of applying the same idea here, I investigated the cost of various computers but found them mostly to be cost-prohibitive. Then, lo and behold, the System 80 was released, and put micro-computing within my grasp. I purchased one and played with same for a few weeks, having great fun in Basic. Now, I thought, the time was ripe to sound out the possibilities of RTTY and proceeded to write to the owner of "Macrotronics" in the States, giving him the pinout of the SYSTEM 80, to see if it was compatible with the Tandy-80 version of the M800. His reply was, "if it is the same as the TRS-80 it will work". According to the information that I had to hand about the SYSTEM 80 (TRS-80 compatible?), there should have been no reason why the Macrotronics gear would not work, so I ordered an M80, (the heart of the M800 system, which consists of a very nice P.C.B. and the software to get on RTTY, also a very comprehensive booklet). The price in Australia is a bit of a con, $180 odd, but if the price of a unit ordered from U.S.A. plus the import duty etc. is taken into consideration, I suppose that it is not too bad.
Now comes the "CRUNCH".
Looking at the connections between the TRS-80 and the M80, the requirements are:-
4 address connections (0-3), 4 data connections (0-3), a Ground and a "funny" called "Not IN".

I supposed that a "Not IN" would be an "OUT" but there was not an output on the SYSTEM 80 called either of these names. Checking with the TRS-80 service manual, readily available, I found that the missing signal was, in fact, a combination of "Not IORQ" and "Not IORQ", being Read bar and IN/OUT Request Bar, the two being combined in a gate to give a resultant "Not IN" at the output. Checking the SYSTEM 80 circuit diagram, it was found that this signal is available and is used internally for another function but is not brought out to a pin on the connector. Therefore, it is only a matter of joining pin 6 of I.C.15 (74LS32), (see Fig.1), to unused Pin 32 on top of the edge connector with about two inches of hook-up wire, and we now have the necessary "NOT IN" function. Having connected up the address, data and NOT IN connections, along with a Ground and 9 volts A.C., we can get the "THING" up and running. Load the program from the cassette supplied, taking note of the loading instructions, and we find that we need to type in an instruction called "MEMORY SIZE".

"MEMORY SIZE"?, what's that? we say; What I did not know at that time, was that the "READY" that comes on the screen when the SYSTEM 80 is first switched on, is in fact a request from the computer to tell it what the memory size is going to be.
Now we can really get going!
Switch off SYSTEM 80, switch on again after 15 seconds, up comes READY?, type in 30000 (as per the instructions in the booklet), press NEW LINE and the READY pops down to the bottom.
Type SYSTEM, it answers with a *?; type M80, press the cassette "play" button and then the "NEW LINE" key and the cassette begins to load with the flashing asterisk.
The first load is in machine language, so when it is loaded, an "*?" will appear.
Press the "Break" key, then type the CLOAD command and the cassette will begin to load the BASIC part of the program. The normal "Ready?" will show that the program is in the memory O.K., so we can now "RUN" the program.

At first, there is displayed the name of the program and the author then the screen is blanked and we really get into the "Nitty-Gritty". The screen tells you what to do and when to do it (referring frequently to the handbook) and we are amazed. However, we find that to get out of one mode and into another, we need to have a function called a "CLEAR". What is a "CLEAR"? you might well ask because you will find that the SYSTEM 80 ain't got one, no matter how hard you try all the other keys. Not to despair; check the System 80 circuit and we find that it has the same keyboard matrix as the TRS-80 but some of the keys have been omitted from the System 80 keyboard for some crazy reason; (also, you will find that another function is required, that is the "forward arrow" function to alter some of the mode activities of the program).
So now we look at the keyboard; all that room with holes and all, to fit more keyswitches but just you try and find some that will go in those holes and leave room for connections. For the "CLEAR" key, I made use of the right hand "SHIFT" key, just by cutting the print and connecting to the right points on the print by following the matrix connections on the underside of the P.C.B. HA! you will say, that's easy but I'll bet that you did not know that although the numbering on the connections says "1 to 20" or at least, "1 to 10", it is in fact, the other way around; so don't forget to number from the other end (pin 1 is in fact pin 20). If you were dumb enough to flop the keyboard over from right to left (as I was), to get to the underside of the P.C.B., you will now be aware of the fact that all of the wires that were plugged into the connector on the top of the

boards, have now fallen out. Just you try to get them all back in in a hurry; the trick is to flop the P.C.B. over from front over to the back and they will hopefully stay in place.

Whilst we are looking at the bottom of the P.C.B., how about looking at the rest of the unused lines.
Apart from the "CLEAR", there are six other function keys that may as well be used (see Fig.2). A few "low profile" keyswitches were obtained locally and these were mounted on a piece of fibreglass (and another piece underneath that for insulation), this assembly being mounted as a next row up from the existing top row but this, of course, is up to the individual. Now we can really get going: connect up the TS820 and the DT600, follow the "MENU" and get the gear on the air.
What?, no receive of RTTY characters?.
Upon checking with a logic probe, it was found that the EX/OR gates in I.C9 on the M80 board were not switching, even with 0-5v transitions on the input (H) and (7), (RS232 in). Changing the OPTO load resistor to 500 ohms from 1K cured this problem and signals were bunging characters onto the T.V. screen but it is all "gibberish". What's up now?, of course, looking at the "MENU" we find that if we press the '/' key, the Mark and Space will be inverted. Hey Presto; it works; but the QSO is in Italian; it seems that we can't win!

I have yet to try the "THING" on C.W. but messing about with the keyboard in the C.W. mode makes transmitting a breeze and the buffer makes typing ahead very easy, so that your fist sounds like a truly professional operator and the bug is getting to me that makes me want to go "Brass-bound" again. For further information on the System 80, there are some "News Letters" (numbers 12 and 17) available from Dick Smith shops for free.

Having seen the M800 in operation here, I have placed one on order and if I can work out all the things that are available on it, I might get the DXCC on RTTY yet.
One other thing comes to mind; as all my RTTY is at TTL levels, I found that the print from Pin 1 of the relay (GROUND) had to be cut and the Pin 1 connected to 5 volts to give me the +5 necessary for the correct switching sense.
Best wishes and good RTTY/Computer operating.

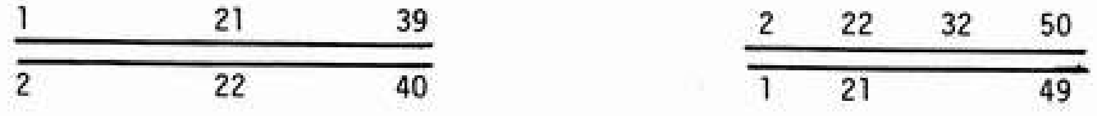
PS. In the Keyboard Matrix, (Fig.2), note the extra characters.

Lines:-	3-16	Left Square Bracket
	1-16	Back Slash
	5-16	Right Square Bracket
	2-16	Inverted Vee
	4-16	Underline
	8-19	Clear
	2-19	Advance Arrow (Actually "Tab 16 Characters").

Line 20 to any other line only duplicates other characters, so is of no use.

** EDGE CONNECTOR PIN CONNECTIONS **

	TRS-80	SYSTEM-80
GND	37-8-29	1-2-49-50
A0	25	10
A1	27	7
A2	40	9
A3	34	8
IN(Bar)	19	32 (After modification Fig 1)
D0	30	15
D1	22	14
D2	32	12
D3	26	16



Edge connectors from rear....

FIG 1 $\overline{IN}$ MODIFICATION FOR SYSTEM 80

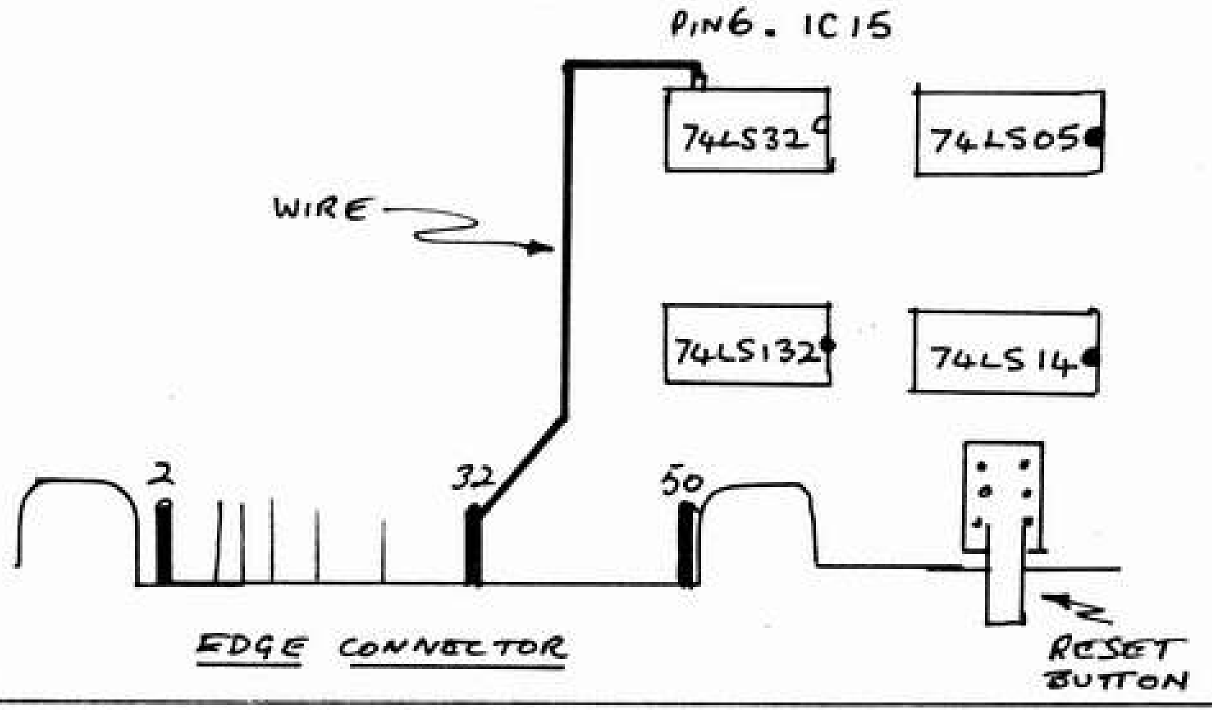
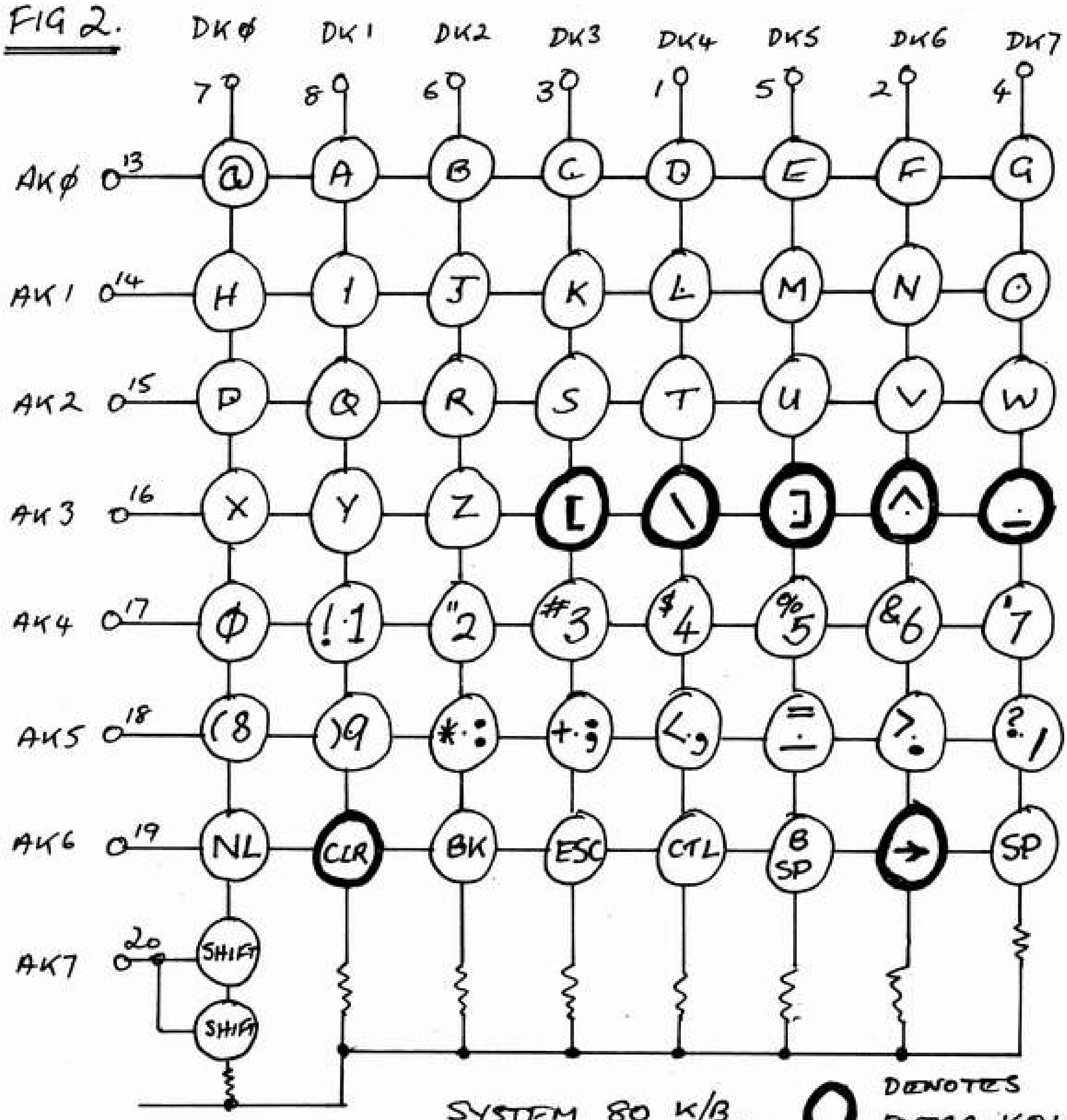


FIG 2.



