

MICRO-80

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

ISSUE 10
AUGUST 1980

PRICE:- AUS. \$2.50
- NZ. \$3.00

***** CONTENTS *****

	PAGE
ABOUT MICRO-80	1
EDITORIAL	2
READER'S REQUESTS	5
G.T. BASIC - Part 2 in a series	6
INPUT/OUTPUT - Letters to the Editor	11
BLOCK MOVE	14
'80 USERS GROUPS	16
ESF FORUM	17
SYSTEM 80 - 32 CHARACTERS PER LINE - CORRECTION	18
BOOT	18
<u>HARDWARE SECTION</u>	
HOW TO GO BEYOND 16K WITHOUT GOING BANKRUPT!	20
<u>SOFTWARE SECTION</u>	
ESCAPEE..... L1/4K	27
THE WORLD L1/4K	28
CUP '80 L1/4K	28
CUP '80 L2/4K	29
LOTTO PREDICTOR DB/32K	29
TRIANGLE L2/16K	33
THE WORLD L2/4K	37
SOLVER L2/4K	37

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

All enquiries should be directed to:-
MICRO-80 PO BOX 213, GOODWOOD, SA 5034

Printed by:-
Shovel & Bull Printers,
312 A Unley Rd.,
Hyde Park, SA 5061

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

MICRO-80

**** ABOUT MICRO-80 ****

EDITOR: Ian Vagg
ASSOCIATE EDITORS:
SOFTWARE: Peter Hartley
HARDWARE: Edwin Paay

MICRO-80 is the only Australian monthly magazine devoted entirely to the Tandy TRS-80 microcomputer and the Dick Smith System 80. It is available by subscription, \$24.00 for 12 months or by mail order at \$2.50 per copy. A cassette containing all the programs in each month's issue is available for an additional \$3.50 or a combined annual subscription to both magazine and cassette is available for \$60.00. Special bulk purchase rates are also available to computer shops etc. Please use the form in this issue to order your copy or subscription.

The purpose of MICRO-80 is to publish software and other information to help you get the most from your TRS-80 or System 80 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 2 BASIC and one in DISK BASIC (or disk compatible Level 2). We also publish Utility programs in Level 2 BASIC and Machine Language. At least every second issue has an article on hardware modifications or a constructional article for a useful peripheral. In addition, we run articles on programming techniques both in Assembly Language and BASIC and we print letters to the Editor and new product reviews.

**** COPYRIGHT ****

All the material published in this magazine is under copyright. That means that you must not copy it, except for your own use. This applies to photocopying the magazine itself or making copies of programs on tape or disk.

**** 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.

***** EDITORIAL *****

This is our first international edition. MICRO-80 magazine is now being distributed in the U.K. and we would like to welcome all our new U.K. readers and tell you a little of our background and what we stand for.

MICRO-80 commenced some 10 months ago in Australia. By now it will be obvious that we are not a glossy, big-budget publication. Our aims, first and foremost, are to support users of the Tandy TRS-80 microcomputer and the more recent System 80 microcomputer or, as it is known in Britain, the Video Genie. The people who produce MICRO-80 are '80 enthusiasts ('80 is our generic term for the TRS-80/System 80/Video Genie, it saves ink and is easier to read than that mouthful). Much of the material we publish is written by our readers, for which we pay a publication fee. You won't make your fortune writing for MICRO-80, our budget is not big enough for that but you will earn some useful extra income which you can point to in those periodic arguments with your spouse when the inevitable question comes up "what use is that wretched computer, anyway?"

Significantly, many of our subscribers who write-in, commence by saying "I am a member of MICRO-80". That is the atmosphere we want to foster, a large, open club where we can all exchange information, programs etc. to our mutual benefit.