	DKØ 7	DK1 8	DK2 6	DK3 3	DK4 1	DK5 5	DK6 2	DK7 4
AKØ 13	@	A	B	C	D	E	F	G
AK1 14	H	I	J	K	L	M	N	O
AK2 15	P	Q	R	S	T	U	V	W
AK3 16	X	Y	Z	[	\	]	^	_
AK4 17	Ø	!1	"2	#3	$4	%5	&6	'7
AK5 18	(8	)9	*:	+;	<,	=-	>.	?/
AK6 19	NL	CLR	BK	ESC	CTL	B SP	→	SP
AK7 20	SHIFT SHIFT							

SYSTEM 80 K/B. ○ DENOTES EXTRA KEYS

****** TAPE & DISK DATA FORMATS ******

by: E. R. PAAY.

This section discusses how the computer stores programs on tape or disk. For cassette I/O, port 255 is used. This port is used for several functions such as character display format and turning the cassette recorder on and off. For this discusion however we are only concerned with bits 0 and 1. These bits can cause the cassette port to output either a zero, positive or negative voltage. Each pulse that is written to tape consists of a negative and positive voltage transition. On tape, we have a leader of 256 zero bytes followed by the actual program. Data is saved on tape serially with clock bits interlaced with data. For each bit of a byte, a clock pulse is written first then, depending whether the data bit is 1 or 0, a bit will or will not be written to tape, folowed by the last clock pulse (i.e. no data pulse for a bit 0).
The read routine in ROM used by CLOAD and SYSTEM uses the following procedure to load a byte from tape :

1. Loop until a clock bit is found.
2. Delay 2 mS.
3. Scan cassette port to see if bit is 1 (pulse present) or 0 (no pulse present).
4. Shift bit into register.
5. Do a short delay to give data pulse time to decay.
6. Repeat sequence until 8 bits have been read.

From this we can conclude that if there happens to be a noise spike on the tape before a valid clock pulse the computer will lose its place and read in a "bad" bit. If the clock pulse however becomes stretched far enough so that it is still there after the 2 mS delay the computer will read this as a 1 and cause a bad bit to be shifted into the byte it is working on. If it hasn't become obvious yet, the clock bit is necessary to keep the computer synchronised to the tape while reading is in progress. For example if a tape is created on one tape recorder and read on another and the two recorders have a slight difference in tape speed, the computer can still read the tape as it will be able to catch up every time it looks for the clock bit. If the clock bits were not there and all data bits were dumped with 2mS delays between them, the computer would very quickly loose sync. If the tape was replayed on a faster recorder causing the gaps between bits to shorten to, for example 1.99mS, after 100 bits the computer would be out by 1mS, whereas with clock bits the computer would be out by a constant amount of 0.01mS for each bit. The values given are of course only an example and don't necessarily relate to real values. I might point out at this stage that the new TRS-80's have a modified ROM in them which overcomes the first problem mentioned by introducing a greater delay after scanning the cassette port for a data bit etc. System 80 users are stuck with the old version of the level II interpreter which is part of their cassette problems.

Data is handled in much the same manner on disk, the recording technique is different however. The recording technique used is a constant amplitude signal which is frequency modulated, people who understand B H curves and magnetic flux will recognise the advantage of this technique. The stringy floppy device uses this technique also and users will know how reliable they are.

There are two different types of formats for tape:- BASIC CSAVEd tapes and SYSTEM compatible tapes, they have the following formats:-

** BASIC PROGRAM FORMAT **

1. 256 byte leader consisting of zeros.
2. A5H sync byte.
3. BASIC header consisting of three bytes D3H.
4. Program name consisting of one character.
5. A two byte long line pointer which points to the start of the next BASIC program line.
6. BASIC line number in hexadecimal.
7. Actual program line up to 240 bytes long.
8. Zero byte to signal end of line.
9. New line starts here OR two zero bytes to mark end of program.

** FORMAT OF SYSTEM TAPE **

1. LEADER as for BASIC program.
2. Sync byte A5H.
3. Header code 55H.
4. Name six characters long padded with blanks if less than six characters long.
   Note that the SYSTEM command only checks the first character of the name, the last five characters are never checked. That had me fooled for a while, too.
5. Block header code C3H.
6. Block length from 1 to 256 bytes long.
7. Two byte starting location of block in memory.
8. Block of data stored here.
9. One byte checksum which is the total of the starting address and all data added together, ignoring any overflow.
10. The block from item 5 on is repeated here for the next block OR a one byte END of program code 78H is stored here.

** BASIC PROGRAMS STORED ON DISK **

1. Basic header code FFH.
   The rest is exactly the same as for tape starting with the line pointer.

- 0000000000 -

***** READERS' REQUESTS *****

This column is a regular feature of MICRO-80. In it, we list all those articles, programs, etc. requested by our readers. We try to work our way through the list as time permits, but if you would like to contribute an article or program, look here for an idea of our readers preferences.

** ARTICLES **

File handling on the '80
Description of the functions performed by the Expansion Interface
Reviews of '80 compatible printers
Reviews of commercially available software (including that produced by us!)
Reviews of commercially available hardware
How to SAVE onto Disk, programs such as Analogue clock and Touchtype
A master index to the appropriate sections in the Tandy Manuals in Level I, Level II, DOS etc.

** SOFTWARE **

GAME OF LIFE relocated to start at 7000H
A m.l. program to enable the break key to work like RESET when using an expansion interface
Stock market program
Horse racing system
"Files" program modified for 48K system
Morse code decoder
Sub-routine Forum
Program to "SET" non-graphical symbols

** HARDWARE **

RS232 printer interface
Interfacing the '80 to external hardware
Review on the performance of line filters
How to convert a black and white T.V. to a monitor
Review of high resolution graphics mods.
Real Time clock
Radio Teletype/Morse interfacing
RFI (Radio Frequency Interference) suppression

- 0000000000 -

** LNW RESEARCH EXPANSION INTERFACE BOARDS **
$79.95+$2.00 p&p

There has been so much interest shown in the LNW Research expansion project described in last month's MICRO-80 that we have decided to import the boards to Australia. These boards are the basis for a high quality, reliable expansion interface for the TRS-80 (or the SYSTEM 80 via our SYSPAND-80 adaptor). Initially, we will offer the boards with their comprehensive construction manual. As soon as possible thereafter, we will offer modular kits of components and fully built up and tested units. Deliveries of boards will commence in late November. Hurry! They are sure to be very popular.

Continued from Page 6 (G.T. BASIC)

Now, I'll leave you all to puzzle that one out and this month, set you some homework!

The problem is to write a simple DECIMAL to HEXADECIMAL conversion routine using only those instructions that we have discussed so far in this course.
It need only convert a single number 0 to 15, and next month I'll use a version of this as the basis for bigger and better things.
I'm afraid that there aren't any prizes for the best solution - even if I could persuade the Editor to put up an expansion interface or three, I'm obliged to start on Part 4 immediately.

- 0000000000 -

** SUPERDOS FORUM ** NEWDOS 80 examined in detail by Peter Hartley.

I ended the last edition of Better Bytes by expressing the view that I would rave on at length about NEWDOS 80. Herewith...the first ravings.

While we at MICRO-80 appreciate that a number of you out there have not yet invested in this amazing new DOS, we have decided to adopt SUPERDOS as our standard DOS. The primary reason for this is the great wealth of power that SUPERDOS vests in the '80 - and we like POWER!!!! We believe that any serious Disk user should have the best. All our readers keep permanent files of MICRO-80 anyway and as YOU all want the BEST, we've decided to start this column now so that when you all eventually get the best, you'll have a file of info to get you going.

First off, let me say that SUPERDOS is upward compatible with both TRSDOS 2.3 and NEWDOS 2.1. However, many of the SUPERDOS features are not downward compatible. That is to say that any software written for TRSDOS or NEWDOS 2.1 will run under SUPERDOS, but software that uses the advanced SUPERDOS features will crash under the earlier DOS's.

SUPERDOS is very very fresh off the drawing board and experience suggests that this would mean bugs galore. APPARAT, however, has an excellent reputation in this regard and the manual comes with a list of minor corrections that are simply applied using SUPERZAP - one of the more famous utilities that forms a part of every APPARAT DOS.

The first thing to do on receiving your SUPERDOS is to make a BACKUP, so that the corrections, or ZAPS can be applied to the backup and not to your $149.00 master disk. However, NEWDOS 80 does not have a backup utility, using instead an amazingly flexible COPY command. A part of the SUPERDOS keeps a record of how many tracks you have on which Drives. On the master disk, this is set to 35 tracks for all. If your Drives are 40 track or anything else other than 35, you will have to reset this information before commencing your original backup.

1.) Locate a nice new blank diskette and then instal your precious master disk into Drive 0 and boot DOS with the reset BUTTON.

2.) If you are only running one 35 track drive, type
```
COPY,:0,TO,:0,mm/dd/yy,PASSWORD,UBB (ENTER)
```

3.) If you are running two or more 35 track drives, insert a new diskette into Drive 1 and type
```
COPY,:0,TO,:1,mm/dd/yy,PASSWORD (ENTER)
```

4.) If you are running one 40 track Drive only, type
PDRIVE :0,0,DTC=40 (ENTER). Now reboot the system with the RESET BUTTON and then type
```
COPY,:0=35,TO,:0,mm/dd/yy,PASSWORD,UBB (ENTER)
```

5.) If you have a mix of 35 and 40 track drives, you will have to make a decision as to which will be used for Drive 0. Because the full SUPERDOS is so comprehensive, and therefore so big, my personal preference would be to use a 40 track Drive for Drive 0. However, it is a simple matter to make smaller "minimum system SUPERDOS system diskettes" to use in certain types of application. (For example, a minimum system BASIC SUPERDOS will leave some 35+ grans free on a 40 track Drive). If drive 1 is 35 track and Drive 0 is 40 track, type
PDRIVE :0,0,DTC=40 (ENTER).
Now reboot the system with the RESET BUTTON, and type
COPY,:1,TO,:0,mm/dd/yy,PASSWORD (ENTER)
**Make sure to transfer your master disk to drive 1 when the system responds with "PRESS 'ENTER' WHEN SOURCE DISKETTE MOUNTED IN DRIVE 1" and to put a new diskette into Drive 0.
If drive 1 is 40 track and Drive 0 is 35 track, type
PDRIVE :0,1,DTC=40 (ENTER).
Now reboot the system with the RESET BUTTON, and then type
COPY,:1=35,TO,:0 (ENTER).
** Again make sure that the source disk is moved to Drive 1 at the appropriate time.

6.) If all your Drives are 40 track, type
PDRIVE :0,0,DTC=40 (ENTER) PDRIVE :0,1,DTC=40 (ENTER)
and so on, once for each drive. Now, insert a blank diskette into Drive 1, reboot the system with the RESET BUTTON and then type
```
COPY,:0=35,TO,:1,mm/dd/yy,PASSWORD (ENTER)
```

You should now have a new master SUPERDOS diskette, formatted with the correct number of tracks for Drive 0 and can put your original carefully away in a safe place.
(Perhaps I should explain that the DTC or Diskette Track Count data is only loaded into the system on boot-up, hence the need to re-boot after changing the data with PDRIVE which writes directly to the disk.)

Now comes the matter of applying those corrections. SUPERZAP is now a machine language program, so from DOS, type SUPERZAP (ENTER) and study sections 11 and 13 of the manual. You may well find that most of the listed corrections have already been applied but this is not always certain, and the time spent in checking will be well worth-while.

The only problem area that I have personally experienced with SUPERDOS is with Electric Pencil, where Pencil cannot list its own directory of files created with the nw DOS. It can still locate them for loading and still saves them correctly. I'm sure that APPARAT will soon have an answer for this.(This fault is not present on the MICRO-80 copy of NEWDOS 80 which works perfectly with electric pencil. Perhaps Peter's copy is missing one or two "zaps" - Ed.)

Finally, a brief word about the FORMAT utility of SUPERDOS. This does not lock out faulty tracks! The overall concept of SUPERDOS is that it was to be a "Programmers DOS" and it is assumed that every user will understand the structure of the Granule Allocation Table used by the Directory. In other words the operator is required to lock out defective diskette tracks using SUPERZAP.
If your drives are in A1 shape and if you are using high-quality diskettes, this presents no problem, as the normal error rate will be minimal.
However, for those of you out there who would prefer that the "system" looked after this aspect of formatting a new diskette, I have prepared a listing of a simple BASIC package that gives this facility, and would be pleased to supply it to anyone who sends $2 and a self addressed envelope.

Next month we'll have a look at the CHAINING FACILITIES of SUPERDOS.

- 0000000000 -

***** '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 that we can publish details. Owners of System '80s are welcome at all the groups.

BRISBANE: Contact: Mr. Lance Lawes,
Tel: Home (07)396 2998
Bus.(07)268 1191 Ext.15
MEETINGS 1st Sunday of the month at 2 p.m. at 21 Rodney St. Lindum, 4178.

MELBOURNE: EASTERN SUBURBS - 1
Contact: Mr. John Fletcher, 89 0677 bet. 9-4

EASTERN SUBURBS - 2
MEETINGS: 3rd Wednesday of the month at Kingswood College, 355 Station St. Box Hill.

FRANKSTON: PENINSULAR GROUP
(Vic.) MEETINGS: 2nd Tues. of the month (except Jan.)
Contact: M.G. Thompson (03)772 2674

GEELONG: *GEELONG COMPUTER CLUB
MEETINGS: 2nd Tues. of the month at TYBAR Engineering, Hampton St. Newtown.

Contact: The Geelong Computer Club,
P.O. Box 6, Geelong, 3220.

DARWIN: Contact: Tony Domigan, P.O. Box 39086,
Winnellie, N.T. 5789.

ADELAIDE: Contact: Rod Stevenson, 51 5241 bet. 9-4

* THE GEELONG COMPUTER CLUB
The Geelong Computer Club held its Annual General Meeting on the 4th Oct. 1980 at which a formal club constitution was adopted. The club has an active membership of approximately 20 persons and is open to all members of the public interested in computers, there is a yearly subscription fee of $5.00.

The club has an active TRS-80 and System 80 subgroup with eight members owning these systems and experiencing the ectasy and agonies of operating them. From addresses of letters sent to readers request section of MICRO-80 it appears as though there are other TRS-80 and System 80 owners in the Geelong Region who could benefit by being in and/or known to the club.

- 0000000000 -

***** NEW FROM MICRO-80 PRODUCTS *****

We have a host of new products and software for TRS-80/SYSTEM 80 microcomputers.

MICROTEK MEMORY EXPANSION/PRINTER MODULES

A number of our readers expressed interest in the memory expansion module we developed earlier in the year. Unfortunately, there was not sufficient interest to make it worthwhile continuing with this rather expensive project, particularly as a completely assembled, tested unit would have had to sell for more than $200.

Fortunately, MICROTEK in America had the same idea and with the larger population and greater resources, has been able to produce a similar unit which we can sell for only $149. We are disappointed that we were unable to produce our own design economically but are happy to offer you the MICROTEK unit. It comes fully assembled and tested in a metal cabinet of similar size to the Tandy expansion interface so that you can sit your monitor atop it. The MICROTEK unit intercepts the Tandy power supply cable and has its own internal rectifiers and regulators. As well as providing sockets for 32K of memory, it also has a Centronics parallel printer port which enables you to run a line printer or the Olivetti ET-121. Full instructions for connecting to your system and adding memory chips are included.

MICROTEK MT32-A	0K	$149
MICROTEK MT32-B	16K	$208
MICROTEK MT32-C	32K	$262

Incl.p&p. The memory supplied is our prime, branded 200 nanosecond kit. Fitting is included in the price if memory is ordered with the module.

***** SYSPAND-80 FOR SYSTEM 80 COMPUTERS *****

$119 incl p&p.

The SYSPAND-80 adaptor designed by MICRO-80 PRODUCTS is now available in fully assembled and tested form. Kits and kit prices will be announced next month. The SYSPAND-80 connects to the rear of the SYSTEM-80 and provides a Centronics parallel printer port plus the Tandy bus so that you can run a Stringy Floppy, the MICROTEK memory expansion module or even a Tandy expansion interface/disk drives etc. from your SYSTEM-80. The SYSPAND-80 has its own plug-pack power supply and internal regulator so it imposes no additional load on the SYSTEM 80 power supply.

40 PIN EDGE CONNECTORS $12.30 +50c. p&p

These difficult to obtain connectors are essential if you are working on a hardware project requiring connection to the TANDY bus. The connector attaches to a 40 way ribbon cable and plugs straight on to the back of your TRS-80.

***** GREEN SCREEN SIMULATOR *****

$19.95 incl. p&p

The GREEN SCREEN SIMULATOR is made from a deep green perspex, cut to fit your monitor. It improves contrast and is much more restful to the eyes than the normal grey and white image. All the editorial staff of MICRO-80 (including Scrooge Mc. Hartley) are now using GREEN SCREEN SIMULATORS on their own monitors. Please make sure to specify whether you have an old (squarish) or new (rounded) style monitor when ordering. Not available for Dick Smith monitors.

##

OLIVETTI ET-121 DAISY WHEEL TYPEWRITER FOR '80 only $1995

##

MICRO-80 PRODUCTS has developed an interface to drive the OLIVETTI ET-121 typewriter from a Centronics parallel printer port. The ET-121 is a quiet, high speed (20 cps) auto-correcting, electronic typewriter utilising a Z 80 microprocessor. We have mounted an interface inside the typewriter which does not interfere with manual operation but enables it to be driven as a printer from a Centronics port such as is available in the SYSPAND-80, MICROTEK module or TRS-80 expansion interface.

This makes an ideal combination for the home or office where you need the flexibility of a quiet, reliable, superior typewriter which can double as a correspondence quality printer when driven from your microcomputer.

This issue of MICRO-80 has been typeset using the MICRO-80 OLIVETTI.

Deliveries commence in December. Secure your order by sending a $50 deposit or write asking for further technical details.

DATA SEPARATOR $37.00 plus $1.20 p&p.

When Tandy designed the TRS-80 expansion interface, they did not include a data separator in the disk-controller circuitry, despite the I.C. manufacturer's recommendations to do so. The result is that many disk drive owners suffer a lot of Disk I/O errors. The answer is a data separator. This unit fits inside your expansion interface. It is supplied with full instructions and is a must for the serious disk user.

DISK DRIVE HEAD CLEANING DISKETTES

$29.00 plus $1.20 p&p

Disk drives are expensive and so are diskettes. As with any magnetic recording device, a disk drive works better and lasts longer if the head is cleaned regularly. In the past, the problem has been, how do you clean the head without pulling the mechanism apart and running the risk of damaging delicate parts. 3M's have come to our rescue with SCOTCH BRAND, non-abrasive, head cleaning diskettes which thoroughly clean the head in seconds. The cleaning action is less abrasive than an ordinary diskette and no residue is left behind. Each kit contains:

- 2 head cleaning diskettes
- 1 bottle of cleaning fluid
- 1 bottle dispenser cap.

****** DISKETTE BREAKTHROUGH ******

40 TRACK NASHUA DISKETTES (in library case)
SINGLE SIDE/SINGLE DENSITY $45.00 box of 10

40 TRACK VERBATIM DISKETTES
DOUBLE SIDE/DOUBLE DENSITY $59.00 box of 10

77 TRACK VERBATIM DISKETTES
SINGLE SIDE/SINGLE DENSITY $59.00 box of 10

All prices include p$&p.

NEWDOS 80 IS IN STOCK. (only $149)

This long-awaited disk operating system has now arrived and has already been delivered to many eager customers around Australia. It is every bit as good as it was cracked-up to be and MICRO-80 has decided to standardise on it for its own systems. Here are just a few of the many things you can do with NEWDOS 80 which is upward compatible with TRSDOS and NEWDOS + (i.e. TRSDOS and NEWDOS + programs will run on NEWDOS 80 but the reverse is not necessarily so.)

* New BASIC commands that support variable record lengths up to 4095 bytes long.
* Mix or match disk drives. Supports any track count from 18 to 96. Use 35, 40, 77 or 80 track 5" mini disk drives, 8" disk drives, OR ANY COMBINATION!
* An optional security boot-up for BASIC or machine code application programs. User never sees "DOSREADY" or "READY" and is unable to "BREAK", clear screen or issue any direct BASIC statements including "LIST".
* New editing commands that allow program lines to be deleted from one location and moved to another or to allow the duplication of a program line with the deletion of the original.
* Enhanced and improved RENUMBER that allows relocation of subroutines.
* Powerful program chaining.
* Device handling for routing to display and printer simultaneously.
* DFG function; simultaneous striking of the D, F and G keys will allow user to enter a mini-DOS to perform some DOS commands without disturbing the resident program.
* Includes Superzap 3.0 (improved, machine language version of Superzap) and all Apparat 2.1 utilities).

NEWDOS +

35 TRACK VERSION	$99.00 incl. p&p
40 TRACK VERSION	$110.00 incl. p&p

NEWDOS by Apparat, is the third generation disk operating system for your TRS-80. NEWDOS corrects over 70 errors and omissions in TRSDOS and Disk BASIC yet the two are completely compatible. Going from TRSDOS to NEWDOS is kike going from Level I to Level II, more power, more convenience, greater speed. NEWDOS + includes the following utilities:

* Editor-assembler for Disk
* Disassembler (Z80 machine code)
* LM offset - allows transfers of any system tape to a Disk file-automatically relocated.
* BASIC 1 - Lets you convert your computer back to Level I
* LVIDKSL - Saves and loads BASIC 1 programs to disk
* SUPERZAP - display/print/modify any location in memory or on disk.
* RENUMBER BASIC program.

and much more.

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

Please use order form on page 36

***** MICROPOLIS 77 TRACK DISK DRIVES *****

These fabulous MICROPOLIS disk drives have more than double the storage capacity of the standard 35 track drives.

** DD-7S only $775 incl. p&p **

77 track MICROPOLIS drive complete with cable for four drives, power supply, chassis and includes NEWDOS 80.

** DD-7 only $649 incl. p&p **

Same as above but no cable or NEWDOS 80.

** DC-4 only $45 incl. p&p **

4 drive connector cable - suitable for any disk drives.

**** MPI DISK DRIVES *****

MPI is the second largest manufacturer of disk drives in the world. MPI drives use the same form of head control as on 8" drives and consequently, they have the fastest track-to-track access time available: 5msec. All MPIr drives are capable of single or double-density operation. (Double-density operation requires the installation of a special PC board in the expansion interface. This board is not yet available in Australia). As well as the single head 40 track disk drive, MPI also make a dual-head 40 track disk drive. A dual-head drive is almost as versatile as two single-head drives but is much cheaper.

Our MPI drives are supplied bare, set up to operate with TRS-80. They can be used bare or a simple cabinet made up from sheet metal (single-drive cabinets will be available in early December - price $10.00). They require the use of a separate power supply giving 5 volt at 0.7 amp and 12 volt at 1 amp. All drives are sold with a 90 day warranty and service is available through MICRO=80 PRODUCTS.

MPI SINGLE HEAD DISK DRIVE only $339 incl. p&p.
MPI DOUBLE HEAD DISK DRIVE only $449 incl. p&p.

AVAILABLE SOON:-
MPI 80 TRACK SINGLE HEAD DRIVE only $499
MPI 80 TRACK DOUBLE HEAD DRIVE only $599

Ring first to check availability.

FLOPPY DOCTOR AND MEMORY DIAGNOSTIC
(by MICRO CLINIC) $29.95 plus 50c. p&p.

Two machine language programs on a diskette together with manual which thoroughly test your disk drives and memory. There are 19 possible error messages in the disk drive test and their likely causes are explained in the manual. Each pass of the memory tests checks every address in RAM 520 times, including the space normally occupied by the diagnostic program itself. When an error occurs the address, expected data, and actual data are printed out together with a detailed error analysis showing the failing bit or bits, the corresponding IC's and their location. This is the most thorough test routine available for TRS-80 disk users.

** PROGRAMS FROM CREATIVE COMPUTING **

** ADVENTURE PROGRAMS **

ADVENTURELAND (L2/16K) $14.95 + 50c. p&p.
Try to find and take treasures as you explore a fantasy world. The computer acts as your puppet and carries out your two word commands. Sometimes you will need special objects to do certain things, often a little magic is necessary. Absorbing and challenging.

THE COUNT ADVENTURE(L2/16K) $14.95 + 50c. p&p.
In this adventure, you awaken in a bed in a castle in Transylvania. You don't know why you are there but you'd better solve the puzzle before it's too late. Just as enthralling as ADVENTURELAND but blood thirstier!

** GAMES **

AIR TRAFFIC CONTROLLER(L2/16K) $9.95 + 50c.p&p
One of the hottest selling games in the USA, you are the Air Traffic Controller and the monitor is your radar screen. Bring down the aircraft safely and avoid mid-air collisions.

Z CHESS (L2/16K) $19.95 + 50c. p&p.
Seven levels of ability, contains all standard moves including castling and En Passant captures. It can play either black or white and its versatile board set-up mode allows specific positions to be played as desired.

SPACE GAMES (L2/16K) $13.50 + 50c. p&p.
4 Space Games including ULTRA-TREK, ROMULAN, STARWARS and STARLANES. Fast, real-time graphics.

STRATEGY GAMES (L2/16K) $9.50 + 50c. p&p.
5 Strategy games including TUNNEL VISION (find your way out of a 3-D maze), EVASION - avoid the deadly snake), JIGSAW (put the puzzle together), THE MASTERS (Golf on the '80 for up to 4 players), MOTOR RACING (Compete against the computer at Indy or the Grand Prix).

GRAPHING PACKAGE (L2/16K) $9.95 + 50c. p&p
A setof 6 utility programs which allow you to draw BAR GRAPHS, GRAPH CARTESIAN COORDINATES, carry out POLAR GRAPHING, PARAMETRIC GRAPHING, LINEAR REGRESSION and PARABOLIC REGRESSION.

** SCOTCH BRAND COMPUTING CASSETTES **

These super-quality cassettes are now back in stock.

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

***** 16K MEMORY EXPANSION KIT *****

REDUCED TO ONLY $59.00 !!!!

These are prime, branded, 200 ns (yes, 200 nanosecond) 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 in the TRS-80 - these cost an additional $4.00. All kits come complete with full, step-by-step instructions which include labelled photographs. No soldering is required. You do not have to be an experienced electronic technician to instal them.

** PROGRAMS BY MICROSOFT **

EDITOR ASSEMBLER PLUS (L2/16K)
REDUCED TO ONLY $37.50 + $1.20 p&p.
A much improved editor-assembler and debug/monitor for L2/16K TRS-80 or SYSTEM 80. Assembles directly into memory, supports macros and conditional assembly, includes new commands-substitute, move, copy and extend.

LEVEL III BASIC $59.95 plus $1.20 p&p.
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)

ADVENTURE ON DISK $35.95 plus $1.20 p&p.
This is the original ADVENTURE game adapted for the TRS-80. The 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)

DISK EDITOR ASSEMBLER
SAVE $30 ONLY $107 plus $2.00 p&p.
Supports Macros, linking loader, editor, cross references. See Assembly language programming article in August MICRO-80 for further details. (Requires 32K One Disk)

BASIC COMPILER $208 plus $2.00 p&p.
Converts Disk BASIC programs to machine code, automatically. A compiled program runs, on average, 3-10 times faster than the original BASIC program and is much more difficult to pirate. Note: MICROSOFT have temporarily withdrawn this program for reworking to make it less memory-hungry. The revised version should be available about the end of NOVEMBER. All existing owners will receive updated versions at no cost to them. Don't wait though, send in your orders now and be near the front of the queue.

***** BOOKS *****

LEVEL II ROM REFERENCE MANUAL $24.95+$1.20 p&p
Over 70 pages packed full of useful information and sample prorams. Applies to both TRS-80 and SYSYEM 80.