To help defray the expenses of producing this magazine, we have a sister business called MICRO-80 PRODUCTS. MICRO-80 PRODUCTS publishes software written by our readers, at sensible prices, and imports goods into Australia for resale by mail order, again at very sensible prices. For the moment, we are not launching MICRO-80 PRODUCTS in the U.K. If any of our U.K. readers would like to purchase items from MICRO-80 PRODUCTS you should write to us in Australia, direct. The easiest way to pay would be by Barclaycard/VISA or ACCESS, both of which we honour. That way, we can charge the exact exchange rate and airmail cost to your credit card. If you prefer to pay by cheque or money order, that is alright too. Simply make it payable in sterling at an exchange rate of Aus.\$2.07 per pound sterling and add 1.00 pound for air-mail postage per program or 2.00 pounds for a book. If the demand from the U.K. for MICRO-80 PRODUCTS goods is great enough we will, of course, make arrangements for direct distribution there.

We would be happy to publish programs, articles, etc. written by our U.K. readers on the same terms as apply to Australian readers. We would also like to establish a regular U.K. correspondent to keep us all up-to-date with happenings in Britain vis-a-vis '80 microcomputing. If you would like to contribute, please write directly to us in Australia. For the moment, all editorial matters will be dealt with in Australia, our U.K. address is a sales office only.

MODEL III TRS-80

Those of us with keen eyesight and an enquiring mind (no, not the Editor but Eddy Paay!) will have noticed something unusual on page 3 of the new Tandy Australia catalogue. There, in a tiny colour picture, about the size of a postage stamp, is a strange looking computer with a keyboard very much like the Model I except that it is integral with the monitor and, to the right of the monitor screen there are two 5-1/4 inch disk drives built-in. Naturally, we contacted Tandy to find out all about it.

This is a picture of the TRS-80 Model III, which has just been released in the U.S.A. (we are not sure if any have actually been delivered yet) and should

come to Australia in January 1981. In Tandy's words, it is designed to fit between the Model I and Model II and is aimed at the small businessman. Interestingly enough, it sells in the U.S.A. in basic, 11/4K form, for \$699 which compares with a price of \$499 for the Model I 11/4K machine. The essential difference is that the expansion interface is built in, so that you do not have to buy extra boxes to up-grade all the way to disks which, incidentally, are 40 track dual-density drives. One other difference is that the BASIC interpreter now occupies 14K, i.e. it fills up that memory space around 3000H that is reserved in the Model I memory map as being for "future DMA Devices". What additional goodies may be in that extra 2K of ROM, we do not yet know. Apparently, it will be possible to convert Model I software to run on the Model III. Oh, one other thing, the Model III has two cassette speeds, 500 band and 1500 band. Incidentally, there is a new, square device to the right of the numeric key pad - could that be a built-in cassette deck or a System 80/Video Genie?

MODS TO THE SYSTEM 80/VIDEO GENIE.

Despite Jim Rowe's (Dick Smith Electronics Technical Director) assurances in all his Technical bulletins that very few pre-recorded cassettes refuse to load into the System 80, he has described how you can add a volume control to the cassette deck and modify the LED to act as a peak-level indicator. Apparently, this has not fallen on deaf ears in Hong Kong and before long, System 80's and presumably Video Genies, will be shipped with a slide type volume control on the keyboard panel and a VU meter, no less, mounted in the sloping panel adjacent to the tape counter. We have installed the volume control/LED mod for one of our readers in Adelaide, with very good results so these mods should clear up the cassette loading problems once and for all. Well done Dick Smith Electronics who, we are sure, has been the driving force behind this improvement.

Future shipments of System 80's/Video Genies will also be fitted with the two missing keys, right arrow and CLEAR, so the most glaring examples of non-compatibility between the two families of 80's will disappear.

TRS-80 2 CHIP ROM's

Tandy started shipping 2 chip ROM sets a few months ago. From one point of view it is surprising that it was so long in coming, since the original A board was clearly designed to accept 2 chip ROM's. Now that 2 chip ROM's are here, a few things have changed. First of all, there are two different sets of ROM's.

Part Nos. 8043364 and 8043732 are for all except G boards and these ROM's still need to be used in conjunction with the XRX III loading modification.