TRS-80 DISK AND OTHER MYSTERIES
$24.95+$1.20 p&p.
The hottest selling TRS-80 book in the U.S.A. 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 and is a perfect companion to any of the NEWDOS's.

** OTHER PROGRAMS **

INFINITE BASIC BY RACET (32K/1DISK)
$49.95 + 50C. P&P.
Full matrix functions - 30 BASIC commands; 50 more STRING functions as BASIC commands

GSF/L2/48K $24.95 + 50c. p&p.
18 machine language routines including RACET sorts.

BUSINESS ADDRESS AND INFORMATION SYSTEM
(48K/DISK) $24.95 + 50c. p&p.
Allows you to store addresses and information about businesses, edit them and print them out.

***** SOFTWARE BY AUSTRALIAN AUTHORS *****

All our software is suitable for either the SYSTEM 80 or the TRS-80 (with the exception of Me Think it Micro Mazin vol.2).

** UTILITIES **

S-KEY by Edwin Paay $15.95 plus 50c. p&p

S-KEY is a complete keyboard driver routine for the TRS-80 and becomes part of the Level II basic interpreter. With S-KEY loaded the user will have many new features not available with the standard machine.

S-KEY features:

* S-KEY provides an auto-repeat for all the keys on the keyboard. If any key is held down longer than about half a second, the key will repeat until it is released.
* Graphic symbols can be typed direct from the keyboard, this includes all 64 graphic symbols available from the TRS-80/SYSTEM 80
* S-KEY allows text, BASIC commands and/or graphics to be defined to shifted keys. This makes programming much easier as whole commands and statements can be recalled by typing shift and a letter key.
* Because S-KEY allows graphics to be typed directly from the keyboard, animation and fast graphics are easily implemented by typing the appropriate graphics symbols directly into PRINT statements.
* S-KEY allows the user to LIST a program with PRINT statements containing graphics, properly. S-KEY does this by intercepting the LIST routine when necessary.
* S-KEY allows the user to list an updated list of the shift key entries to the video display or line printer.
* S-KEY can be disabled and enabled when required. This allows other routines which take control of the keyboard to run with S-KEY as well.

Each cassette has TRS-80, DISK and SYSTEM 80 versions and comes with comprehensive documentation.

BMON by Edwin Paay $19.95 plus 50c. p&p
THE ULTIMATE HIGH MEMORY BASIC MONITOR
L2/16-48K

Our own personnel refuse to write BASIC without first loading this amazing machine language utility program into high memory! BMON Renumbers; Displays BASIC programs on the screen while they are still loading; tells you the memory locations of the program just loaded; lets you stop a load part-way through; merges two programs, with automatic renumbering of the second so as to prevent any clashes of line numbers; recovers your program even though you did type NEW: makes one program invisible while you work on a second (saves hours of cassette time!); lists all the variables used in the program; makes SYSTEM tapes; lets you Edit memory directly...the list goes on and on. Cassette comes with 16K, 32K and 48K versions, ready to load. Can anyone affort NOT to have BMON?

** EDUCATIONAL **

RPN CALCULATOR (L2/16K & 32K) $24.95+50c p&p.

Give your computer the power of a $650 reverse polish notation calculator with 45 functions and selectable accuracy of 8 or 16 digits. The main stack and registers are continuously displayed whilst the menu is always instantly accessible without disturbing any calculations or register values. The cassette comes with both the 16K and 32K versions, the latter giving you the additional power of a programmable calculator. Comes with a very comprehensive 15 page manual, which includes instructions to load and modify the 32K programmable version to run in 16K. Whether for business or pleasure, this package will prove invaluable, and turn your '80 into a very powerful instrument.

TOUCHTYPE (L2/4K) $19.95 + 50c. p&p.

An interactive, 22 lesson typing course which uses the computer's keyboard and screen to teach you to type rapidly and accurately and, a massive cassette data dump to control your progress. The computer checks for accuracy, and sets timed exercises to check your progress. If you have to look at each key before you press it, or only use two fingers, then this program, plus a little perseverance, will do some amazing things to your typing speed.

** GAMES **

U BOAT $7.50 plus 50c. p&6p.

Real time simulation at its best! Comes with working sonar-screen and periscope, a full rack of torpedoes, plenty of targets, working fuel and battery meters, helpful Mothership for high-seas reprovisioning and even has emergency radio for that terrible moment when the depth charges put your crew at risk. Requires Level II/16K.

SPACE INVADERS $7.50 plus 50c. p&p.

Much improved version of this arcade favourite with redesigned laser and cannon blasts, high-speed cannon, 50 roving drone targets, 10 motherships and heaps of fun for all. Level II with 4K and 16K versions on this cassette.

MMM- vol.1 and vol.2. $7.50 ea. plus 50c. p&p.

Two, three-game cassettes from that master of TRS-80 graphics, Charlie Bartlett. Vol.1 brings you INDY 500, an exciting road race that gets faster and faster the longer you play, SUBHUNT in which your warship blows up unfortunate little submarines all over the place, and KNIEVEL (as in motorcycle, ramp and buses).

Vol.2 (not System 80 compatible) containss TANK, a two player battle of speed and skill, THIEF (steal an old man's hoard of loot then escape from the labyrinth before he gets you), and SHOOTOUT (a Mexican style high-noon for two).

DALEK CHASE (L2/16K and up) $7.50 + 50c. p&p.

An absorbing game in which you save Dr. Who from pursuing Daleks. The better you play, the harder it gets (and vice versa). Infinite range of play levels from HARD to UTTERLY IMPOSSIBLE, and comes complete with messages of praise and mediocre insults. (If you are really bad enough at this game, it actually refuses to play any more!) ENTHRALLING FUN FOR ALL.

***** E.S.F. FORUM - by Charlie Bartlett *****

Judging by the number of ESF's that MICRO-80 PRODUCTS is selling, I get the idea that I am not the only one who thinks they are the best thing since the invention of the TRS-80.
This month I would like to tell you something about the ESF that is not described in the manual. Hang on to your hat,
YOUR ESF PACKS DATA BEFORE WRITING TO WAFER.
For example, using a (YUK) cassette to achieve a reasonable save or load data time you would have to write something like this.

```
105 FOR X=1TO25
110 PRINT#-1,S(X),S(X+25),S(X+50),S(X+75)
115 NEXT X
```

The ESF operating system, however, packs data automatically. All you need to type is this:-

```
 90 DIM S(256)
100 CLS:@OPEN1
105 FOR X=1TO256
110 @PRINT S(X)
115 NEXT X
125 @CLOSE
```

Now, load the DATA I/O program and RUN this BASIC program.

As the program goes through the loop the ESF stores up data until it has 255 bytes and then switches on the drive.
You can observe this happening if you make the following change to Line 105:-

```
105 FOR X=1 TO 256:PRINT@ 64,"SAVING ITEMS";X
```

Now RUN the revised program. X will build up to 52 before the ESF turns on.

Thats all for this month, There may be no ESF Forum next month due to yours truly spending a short time in hospital, so that should give all you new ESF owners time to send in some programs to do SUPER DUPER things on this incredible machine.

***** A QUICK REVIEW OF THE TRS-80 POCKET COMPUTER. by C. Bartlett *****

I recently got my hands on the new TRS-80 pocket computer, there was not enough time for a full checkout but I can tell you right now that it is worth every cent. It will operate just like a normal calculator in one mode but becomes a fully fledged computer in another mode. It has nearly all the Level 2 commands available plus one function called BEEP which, as the name implies, can be written into the program to cause the computer to beep any number of times; for instance the statement BEEP(10) will cause ten beeps.

Print statements operate a little differently in that they stay on the display line until the ENTER key is pressed. To compensate for this there is another statement called PAUSE which makes the display stay on view for about one second. Programs in memory are not lost when the computer is switched off which is a very handy feature. Another useful feature is the KEY WORDS function. This gives a user definable key rather like Eddy Paay's S KEY program.

The computer comes with a large operating manual and the computer itself is provided with a wallet for protection. There were many more things it could do but as I said I didn't have it for long as the borrowed unit had to be returned in time for a stocktake.

For the beginner who cannot afford a proper TRS-80 this is an ideal stepping stone in that the slight differences in the BASIC are actually less than going from Level 1 to Level 2. For the experienced programmer, a great thing to carry around in your pocket for those ideas which suddenly come in the middle of the day but which you have usually forgotten by the time you get home.

All in all a good buy for anyone, well done Tandy.

- 0000000000 -

** MICRO-BUGS **

Last month we omitted to publish the explanatory text for the program ESCAPEE. Here it is now.

** ESCAPEE **
In this game you have to find an escaped criminal, he is cornered in an underground garage, which has 64 pillars (8 x 8). You have to gues which pillar the escapee is hiding behind to catch him. After each guess you'll either be told the number of columns or the number of rows you are away from the escapee. If you guess is incorrect, the escapee will move one pillar horizontally or vertically on the plan (the plan is redrawn after each guess just in case it is messed up when you have a guess).

The subroutine for drawing the plan starts at line 700 and ends at line 760. It is this subroutine which is called each time you have a guess. If line 350 is dissected it reads like this - INPUT column ' row : IF (column is less than 1) or (column is greater than 8) or (row is less than 1) or (row is greater than 8) GO TO line 340. At line 360 the column and row are compared with the current position of the escapee. If both of these are correct, then the program jumps to line 600. The formula at line 465 tells the computer to print an asterisk at the current guess.

- 0000000000 -

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

** SPACE INVADERS - L1/4K**
That's right, it's another inkey simulator and loads of fun too. There are four types of alien space craft, if you shoot down the entire fleet the game continues, after the reinforcements have grouped, the aliens continue to regroup until all of your ships have been shot down.

Line 8 GOS.830:IFP.(112,1)G.8 is the same as:
8 GOS.830:IFPOINT(112,1)=1 GOTO8...
Line 92 IFI.(Z/2)=Z/2 etc. is testing Z,if Z is an even number then line 92 is used, if not the program falls through to the next line.
Lines 150 to 160 may be of interest, I've condensed the original version slightly and increased the speed (these lines draw the three structures between your ship and the aliens).

```
ORIGINAL VERSION
150 F.Z=38T.32S.-1:F.X=18T.102
155 IFX=34X=52
160 IFX=68X=87
165 S.(X,Z):N.X:GOS.92:N.Z
NEW VERSION
150 F.Z=38T.32S.-1:F.Y=18T.102S.34:F.X=YT.Y+15
160 S.(X,Z):N.X:N.Y:GOS.92:N.Z
```

(Note: there was a full explanation of how the INKEY function can be simulated in a Level I machine in MICRO-80 Issue No. 3 - February 1980. - Ed.)

SPACE INVADERS - L1/4K BY BR. P. VAN EEKEN

```
1 C.:A(0)=106:F.Z=1T.50:GOS.92:N.Z:H=0
2 P:A.320;"YOUR CONTROLS ONLY WORK WHEN BOTH LINES ABOVE ARE VISIBLE.
3 P."PRESS ] OR ↑ TO MOVE YOUR SHIP LEFT OR RIGHT.
4 P."PRESS SPACE BAR TO SHOOT THE ALIENS BELOW.
5 Y=30:F.X=586T.842S.128:P.A.X;"SCORES";Y;"POINTS.":Y=Y+30:N.X
6 X=5:Y=28:GOS.835:Y=34:GOS.825:Y=39:GOS.815
7 P.A.938;"PRESS CLEAR TO START";:GOS.850
8 GOS.830:IFP.(112,1)G.8
9 C.:I=0:F=0:E=0:T=R.(4)+3:D=T-1:G.99
```

(No we haven't left anything out! The next line number really is 92 - Ed.)

```
92 IFI.(Z/2)=Z/2P.A.24;"SPACE INVADERS":G.94
93 P.A.24
94 R.(0,3):RET.
99 IFH>29P.A.983;"TOP SCORE =";H;
110 X=0:F.Z=1T.7:W=100*(7+X*17):A(Z+21)=W/100
120 A(Z)=W+12:A(Z+7)=W+17:A(Z+14)=W+22:X=X+1:N.Z
140 F.Z=1T.21:GOS.999:J=I.((Z-.5)/7)+1:ONJGOS.815,825,835
143 GOS.92:N.Z
150 F.Z=38T.32S.-1:F.Y=18T.102S.34:F.X=YT.Y+15
160 S.(X,Z):N.X:N.Y:GOS.92:N.Z
170 F.Z=21T.90S.34:F.X=ZT.Z+9S.3:R.(X,32):N.X:N.Z
188 N=1:P=R.(120)+3:GOS.845:GOS.850:IFD>1A$=SHIPS
205 IFP.(110,1)=0L=R.(9)*(-1):GOS.850:GOS.840
210 IF(P.(110,1))*(P.(112,1)=0)*(P.(114,1))L=R.(9)
211 IFL>0GOS.850:GOS.840
215 IFP.(114,1)=0GOS.850:GOS.250
220 IF(P.(112,1))*(R.(7)=7)GOS.304:GOS.850
222 IFD=0P.A.65,"                        ";:G.235
225 IFD<TP.A.65;"YOU HAVE";D;A$;:T=D:GOS.850
229 IFE>FP.A.88;"SCORE ↑";E;A.106;I;B$;" SHOT !":F=E:GOS.850:IFH<EH=E
231 F.G=1T.21:IFA(G)>0G.233
232 N.G:C.:P.A.88;"REINFORCEMENTS":F=F-1:T=T+1:G.99
233 IFN=0G.188
234 G.205
235 P.A.256:P."THE ALIENS HAVE SHOT DOWN YOUR FLEET !!
236 P.:P.:I."WOULD YOU LIKE ANOTHER TRY ";A:IFA=YG.9
237 C.:E.
250 F.A=40T.7S.-1:IFP.(P,A-1)G.280
260 S.(P,A):S.(P,A-1):R.(P,A):R.(P,A-1):N.A:G.300
280 R.(P,A-1):IFA>26G.300
283 P.A.6;"WELL";A.55;"SHOT":IFA<11Z=0:G.287
284 F.Z=1T.21:GOS.999:IF(P>X-1)*(P<X+7)*(A>Y-1)*(A<Y+3)G.287
285 N.Z
287 GOS.999:I=I+1:F.C=1T.3:GOS.770:IFZ>0GOS.820
289 IFZ<1GOS.800
295 N.C:IFZ>0A(Z)=0
296 IFZ<1GOS.830
300 B$="ALIEN":IFI>1B$=ALIENS
303 RET.
304 P.A.55;"WAIT";:GOS.790:GOS.795
305 IF(K=3)*(Q=Z+7)Z=0:G.320
306 IFZ<1Z=R.(7)+14
307 IF(Z=14)*(A(21)>0)Z=21
320 GOS.999:IFZ=0Y=Y+6
321 F.B=Y+2T.44:IFP.(X+3,B)G.330
327 IFX=0RET.
328 S.(X+3,B):R.(X+3,B):N.B:G.345
330 IFB>39D=D-1:P.A.6;"HIT!";:F.J=1T.5:GOS.840:N.J
331 IFD<TN=0:GOS.840
332 F.M=X+R.(3)T.X+2+R.(3):R.(M,B):N.M
333 IFD=1A$=SHIP
345 Z=R.(21):G=3-R.(5):GOS.999
351 C=2:IFZ>14C=1
353 IFZ<8C=3
355 IF(X+G>A(Z+C*7)+5)+(X+G<A(Z+C*7)-5)G.345
356 IFY>0GOS.820:X=X+G:GOS.998
360 IFY>0ONCGOS.835,825,815
700 GOS.830:RET.
770 IFZ>14E=E+10:GOS.835
771 IF(Z>7)*(Z<15)E=E+20:GOS.825
772 IFZ<1E=E+R.(99):GOS.800:GOS.805
773 IF(Z>0)*(Z<8)E=E+30:GOS.815
779 RET.
790 F.B=7T.1S.-1:IF(P>A(B+21)-8)*(P<A(B+21)+8)RET.
791 N.B:RET.
795 K=0:F.Z=14+BT.1S.-7:IFA(Z)=0K=K+1:N.Z
796 RET.
800 R.(X+4,Y+3):F.V=X+2T.X+6:R.(V,Y+2):N.V
801 R.(X+1,Y+1):R.(X+2,Y+1):R.(X+6,Y+1):R.(X+7,Y+1)
802 R.(X,Y):R.(X+1,Y):R.(X+7,Y):R.(X+8,Y):RET.
805 S.(X,Y):S.(X+1,Y):S.(X+7,Y):S.(X+8,Y)
806 S.(X+1,Y+1):S.(X+2,Y+1):S.(X+6,Y+1):S.(X+7,Y+1)
807 F.V=X+2T.X+6:S.(V,Y+2):N.V:S.(X+4,Y+3):RET.
```

```
815 IFX<0X=0
816 S.(X+2,Y-1):S.(X+4,Y-1):S.(X+3,Y):S.(X+3,Y+1):RET.
820 F.W=Y+1T.Y-1S.-1:F.V=XT.X+6:R.(V,W):N.V:N.W:RET.
825 S.(X+1,Y-1):S.(X+5,Y-1)
826 F.V=X+2T.X+4:S.(V,Y):N.V:S.(X+3,Y+1):RET.
830 Z=0:GOS.999:GOS.800:Q=R.(7):X=(Q-1)*17+6:GOS.998:GOS.805:RET.
835 F.V=X+1T.X+5:S.(V,Y):N.V:F.V=XT.X+6S.3:S.(V,Y-1):N.V
837 S.(X+3,Y+1):RET.
840 R.(P,41):R.(P+1,41):F.X=P-1T.P+2:R.(X,42):N.X
842 F.X=P-2T.P+3:R.(X,43):N.X
843 F.X=P-3T.P+4:R.(X,44):N.X:P=P+L:IFP<3P=3
844 IFP>123P=123
845 IFN=0RET.
846 S.(P,41):S.(P+1,41):F.X=P-1T.P+2:S.(X,42):N.X
847 F.X=P-2T.P+3:S.(X,43):N.X
848 F.X=P-3T.P+4:S.(X,44):N.X:L=0:RET.
850 P.A.56;:F.V=12T.19S.2:S.(V,1):S.(V+98,1):N.V:RET.
998 IFZ>=0A(Z)=X*100+Y:RET.
999 IFZ>=0X=I.(A(Z)/100):Y=A(Z)-X*100
1000 RET.
```

** BIG LETTERS - L1/4K **

This program asks you to input letters and punctuation and, when you have finished, prints them out in a 4cm. by 4 cm. format, 8 letters per line. You also have the option of having two rows of 16 small letters at the bottom of the screen. When the program has completed printing your message, it pauses for a moment and then commences to redraw the same message again. This continues until you change the message.

The program was originally developed by a teacher to assist children learn to recognise the letters of the alphabet and to spell. The repeat feature was included so the computer could continue to reinforce its lesson whilst the teacher moved on to other things. It would be a useful routine to build into a longer program such as a spelling test for young children.

BIG LETTERS - L1/4K BY BR. P. VAN EEKEN

```
2 C.:A(0)=1:P."TYPE IN YOUR BIG LETTERS, OR:
3 P."0=SPACE  27=HYPHEN  28=LINE END  29='  30=?  99=FINISH
4 A=1:B=2:C=3:D=4:E=5:F=6:G=7:H=8:I=9:J=10:K=11:L=12:M=13:N=14
5 O=15:P=16:Q=17:R=18:S=19:T=20:U=21:V=22:W=23:X=24:Y=25:Z=26
7 P.A.448:P.A.448;"LETTER";A(0);:I.A(A(0)):IFA(A(0))=99G.14
8 IFA(A(0))>30P." NO!":G.7
9 P.:A(A(0))=A(A(0))+50:A(0)=A(0)+1:G.7
14 I."CARE FOR 2 SETS OF 16 SMALL LETTERS AT SCREEN BASE ";A
15 A$="":B$="":IFA=YI."SET 1";A$:I."SET 2";B$
30 C.:P.A.916;:X=0:Y=J:R=0:GOS.50:P.A$;" ";B$
33 F.B=1T.5:F.D=1T.999:N.D:N.B:R=1:X=0:Y=J:GOS.50:G.30
38 F.H=Y-9T.Y-8
39 IFR=0F.B=X+6T.X+7:S.(B-6,H):S.(B,H+4):S.(B,Y):N.B
40 IFR=1F.B=X+6T.X+7:R.(B-6,H):R.(B,H+4):R.(B,Y):N.B
41 N.H:G.90
50 F.L=1T.A(0)-1:S=A(L)
51 IFS=50A=10:G.91
52 IFS=78X=-A:Y=Y+O:G.91
53 REA.A:IFS<>AG.53
54 REA.A:IFX+A>129X=0:Y=Y+O:IFY>40L=A(0):G.91
55 IF(S=59)+(S=60)+(S=70)X=X-2
56 REA.F:F.C=1T.F:REA.B
57 IF(B=1)+(B=25)+(B=23)+(B=29)GOS.200
58 IF(B=2)+(B=16)+(B=22)+(B=26)GOS.210
59 IFB=13G.270
60 IFB=4GOS.230
61 IF(B>4)*(B<8)GOS.240
62 IF(B>7)*(B<11)GOS.250
63 IF(B=11)+(B=12)+(B=3)GOS.220
64 IFB=14GOS.260
65 IF(B=15)+(B=27)GOS.280
66 IF(B>16)*(B<20)GOS.290
```

```
67 IFB=20GOS.300
68 IFB=21GOS.240
69 IFB=24GOS.216
70 IFB=30GOS.310
72 IFB=31G.38
73 IFB<32G.90
74 K=X+7:IFS=77K=X+2
76 F.H=KT.K+5:IFR=0S.(H,Y-5)
77 IFR=1R.(H,Y-5)
78 N.H
90 N.C
91 X=X+A
92 IFA(L+1)<=SREST.
93 N.L:REST.:RET.
100 D.51,22,3,1,2,4,52,16,6,3,5,6,7,8,9,53,15,3,3,5,7
103 D.54,16,4,3,5,7,10,55,15,4,3,5,6,7,56,15,3,3,5,6
106 D.57,16,5,3,5,7,8,32,58,16,3,3,11,6,59,15,3,5,7,12
109 D.60,16,3,13,11,7,61,14,2,3,14,62,15,2,3,7
112 D.63,16,3,3,11,15,64,16,3,3,16,11,65,16,4,3,5,11,7
115 D.66,16,4,3,5,9,6,67,19,5,3,5,11,7,17,68,16,5,3,5,9,6,18
118 D.69,16,5,5,20,6,8,21,70,15,2,5,12,71,16,3,3,7,11
121 D.72,21,2,22,23,73,31,4,22,24,19,25,74,12,2,1,26
124 D.75,14,2,29,27,76,16,3,5,29,7,77,12,1,32
125 D.79,8,1,30,80,16,4,5,9,31,32
200 E=Y:K=X:IFB=23K=X+9
201 IFB=25K=X+19
203 IFB=29K=X+2
204 F.D=KT.K+9:IFR=0S.(B,E):S.(D,E-1)
205 IFR=1R.(D,E):R.(D,E-1)
206 E=E-1:N.D:RET.
210 H=X+2:E=Y-10:IFB=2H=X+10
211 IFB=22H=X
212 IFB=26H=X
213 F.D=HT.H+9:IFR=0S.(D,E):S.(D,E+1)
214 IFR=1R.(D,E):R.(D,E+1)
215 E=E+1:N.D:RET.
216 E=Y:D=X+10:F.H=DT.D+3:E=E-1:IFR=0S.(H,E)
217 IFR=1R.(H,E)
218 N.H:RET.
220 K=0:IFB=11K=12
221 IFB=12K=7
223 F.D=Y-10T.Y:IFR=0S.(X+K,D):S.(X+K+1,D)
224 IFR=1R.(X+K,D):R.(X+K+1,D)
225 N.D:RET.
230 E=Y-5:F.D=X+6T.X+14:IFR=0S.(D,E)
231 IFR=1R.(D,E)
232 N.D:RET.
240 IFB=5E=Y-10
241 IFB=6E=Y-5
242 IF(B=7)+(B=21)E=Y
243 H=2:IFB=21H=0
245 F.D=X+HT.X+12:IFR=0S.(D,E)
246 IFR=1R.(D,E)
247 N.D:RET.
250 IFB=8E=Y-1
251 IFB=9E=Y-6
252 F=3:IFB=10E=Y-1:F=8
255 F.D=X+12T.X+13:F.H=E-FT.E:IFR=0S.(D,H)
256 IFR=1R.(D,H)
257 N.H:N.D:RET.
260 E=Y-5:F.D=X+2T.X+6:IFR=0S.(D,E)
261 IFR=1R.(D,E)
262 N.D:D=D-1
263 E=E-1:F.G=DT.D+1:IFR=0S.(G,E)
264 IFR=1R.(G,E)
265 N.G:IFE<>Y-10D=D+1:G.263
266 E=Y-4:D=X+6:F.M=1T.5:F.G=DT.D+1:IFR=0S.(G,E)
268 IFR=1R.(G,E)
269 N.G:E=E+1:D=D+1:N.M:RET.
270 F.D=X+8T.X+9:IFR=0S.(D,Y-10)
271 IFR=1R.(D,Y-10)
272 N.D:F.D=X+2T.X+3:F.H=Y-2T.Y:IFR=0S.(D,H)
```

(We haven't left anything out here either! There is no Line 273 - Ed.)

```
274 IFR=1R.(D,H)
275 N.H:N.D:G.90
280 E=Y-10:F.D=XT.X+5:IFR=0S.(D,E):S.(D+1,E)
281 IFR=1R.(D,E):R.(D+1,E)
282 E=E+1:N.D:E=Y-10:IFB=27RET.
283 F.D=X+12T.X+6S.-1
284 IFR=0S.(D,E):S.(D+1,E)
285 IFR=1R.(D,E):R.(D+1,E)
286 E=E+1:N.D:RET.
290 E=Y-5:H=X+10:IFB=18H=X+6
291 IFB=19H=X+14
292 F.D=HT.H+5:IFR=0S.(D,E):IFB<19S.(D+1,E)
293 IFR=1R.(D,E):IFB<19R.(D+1,E)
294 E=E+1:N.D:RET.
300 F.D=Y-10T.Y-5:IFR=0S.(X,D):S.(X+1,D)
301 IFR=1R.(X,D):R.(X+1,D)
302 N.D:RET.
310 F.H=X+2T.X+5:IFR=0S.(H,Y-10):S.(H,Y-9):IFH>X+3S.(H,Y-8)
312 IFR=1R.(H,Y-10):R.(H,Y-9):IFH>X+3R.(H,Y-8)
315 N.H:RET.
```

***** TANK BATTLE (L2/4K) *****

This one will kill you! Two tanks move around in a maze trying to blow each other to bits. The first player to destroy ten tanks is the winner. To control the tanks the left hand player presses "1" to turn and "2" to move. To fire he presses both keys at the same time. The right hand player presses "9" to turn and "8" to move and presses both together to fire.
After a tank has been hit it is automaticly moved to a different position in the maze to prevent the other player scoring a second hit and to give the injured party time to recover from being wiped off the face of the earth.

```
0 '         TANK BATTLE LII/4K - BY D. J. MOORHOUSE
1 '         315 EUREKA STREET , BALLAARAT , VIC. 3350


2 CLS:FORX=2TO125:SET(X,2):SET(X,45):NEXT:FORY=2TO45:SET(2,Y):SET(125,Y):NEXT
5 FORX=15370TO15419STEP10:FORY=X+64TOX+896STEP64:T=RND(3):IFT=1THEN7
6 POKEY,191
7 NEXTY,X
8 GOTO300
10 IFPEEK(PS+D)<>32THENRETURN
20 POKEPS,32:POKEPS+1,32:POKEPS+2,32:PS=PS+E
30 ONRTGOTO50,60,70
40 POKEPS,179:POKEPS+1,191:POKEPS+2,140:RETURN
50 POKEPS,188:POKEPS+1,143:POKEPS+2,188:RETURN
60 POKEPS,140:POKEPS+1,191:POKEPS+2,179:RETURN
70 POKEPS,143:POKEPS+1,188:POKEPS+2,143:RETURN
80 ONRTGOTO100,110,120
90 D=3:E=1:RETURN
100 D=-64:E=-64:RETURN
110 D=-1:E=-1:RETURN
120 D=64:E=64:RETURN
130 F=PS+D
140 IFF=P9ORF=P9+1ORF=P9+2THEN180
150 IFPEEK(F)<>32THENPOKEF-E,32:RETURN
160 POKEF,42:POKEF-E,32
170 F=F+E:GOTO140
180 S=-1:POKEF-E,32:FORW=0TO19:POKEP9,174:POKEP9+1,140:POKEP9+2,157:POKEP9,32:PO
KEP9+1,32:POKE P9+2,32:NEXTW
182 P9=RND(1024)+15360:IFPEEK(P9)<>32ORPEEK(P9+1)<>32ORPEEK(P9)<>32THEN182
183 RETURN
300 P1=15746:P2=15802:R1=4:R2=2
310 PS=P1:RT=R1:GOSUB30:PS=P2:RT=R2:GOSUB30
320 IFPEEK(16442)=6THENPS=P1:S=0:RT=R1:P9=P2:GOSUB80:GOSUB130:P2=P9:PS=P2:RT=R2:
GOSUB30:IFSTHEN1000ELSE370
330 IFPEEK(16442)=4THENPS=P1:RT=R1:GOSUB80:GOSUB10:P1=PS:GOTO370
350 IFPEEK(16442)<>2THEN370
360 R1=R1+1:IFR1=5THENR1=1
365 PS=P1:RT=R1:GOSUB30:GOTO370
370 IFPEEK(16443)=3THENPS=P2:S=0:RT=R2:P9=P1:GOSUB80:GOSUB130:P1=P9:PS=P1:RT=R1:
GOSUB30:IFSTHEN1050ELSE320
```

```
380 IFPEEK(16443)=1THENPS=P2:RT=R2:GOSUB80:GOSUB10:P2=PS:GOTO320
390 IFPEEK(16443)<>2THEN320
400 R2=R2+1:IFR2=5THENR2=1
410 PS=P2:RT=R2:GOSUB30:GOTO320
1000 S1=S1+1:IFS1=10THENA$="LEFT":GOTO1500
1010 GOTO370
1050 S2=S2+1:IFS2=10THENA$="RIGHT":GOTO1500
1060 GOTO320
1500 CLS:PRINT@216,"G A M E  O V E R "
1510 PRINT:PRINTA$;" PLAYER WON THAT GAME"
1520 PRINT:PRINT"SCORES   LEFT PLAYER ";S1;" HITS"
1530 PRINT"            RIGHT PLAYER ";S2;" HITS"
```

***** RESISTOR DECODER (L2/4K) *****

Here is a program that will be a great help to all those amateur electronic buffs who cannot remember what the coloured bands on resistors stand for without looking them up. With this program, you just type in the colours of the first three bands from left to right, separated by commas and then press ENTER/NEW LINE.

The program then asks for the colour of the fourth band (which indicates the tolerance of the resister). If there is no fourth colour, enter NO COLOUR. The program then returns with the value of the resistor and it's tolerance.

```
          RESISTOR COLOUR CODE DE-CODER LII/4k - BY J. ANDERSON
          2250 OCEAN BEACH ROAD, SORRENTO, 3943


10 CLS
20 PRINT@19,"RESISTOR COLOUR CODE TRANSLATOR"
30 INPUT"ENTER THE  THREE COLOUR'S FROM LEFT TO RIGHT SEPARATING THEM BY A COMMA
 ( , ) ";O$,S$,F$
31 INPUT"NOW THE FOURTH COLOUR IF NONE THEN ENTER 'NO COLOUR' ";T$
50 IFO$="BLACK"THEN O=00
60 IFO$="BROWN"THEN O=10
61 IFO$="RED"THENO=20
62 IFO$="ORANGE"THENO=30
63 IFO$="GREEN"THENO-50
64 IFO$="BLUE"THENO=60
70 IFO$="YELLOW"THENO=40
80 IFO$="VIOLET"THENO=70
81 IFO$="GREY"THENO=80
82 IFO$="WHITE"THENO=90
90 IFS$="BLACK"THENS=0
100 IFS$="BROWN"THENS=1
110 IFS$="RED"THENS=2
120 IFS$="ORANGE"THENS=3
130 IFS$="YELLOW"THENS=4
140 IFS$="GREEN"THENS=5
150 IFS$="BLUE"THENS=6
160 IFS$="VIOLET"THENS=7
170 IFS$="GREY"THENS=8
180 IFS$="WHITE"THENS=9
300 IFF$="BROWN"THENF=10
310 IFF$="RED"THENF=100
320 IFF$="ORANGE"THENF=1000
330 IFF$="YELLOW"THENF=10000
340 IFF$="GREEN"THENF=100000
350 IFF$="BLUE"THENF=1000000
360 IFF$="VIOLET"THENF=10000000
370 IFF$="GREY"THENF=100000000
380 IFF$="WHITE"THENF=1000000000
390 IFT$="GOLD"THENP$="5%"
400 IFT$="SILVER"THENP$="10%"
410 IFT$="NO COLOUR"THENP$="20%"
420 R=(O+S)*F
430 PRINT"RESISTANCE=";R;"OHMS TOLERANCE=";P$
440 RI=((O+S)*F)/1000:PRINT"RESISTANCE=";RI;"KILOHMS      TOLERENCE=";P$
999 INPUT"DO YOU HAVE ANY MORE PROBLEMS";Z$:IFZ$="YES"THENRUN
```

***** CONCENTRATION L2/16K *****

This game is for 1 to 5 players. If you wish, the computer will be one of the players just by typing in COMPUTER when the program asks for the players names. The program draws three rows of cards, you select a card from one row and the program turns the card over, then you select a second card. If they match you score 5 points, if they do not match the cards are turned back over and thats where the concentration comes in, trying to remember what card was where. Lots of instructions are included in the game, so not a lot more need be said here except CONCENTRATE.

```
                    CONCENTRATION LII/16K - BY C.PARKINSON
                    47 SILLMON WAY
                    DUNCRAIG W.A. 6023

10 CLS:PRINTCHR$(23):PRINT@390,"C O N C E N T R A T I O N";
15 FORT=1TO40
20 X=RND(1023):IFPEEK(X+15360)<>32THEN20ELSEPOKE(X+15360),63
25 FORTT=1TO50:NEXT:NEXT
30 PRINT:PRINT:PRINT:PRINT:PRINTTAB(5)"DO YOU WANT THE RULES ?";
40 R$="":R$=INKEY$:IFR$=""THEN40
50 IFR$="Y"THENGOSUB600:CLS:GOTO70
60 IFR$<>"N"THEN40ELSECLS
70 PRINT@398,"HOW MANY PLAYERS (1 - 5)"
72 P$=INKEY$:IFP$=""THEN72
73 P=VAL(P$)
75 IFP>5ORP<1THENPRINTTAB(15)"I SAID 1 - 5 DUMMY";:FORT=1TO500:NEXT:GOTO70
76 PRINT:PRINT"PLEASE ENTER YOUR NAMES":PRINT
80 FORI=1TOP:PRINT"PLAYER";I;:INPUTP$(I):NEXT
100 DIMC(3,9),D(27),A(3,9),B(3,9),ME(3,9)
104 CLS:PRINT@398,CHR$(23);"SHUFFLING"
105 GOTO220
110 CLS:FORX=4TO100:SET(X,3):SET(X,11):SET(X,15):SET(X,23):SET(X,27):SET(X,35):N
EXT
120 FORX=13TO100STEP11:GOSUB150:NEXT
130 FORX=14TO100STEP11:GOSUB150:NEXT
140 GOTO160
150 RESET(X,3):RESET(X,11):RESET(X,15):RESET(X,23):RESET(X,27):RESET(X,35):RETUR
N
160 FORY=3TO11:GOSUB200:NEXT
170 FORY=15TO23:GOSUB200:NEXT
180 FORY=27TO35:GOSUB200:NEXT
190 GOTO210
200 SET(4,Y):SET(12,Y):SET(15,Y):SET(23,Y):SET(26,Y):SET(34,Y):SET(37,Y):SET(45,
Y):SET(48,Y):SET(56,Y):SET(59,Y):SET(67,Y):SET(70,Y):SET(78,Y):SET(81,Y):SET(89,
Y):SET(92,Y):SET(100,Y):RETURN
210 PRINT@3,"1      2     3      4      5     6     7      8       9";:PRINT@180,"ROW 1";
:PRINT@436,"ROW 2";:PRINT@692,"ROW 3";
211 PRINT@259,"1      2     3      4      5     6     7      8       9";:PRINT@515,"1
 2     3      4      5     6     7      8       9";
215 GOTO320
220 FORI=1TO3:FORJ=1TO9:C(I,J)=0:ME(I,J)=12:NEXT:NEXT:FORI=1TOP:SC(I)=0:NEXT
230 DATA1,2,3,4,5,6,7,8,9,132,137,143,148,154,159,165,170,176,388,393,399,404,41
0,415,421,426,432,644,649,655,660,666,671,677,682,688
240 RESTORE:FORI=1TO9:READD(I):NEXT
245 FORI=1TO3:FORJ=1TO9:READB(I,J):NEXTJ,I
250 RESTORE:FORI=10TO18:READD(I):NEXT
260 RESTORE:FORI=19TO22:READD(I):NEXT
265 RESTORE:FORI=23TO26:READD(I):NEXT
266 D(27)=10
280 FORI=1TO3:FORJ=1TO9
290 R=RND(27):IFD(R)=0THEN290
295 IFD(R)=10THEND(R)=0
300 C(I,J)=D(R):D(R)=0
310 NEXTJ,I
315 GOTO110
320 FORI=1TO3:FORJ=1TO9:PRINT@B(I,J),"#";:NEXTJ,I
330 FORPL=1TOP
335 IFTC=26THEN500
336 IFP$(PL)="COMPUTER"GOSUB1000:GOTO342
340 PRINT@832,CHR$(30):PRINT@832,P$(PL);" FIRST CARD. ROW";:INPUTL:PRINT@866," C
OLUMN";:INPUTM:IFL<1ORL>3ORM<1ORM>9THENPRINT@832,CHR$(30);:GOTO340
342 IFC(L,M)=10THEN336
```

```
343 PRINT@B(L,M)-1,C(L,M);:ME(L,M)=C(L,M)
346 IFC(L,M)=0THENPRINT@832,CHR$(30):PRINT@840,"SORRY ";P$(PL);" YOU MISS A TURN
";:FORT=1TO2000:NEXT:PRINT@B(L,M),"◆";::NEXT:GOTO330
347 IFP$(PL)="COMPUTER"GOSUB1100:GOTO356
350 PRINT@900," SECOND CARD ROW";:INPUTN:PRINT@930,"COLUMN";:INPUTO:IFN<1ORN>3OR
O<1ORO>9THENPRINT@896,CHR$(30);:GOTO350
355 IFL=NANDM=OTHENPRINT@896,CHR$(30):GOTO350
356 IFC(N,O)=10THEN347
360 PRINT@B(N,O)-1,C(N,O);:ME(N,O)=C(N,O)
370 IFC(N,O)=C(L,M)THENSC(PL)=SC(PL)+5:TC=TC+2:GOTO400
380 PRINT@832,CHR$(30):PRINT@840,"SORRY ";P$(PL);" THEY DON'T MATCH":FORT=1TO200
0:NEXT:PRINT@B(L,M),"◆";:PRINT@B(N,O),"◆";:NEXT:GOTO330
400 PRINT@832,CHR$(30):PRINT@840,"WELL DONE ";P$(PL);" YOU SCORE 5 POINTS";:FORT
=1TO2000:NEXT:PRINT@B(L,M)," ";:PRINT@B(N,O)," ";:C(L,M)=10:C(N,O)=10:ME(L,M)=10
:ME(N,O)=10:GOTO335
500 CLS:PRINT@200,"THEN GAME IS OVER.  THE SCORES ARE AS FOLLOWS"
510 PRINT:FORI=1TOP
520 PRINTTAB(15)P$(I);" SCORED ";SC(I)
530 NEXT
540 PRINT:PRINT"DO YOU WISH TO PLAY AGAIN ?"
550 A$="":A$=INKEY$:IFA$=""THEN550
560 IFA$="Y"THEN590ELSEIFA$<>"N"THEN540
570 CLS:PRINT@400,CHR$(23);"BYE BYE":FORT=1TO2000:NEXT
580 CLS:END
590 CLS:PRINT@400,CHR$(23);"SHUFFLING":GOSUB700:FORPL=1TOP:SC(PL)=0:NEXT:TC=0:GO
TO105
600 CLS:PRINTTAB(20)"** RULES **"
610 PRINT:PRINT:PRINTTAB(5)"THIS GAME IS SIMILAR TO THE GAME OF CONCENTRATION":P
RINTTAB(5)"PLAYED WITH CARDS""
620 PRINTTAB(5)"THERE MAY BE UP TO 5 PLAYERS. EACH PLAYER TAKES A":PRINTTAB(5)"T
URN AT CHOOSING TWO CARDS":PRINTTAB(5)"IF THE TWO CARDS ARE THE SAME THE PLAYER
SCORES 5":PRINTTAB(5)"POINTS AND HAS ANOTHER TURN"
630 PRINTTAB(5)"IF THE CARDS ARE NOT THE SAME. THAT PLAYER ENDS":PRINTTAB(5)"HIS
 GO AND IT IS THE NEXT PLAYERS GO.":PRINTTAB(5)"IF A PLAYER TURNS UP THE '0' ON
HIS FIRST GO HE":PRINTTAB(5)"LOSES HIS TURN"
640 PRINT:PRINTTAB(15)"PRESS ANY KEY TO CONTINUE"
650 A$="":A$=INKEY$:IFA$=""THEN650
660 CLS:FORX=25TO32:SET(X,3):SET(X,11):SET(X,15):SET(X,23):SET(X,27):SET(X,35):N
EXT:FORY=3TO11:SET(25,Y):SET(33,Y):NEXT:FORY=15TO23:SET(25,Y):SET(33,Y):NEXT:FOR
Y=27TO35:SET(25,Y):SET(33,Y):NEXT
670 PRINT@142,"◆";:PRINT@147,"THIS IS HOW THE CARDS LOOK WHEN DEALT";
675 FORT=1TO1000:NEXT
680 PRINT@398,"5";:PRINT@403,"THIS IS A TURNED UP CARD";:FORT=1TO1000:NEXT:PRINT
@659,"THIS IS A CARD THAT HAS BEEN USED";
690 FORT=1TO2000:NEXT:PRINT@832,"IF YOU WANT THE COMPUTER TO PLAY ONE OF THE HAN
DS ENTER COMPUTERAS ONE OF THE PLAYERS":FORT=1TO4000:NEXT:RETURN
700 IFP=1THENRETURN
701 FORI=1TOP:PP$(I)="":NEXT
705 FORI=1TOP:IFPP$(I)=""THEN710ELSENEXT
706 GOTO740
710 X=RND(P)
715 IFPP$(X)<>""THEN705
720 FORI=1TOP:IFP$(I)<>""THENPP$(X)=P$(I):P$(I)="":GOTO705
730 NEXT:GOTO705
740 FORX=1TOP:P$(X)=PP$(X):PP$(X)="":NEXT
750 TROFF:RETURN
1000 RANDOM:TM=0:PRINT@832,CHR$(30):PRINT@842,"COMPUTERS TURN":FORT=1TO1000:NEXT

1005 L=RND(3):M=RND(9)
1010 IFME(L,M)=0ORME(L,M)=10THEN1005
1011 TM=TM+1:IFTM<100ANDTC<20ANDME(L,M)<>12THEN1005
1015 PRINT@832,CHR$(30)
1020 PRINT@842,"I CHOOSE";L;",";M;" AND";:PRINT@B(L,M)-1,C(L,M);:FORT=1TO2000:NE
XT:RETURN
1100 FORI=1TO3:FORJ=1TO9:IFME(I,J)=0ORME(I,J)=10THENNEXT:NEXT:GOTO1120
1110 IFME(I,J)=C(L,M)THENN=I:O=J:GOTO1115
1113 NEXT:NEXT:GOTO1120
1115 IFN=LANDO=MTHENNEXT:NEXT
1116 IFL<>NORM<>OTHENGOTO1140
1120 N=RND(3):O=RND(9)
1125 IFN=LANDM=O THEN1120
1130 IFME(N,O)=0ORME(N,O)=10THEN1120
1140 PRINT@B(N,O)-1,C(N,O);:PRINT@861,N;",";O;:FORT=1TO2000:NEXT:RETURN
```

** SYSCOPY L2/16K Machine Language.**

Have you ever wanted to make a back-up copy of a machine langage tape? First you need a monitor program such as BMON (described in MICRO-80 Issues 3, 4 & 5 or available on cassette from MICRO-80 PRODUCTS) which enables you to make SYSTEM tapes. However, that is not much help if you do not know where the machine language program to be copied starts or finishes, or its entry point. SYSCOPY is the answer. This program loads in a machine language program from tape and then displays its START, END and ENTRY addresses in Hex. Even better, it has been designed to reside just below BMON so you can use the two programs in conjunction with each other.

SYSTEN 80/Video Genie users who have been having loading problems with their on-board cassette deck get an added bonus from SYSCOPY because it allows you to specify whether the machine language program is to be loaded in via cassette port #1 or #2. Until now, the only way you could load m/l programs via the external cassette port was by installing a changeover switch in your machine to reverse the roles of the two sockets. All you need to do now is use SYSCOPY and specify the cassette port to be used (#-1, or #-2), in accordance with the instructions included in the listing (This feature also works on TRS-80's).

A commented source code listing has been provided to help you understand the operation of the program. It will also enable those readers with an editor/assembler to relocate the program to reside anywhere in memory. As listed, the origin of the program is 70BDH (28861 Dec), which is below BMON in 16K machines. The program entry point is 7110H (28944 Dec). If you wish to relocate the program, change the "ORG" address in line 790 to the new address and reassemble the program.

Those who do not have an editor/assembler or are happy with the location of the program, may type it in directly using a monitor such as BMON, starting at program line 00810. The column to the extreme left of the page contains the address in memory (in Hex) of each Z80 instruction. The next column contains the instructions and their operands etc. in Hex, two Hex digits per byte. Commence by putting your monitor in the Edit Memory mode at address 70BD Hex. Change the data at that address to 2B then change the data at address 70BE to 22, 70BF to 54, 70C0 to 41, 70C1 to CD and so on until you reach 7200 which you can leave unchanged. Exit the Edit mode.

To check the accuracy of your input, Jump to address 7110 when the program should operate correctly. Before doing this, however, it is good practice to save the program onto tape. That way, if one or two bytes have been entered incorrectly and cause the program to crash or hang-up, you need only reload from tape rather than having to type the whole lot in again!

You can use your monitor to punch out a m/l tape of SYSCOPY with the following parameters:-

START	70BD
END	71FF
ENTRY	7110
NAME	SYSCPY (note that the O has been dropped since only 6 letters are allowed in the name)

To reload your tape (or the cassette edition tape):-

1. Switch on and answer the MEMORY SIZE? or READY? with 28860. Press the ENTER/NEWLINE key.

2. Type in SYSTEM. Press the ENTER/NEWLINE key.

3. Ready the cassette by setting the tape to the beginning of the SYSCOPY program, adjusting the volume control if necessary and pressing the PLAY key.

4. Answer the *? with SYSCPY. Press ENTER/NEWLINE key.

5. Two asterisks should appear in the upper right-hand corner of the screen. The right-hand asterisk should turn on for about 4 seconds, off for a further 4 seconds on again and so on. Those with the cassette edition who listen to their tape will hear the signal in bursts. This tape has been made directly from the Editor/Assembler and the data is dumped in blocks of 256 bytes with short, quiet periods between.

6. When the program has finished loading the screen will display *?-. Answer this with / press the ENTER/NEWLINE key.

7. Again, the screen will display *?- This indicates that the program is functioning and it is requesting the name of a machine language program that you want to load in. Put a tape containing a machine language program in the cassette deck. Ready if for loading. Type in the name of the program. Press ENTER/NEWLINE.

8. Again, you will get the two asterisks in the upper right corner of the screen with the right hand one blinking at four-second intervals.

9. When the program has finished loading, the screen will display:-

```
START = XXXX END = XXXX ENTRY = XXXX
```

On the next line you will again get *?- . This is inviting you to load in another m/l program. If you do not want to do that you can escape back to BASIC by pressing the BREAK key.

10. At any time, you can re-enter SYSCOPY by typing LOAD then pressing the ENTER/NEWLINE key.

```
                 00100 ;         SYSTEM SIMULATOR
                 00110 ;
                 00120 ;         ANDREW SMALLRIDGE
                 00130 ;         15/9/80
                 00140 ;
                 00150 ;THE FUNCTION OF THIS PROGRAM IS TO SIMULATE
                 00160 ;THE SYSTEM COMMAND. IN ITS INITIAL FORM THIS
                 00170 ;ROUTINE WILL LOAD A SYSTEM TAPE AS  PER THE
                 00180 ;SYSTEM COMMAND, THE DIFFERENCE BEING AT THE
                 00190 ;END OF THE LOAD THIS ROUTINE WILL DISPLAY
                 00200 ;THE START, END AND ENTRY POINT OF THE PROGRAM
                 00210 ;THIS WILL ENABLE THE COPYING OF A MACHINE
                 00220 ;LANGUAGE PROGRAM WITH THE BMON PROGRAM
                 00230 ;
                 00240 ;THIS PROGRAM WILL ALSO ENABLE THE USER TO
                 00250 ;SELECT THE CASSETTE DRIVE REQUIRED TO LOAD
                 00260 ;A SYSTEM TAPE
                 00270 ;E.G. TO LOAD 'BMON'. ON ENTRY SYSCPY WILL
                 00280 ;RESPOND WITH *?  THE USER MAY ENTER THE
                 00290 ;FOLLOWING OPTIONS:
                 00300 ;
                 00310 ;         *? BMON
                 00320 ;THE SYSCPY COMMAND WILL NOW LOAD THE PROGRAM
                 00330 ;'BMON' FROM CASSETTE DRIVE 1
                 00340 ;
                 00350 ;         *? #-1,BMON
                 00360 ;THIS WILL ALSO RESULT IN 'BMON' BEING LOADED
                 00370 ;FROM CASSETTE DRIVE 1
                 00380 ;
                 00390 ;         *? #-2,BMON
                 00400 ;THIS WILL RESULT IN 'BMON' BEING LOADED FROM
                 00410 ;CASSETTE DRIVE 2
                 00420 ;
                 00430 ;
                 00440 ;ONCE INITIALIZED THIS PROGRAM CAN BE
                 00450 ;REENTERED USING THE LOAD COMMAND
                 00460 ;E.G. ENTERING LOAD WILL RESPOND WITH
                 00470 ;*?
                 00480 ;
                 00490 ;AT THE END OF THE LOAD FROM CASSETTE THE
                 00500 ;SCREEN WILL DISPLAY:
                 00510 ;
                 00520 ;START=XXXX END=XXXX ENTRY=XXXX
                 00530 ;
                 00540 ;THIS PROGRAM WILL CORESIDE WITH 'BMON'
                 00550 ;
                 00560 ;         DEFINE GLOBALS
                 00570 ;
1997             00580 SNERR    EQU     1997H   ;SN ERROR ENTRY
031D             00590 ENTRY    EQU     031DH   ;JUMP TO ENTRY ON "/"
0212             00600 CASON    EQU     0212H   ;TURN ON CASSETTE
0235             00610 READT    EQU     0235H   ;READ BYTE FROM TAPE
022C             00620 BLINK    EQU     022CH   ;BLINK ASTERIX
4152             00630 START    EQU     4152H   ;TEMP STORE FOR START ADDRESS
4154             00640 ENDAD    EQU     START+2 ;TEMP STORE FOR END ADDRESS
0314             00650 READ2    EQU     0314H   ;READ 2 BYTES FROM TAPE
4156             00660 PASS1    EQU     START+4 ;FLAG INDICATES FIRST BLOCK READ
3C3E             00670 ASRIX    EQU     3C3EH
40DF             00680 ENTAD    EQU     40DFH   ;ENTRY POINT ADDRESS
01F8             00690 CASOF    EQU     01F8H   ;TURNS CASSETTE OFF
20FE             00700 OPCRV    EQU     20FEH   ;OUTPUT CR TO VIDEO
032A             00710 GPOP     EQU     032AH   ;GENERAL PURPOSE O/P ROUTINE
1BB3             00720 IPFN     EQU     1BB3H   ;INPUT FILE NAME
06CC             00730 REBAS    EQU     06CCH   ;REENTER BASIC
4188             00740 LOAD     EQU     4188H   ;LOAD COMMAND VECTOR
0296             00750 FINDSC   EQU     0296H   ;FIND SYNC
                 00760 ;
                 00770 ;
                 00780 ;
70BD             00790          ORG     70BDH   ;BEFORE BMON
                 00800 ;
70BD 2B          00810 END      DEC     HL      ;CORRECT ENDAD POINTER
70BE 225441      00820          LD      (ENDAD),HL; STORE END ADDRESS
```

```
70C1 CD1403   00830          CALL    READ2   ;READ ENTRY POINT FROM TAPE
70C4 22DF40   00840          LD      (ENTAD),HL; STORE ENTRY POINT
70C7 CDF801   00850          CALL    CASOF   ;TURN CASSETTE OFF
70CA CDFE20   00860          CALL    OPCRV   ;O/P CR LF TO VIDEO
70CD 21E771   00870          LD      HL,MESG1; POINT TN START MESSAGE
70D0 CDDD71   00880          CALL    OPMES   ;O/P MESSAGE TO VIDEO
70D3 2A5241   00890          LD      HL,(START); POINT TO START ADDRESS
70D6 7C       00900          LD      A,H
70D7 CDD171   00910          CALL    OP2AS   ;O/P 2 ASCII CHAR TO VIDEO
70DA 7D       00920          LD      A,L
70DB CDD171   00930          CALL    OP2AS
70DE 21EE71   00940          LD      HL,MESG2; POINT TO END MESSAGE
70E1 CDDD71   00950          CALL    OPMES   ;O/P MESSAGE TO VIDEO
70E4 2A5441   00960          LD      HL,(ENDAD); POINT TO END ADDRESS
70E7 7C       00970          LD      A,H
70E8 CDD171   00980          CALL    OP2AS   ;O/P END H.O.A.
70EB 7D       00990          LD      A,L
70EC CDD171   01000          CALL    OP2AS   ;O/P END L.O.A.
70EF 21F671   01010          LD      HL,MESG3; POINT TO ENTRY MESSAGE
70F2 CDDD71   01020          CALL    OPMES   ;O/P ENTRY MESSAGE
70F5 2ADF40   01030          LD      HL,(ENTAD);POINT TO ENTRY ADDRESS
70F8 7C       01040          LD      A,H
70F9 CDD171   01050          CALL    OP2AS   ;O/P ENTRY H.O.A.
70FC 7D       01060          LD      A,L
70FD CDD171   01070          CALL    OP2AS   ;O/P ENTRY L.O.A.
7100 180E     01080          JR      BEGIN
              01090 ;
              01100 ;
              01110 ;THE FOLLOWING SECTION OF THIS PROGRAM
              01120 ;WILL SETUP THE LOAD COMMAND VECTOR
              01130 ;
7102 018841   01140 INIT     LD      BC,LOAD
7105 3EC3     01150          LD      A,0C3H  ;JUMP INSTRUCTION
7107 02       01160          LD      (BC),A
7108 DD211071 01170          LD      IX,BEGIN
710C DD228941 01180          LD      (LOAD+1),IX ;SETUP ENTRY POINT
              01190 ;
              01200 ;
              01210 ;MAINLINE PROGRAM
              01220 ;
              01230 ;
              01240 ;
7110          01250 BEGIN    EQU     $       ;PROGRAM ENTRY POINT
7110 318842   01260          LD      SP,4288H; INITIALISE STACK
7113 AF       01270          XOR     A       ;CLEAR A REG
7114 325641   01280          LD      (PASS1),A; CLEARS PASS1 FLAG
7117 CDFE20   01290          CALL    OPCRV   ;O/P CR LF TO VIDEO
711A 3E2A     01300          LD      A,2AH   ;ASCII VALUE FOR *
711C CD2A03   01310          CALL    GPOP    ;O/P * TO VIDEO
711F CDB31B   01320          CALL    IPFN    ;INPUT FILENAME FROM K/B
7122 DACC06   01330          JP      C,REBAS ;REENTER BASIC
7125 D7       01340          RST     10H     ;POINT HL TO START OF FILENAME
7126 CA9719   01350          JP      Z,SNERR ;SYNTAX ERROR
7129 FE2F     01360          CP      2FH     ;TEST FOR /
712B CA1D03   01370          JP      Z,ENTRY ;JUMP IF /
              01380 ;THE FOLLOWING SECTION IS THE CASSETTE DRIVE
              01390 ;SELECT LOGIC
              01400 ;
712E 7E       01410          LD      A,(HL)  ;TEST IF DRIVE SPECIFIED
712F FE23     01420          CP      23H     ;#?
7131 2023     01430          JR      NZ,PRESEL
7133 23       01440          INC     HL
7134 7E       01450          LD      A,(HL)  ;DETERMINE REQUIRE DRIVE
7135 FE2D     01460          CP      2DH     ;-?
7137 C29719   01470          JP      NZ,SNERR
713A 23       01480          INC     HL
713B 7E       01490          LD      A,(HL)
713C FE31     01500          CP      31H     ;DRIVE 1?
713E 2809     01510          JR      Z,SETUP1
7140 FE32     01520          CP      32H     ;DRIVE 2?
7142 C29719   01530          JP      NZ,SNERR
7145 1E10     01540          LD      E,10H   ;DRIVE 2 MASK
7147 1802     01550          JR      COMTST  ;TEST , SWITCH
              01560 ;
```

```
7149 1E00      01570 SETUP1  LD      E,0      ;DRIVE 1 MASK
714B 23        01580 COMTST  INC     HL
714C 7E        01590         LD      A,(HL)
714D FE2C      01600         CP      2CH      ;,?
714F C29719    01610         JP      NZ,SNERR
7152 7B        01620         LD      A,E      ;RECOVER MASK
7153 23        01630         INC     HL
7154 1802      01640         JR      SELCAS
               01650 ;
7156 3E00      01660 PRESEL  LD      A,0      ;DRIVE 1 MASK
7158 CD1202    01670 SELCAS  CALL    CASON    ;CASON TURNS ON THE
               01680 ;CASSETE DRIVE SELECTED.
               01690 ;
715B CD9602    01700         CALL    FINDSC   ;THIS ROUTINE
               01710 ;SEACHES THE TAPE UNTIL THE SYNC
               01720 ;PULSE IS FOUND. IT THEN PUTS TWO ASTERIX IN
               01730 ;THE TOP RIGHT HAND CORNER OF THE SCREEN
               01740 ;
715E CD3502    01750 NOSYNC  CALL    READT    ;READS A BYTE FROM
               01760 ;TAPE AND RETURNS WITH THE BYTE IN THE A REG
               01770 ;NO OTHER REGISTERS ARE AFFECTED.
               01780 ;
7161 FE55      01790         CP      55H      ;TEST FOR SYNC
7163 20F9      01800         JR      NZ,NOSYNC; JUMP IF NO SYNC
7165 0606      01810         LD      B,6      ;LENGTH OF FILE NAME
7167 7E        01820 TSTFN   LD      A,(HL)
7168 B7        01830         OR      A
7169 2809      01840         JR      Z,NAMOK ;JUMP IF NAME MATCH
716B CD3502    01850         CALL    READT    ;I/P NEXT FILENAME CHAR
716E BE        01860         CP      (HL)     ;COMPARE FILENAMES
716F 20ED      01870         JR      NZ,NOSYNC; NO MATCH
7171 23        01880         INC     HL       ;POINT TO NEXT CHAR
7172 10F3      01890         DJNZ    TSTFN    ;DONE ALL SIX?
7174 CD2C02    01900 NAMOK   CALL    BLINK    ;BLINK ASTERIX
7177 CD3502    01910 NOCODE  CALL    READT    ;I/P NEXT BYTE
717A FE78      01920         CP      78H      ;END OF BLOCK?
717C CABD70    01930         JP      Z,END    ;LOAD COMPLETE
717F FE3C      01940         CP      3CH      ;START OF BLOCK?
7181 20F4      01950         JR      NZ,NOCODE; WAIT FOR CODE
7183 CD3502    01960         CALL    READT    ;I/P BLOCK LENGTH
7186 47        01970         LD      B,A      ;STORE B/L IN "B"
7187 CD1403    01980         CALL    READ2    ;I/P START ADDRESS
               01990 ;CHECKSUM CORRELATION BEGINS AT THIS POINT
718A 85        02000         ADD     A,L      ;UPDATE CHECKSUM
718B 4F        02010         LD      C,A      ;CHECKSUM IN "C"
718C 3A5641    02020         LD      A,(PASS1)        ;START OF PROGRAM
718F FE00      02030         CP      0        ;PASS 1?
7191 2008      02040         JR      NZ,NOT1 ;IF NOT
7193 225241    02050         LD      (START),HL; STORE START ADD
7196 3EFF      02060         LD      A,0FFH
7198 325641    02070         LD      (PASS1),A; SET PASS 1 FLAG
719B CD3502    02080 NOT1    CALL    READT    ;I/P BYTE
719E 77        02090         LD      (HL),A   ;O/P BYTE TO MEMORY
719F 23        02100         INC     HL       ;UPDATE MEMORY POINTER
71A0 81        02110         ADD     A,C
71A1 4F        02120         LD      C,A      ;UPDATE CHECKSUM
71A2 10F7      02130         DJNZ    NOT1     ;LOOP TILL E.O.R.
71A4 CD3502    02140         CALL    READT    ;I/P TAPE CHECKSUM
71A7 B9        02150         CP      C        ;COMPARE CHECKSUMS
71A8 28CA      02160         JR      Z,NAMOK ;TEST FOR E.O.F.
               02170 ;IN THE EVENT OF INCORRECT CHECKSUM THE PROGRAM
               02180 ;CONTINUES
71AA 3E43      02190         LD      A,43H    ;ASCII C
71AC 323E3C    02200         LD      (ASRIX),A; O/P C TO VIDEO
71AF 18C6      02210         JR      NOCODE
               02220 ;
               02230 ;
               02240 ;       SUBROUTINES USED BY SYSCPY
               02250 ;
               02260 ;
               02270 ;       SUBROUTINE HX2AS
               02280 ;
```

```
                02290 ;CONVERTS THE HEX VALUE IN THE A REGISTER
                02300 ;INTO TWO ASCII CHARACTERS WHICH ARE STORED
                02310 ;IN THE BC REG PAIR
                02320 ;
71B1 47         02330 HX2AS   LD      B,A      ;SAVE HEX CHARACTER
71B2 E60F       02340         AND     0FH      ;ISOLATE 4 LSB
71B4 CDC871     02350         CALL    CONAS    ;CONVERT A REG TO ASCII
71B7 4F         02360         LD      C,A
71B8 78         02370         LD      A,B
71B9 E6F0       02380         AND     0F0H     ;ISOLATE 4 MSB
71BB CB3F       02390         SRL     A
71BD CB3F       02400         SRL     A
71BF CB3F       02410         SRL     A
71C1 CB3F       02420         SRL     A        ;MOVE TO 4 LSB
71C3 CDC871     02430         CALL    CONAS
71C6 47         02440         LD      B,A
71C7 C9         02450         RET
                02460 ;
                02470 ;
                02480 ;
                02490 ;       SUBROUTINE CONAS
                02500 ;
                02510 ;CONVERTS THE A REGISTER CONTENTS FROM 0-F
                02520 ;INTO AN ASCII CHARACTER
                02530 ;
71C8 FE0A       02540 CONAS   CP      0AH      ;CARRY CLEAR IF A REG>=0A
71CA 3802       02550         JR      C,ADJAL ;ADJUST ALPHABETIC CHARACTERS
71CC C607       02560         ADD     A,7
71CE C630       02570 ADJAL   ADD     A,30H    ;CONVERT TO ASCII
71D0 C9         02580         RET
                02590 ;
                02600 ;
                02610 ;
                02620 ;       SUBROUTINE OP2AS
                02630 ;
                02640 ;OUTPUT 2 ASCII CHARACTERS IN THE BC REGISTER PAIR
                02650 ;TO THE O/P DEVICE USING GPOP ROUTINE
71D1 CDB171     02660 OP2AS   CALL    HX2AS
71D4 78         02670         LD      A,B
71D5 CD2A03     02680         CALL    GPOP     ;O/P TO DEVICE
71D8 79         02690         LD      A,C
71D9 CD2A03     02700         CALL    GPOP     ;O/P TO DEVICE
71DC C9         02710         RET
                02720 ;
                02730 ;
                02740 ;
                02750 ;       SUBROUTINE OPMES
                02760 ;
                02770 ;OUTPUTS ASCII MESSAGE POINTED TO BY THE HL REGISTER
                02780 ;PAIR USING GPOP. THE ASCII MESSAGE MUST BE TERMINATED
                02790 ;WITH 04 (E.O.T.).
                02800 ;
71DD 7E         02810 OPMES   LD      A,(HL)   ;INPUT CHARACTER
71DE FE04       02820         CP      4        ;E.O.T.?
71E0 C8         02830         RET     Z        ;MESSAGE COMPLETE
71E1 CD2A03     02840         CALL    GPOP     ;O/P CHARACTER
71E4 23         02850         INC     HL       ;UPDATE MESSAGE POINTER
71E5 18F6       02860         JR      OPMES
                02870 ;
                02880 ;
                02890 ;
                02900 ;       MESSAGES FOR OUTPUT
                02910 ;
71E7 53         02920 MESG1   DEFM    'START='
71ED 04         02930         DEFB    4
71EE 20         02940 MESG2   DEFM    '    END='
71F5 04         02950         DEFB    4
71F6 20         02960 MESG3   DEFM    '    ENTRY='
71FF 04         02970 FINSH   DEFB    4
                02980 ;
                02990 ;
7102            03000         END     INIT     ;DEFINE ENTRY
00000 TOTAL ERRORS
```

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

Next month's issue will contain at least the following programs plus the usual features, articles, news, letters, etc..

FIGHTER SQUADRON (L1)
Would you like to be a squadron leader and destroy Gestapo Headquarters, all you have to do is wait till next month.

FOOD REQUIREMENTS (L1)
Eating too much, too little, or don't you know. Next month you'll be able to find out.

DRAW II (LII/4K)
This program is for our more artistic readers, it lets you sKetch on the screen, make a negative of your sKetch, save it on tape, load it from tape and lots more.

BLEEPER (LII/4K)
A little beauty this one, makes your '80 bleep every time a key is pressed, not only that but pressing "ENTER" or "BREAK" makes a different bleep. It's got a keyboard debounce routine built in as well.

LOWER CASE DRIVER (L2 ml.)
This machine language program will allow you to display lower case letters on your screen in BASIC programs, if you have a Lower Case modification fitted as described in MICRO-80 Issue 5, April 1980.

FLASHING CURSOR (L2 ml. & BASIC)
A flashing cursor program that really works!! Written by Eddy Paay, this program works with both LII and Disk BASICS. The second program, in BASIC, allows you to adjust the size/shape of the cursor and its rate of flashing so that you can, if you wish, have a transparent cursor. The ml. program demonstrates how to use a ROM routine to reset MEMORY SIZE from within your own program.

***** APPLICATION FOR PUBLICATION OF A PROGRAM IN MICRO-80 *****

Tick where appropriate.

To MICRO-80
Please consider the enclosed program for...

(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 separated 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.

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 –

QTY.	CAT NO	DESC.	ADV. PRICE

SUB TOTAL ....

LESS 10% ....

FIND CHEQUE FOR **TOTAL** ....

SEND **FREIGHT FREE** TO

NAME ....

ADDRESS ....

.... P/ Code ....

TANDY ELECTRONICS DEALER

CONQUEST ELECTRONICS PTY. LTD.

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

MICRO-80

LEVEL II ROM REFERENCE MANUAL

by Edwin Paay

Published by MICRO-80 PRODUCTS

Written by Eddy Paay, the LEVEL II ROM REFERENCE MANUAL is the most complete explanation of the Level II BASIC interpreter ever published.

Part 1 lists all the useful and usable ROM routines, describes their functions explains how to use them in your own machine language programs and notes the effect of each on the various Z 80 registers.

Part 1 also details the contents of system RAM and shows you how to intercept BASIC routines as they pass through system RAM. With this knowledge, you can add your own commands to BASIC, for instance, or position BASIC programs in high memory—the only restriction is your own imagination!

Part 2 gives detailed explanations of the processes used for arithmetical calculations, logical operations, data movements, etc. It also describes the various formats used for BASIC, SYSTEM and EDITOR/ASSEMBLER tapes. Each section is illustrated by sample programs which show you how you can use the ROM routines to speed up your machine language programs and reduce the amount of code you need to write.

The LEVEL II ROM REFERENCE MANUAL is intended to be used by machine language programmers. It assumes a basic understanding of the Z 80 instruction set and some experience of Assembly Language programming. But BASIC programmers too will benefit from reading it. They will gain a much better insight into the functioning of the interpreter which should help them to write faster, more concise BASIC programs.

MICRO-80