Part Nos. 8044364 and 8044732 are for G boards and do not require the XRX III loading modification. The checksums for these latter two ROM's are:

ROM A	B078
ROM B	DA45
ROM C	4006

ROM A and ROM B are in the first chip, ROM C is in the second chip. One interesting/disturbing change in the most recent TRS-80's shipped is that the character generator no longer has lower-case letters in it but instead, appears to have some Japanese characters. Suspicious souls that we are, we suspect that this is Tandy's way to foul-up the cheap, lower case mod. kits that are around. If that is the case, it is pretty dirty pool.

TANDY'S LOWER-CASE MOD.

We are not particularly enamoured with Tandy's pricing policy in Australia anyway, but its lower case Mod. kit must be one of the greatest rip-offs ever. It consists of three IC's, two DIL sockets, a driver program (with bugs in it) on cassette, and some documentation. It sells for \$99 + fitting.

Two of the IC's are 2102L RAMs which Tandy itself sells for \$4.95 each. The other is a character generator, i.e. a mask-programmed ROM which is specially made for the TRS-80 (as far as we know) but such ROMs generally sell for \$7 - \$9, let's say \$10, to be generous. The two DIL sockets, Tandy sells for 60 cents and 69 cents respectively and the cassette \$1.95 say. There is a little labour included too, in that the two RAM chips are soldered to each other pick-a-back and there are a couple of wires joining them to the character generator. Let's allow \$10 for that labour, which is very generous considering that it would be done in a factory. Finally, let us allow 86 cents for documentation. Therefore, the total value of the lower case mod kit at Tandy's normal prices is \$34.00! That, Tandy sells to you for \$99, about 3 times the sum of the individual prices which include their normal mark-up anyway. If that is not profiteering on a grand scale then we don't know what is. The key which enables Tandy to get away with this, of course, is the character generator. However, this will not be an obstacle for much longer as we are already making arrangements to obtain supplies of a suitable substitute. In the meantime, perhaps the decision-makers in Tandy will read this and feel so embarrassed that they will drop their price to a reasonable level (See, we said we were naive enthusiasts).

TANDY'S ANSWER TO LAST MONTH'S COMPLAINTS

You will remember that, one way and another, we had a few harsh words to say about some of Tandy's other policies last month. As promised then, we have provided Tandy with space to reply. Our first criticism related to the decision to oust TRS-80 Users group from Tandy stores (computer centres excepted). Tandy's reply is that Tandy Australia is simply following orders. The directive is world-wide and came from Fort Worth.

The second criticism was made by Peter Hartley in an open letter in which he took Tandy to task over its apparent policy of removing all owner-fitted mods before servicing TRS-80's, then charging for the privilege. Tandy's reply is that it is their policy not to remove user-fitted mods unless the serviceman concerned considers that they are causing/aggravating the problem. It was pointed out that so many different mods have been made to TRS-80's that it is virtually impossible for the serviceman to become familiar with all of them. In the case of Peter Hartley's machine, there were a lot of mods (true - Ed) and the serviceman obviously felt that these were causing the trouble. In the event, it turned out to be faulty buffer IC's but by then it was too late.

Well, there you have it, you have heard both sides of both stories and you must judge what is reasonable. (See also Input/Output).

ON PRESENTATION OF MICRO-80

This month, we have reverted to the production technique of a few months ago and have some mixed type-faces as well. We are sorry about this and assure you that it is a temporary state of affairs only. Next month, we should have a very much improved presentation with consistent type-face, photographic reduction and right-justification. All this will be done via our new Olivetti ET-121 daisy wheel typewriter which we have recently converted to interface with the '80. We will shortly be making these converted typewriters available for sale, with Olivetti's blessing so, if you are in the market for a fast (20 cps) quiet, automatic correcting typewriter that you can also drive from your computer, then be patient for just a little longer and you will be able to buy the MICRO-80 converted Olivetti ET-121 for under \$2000.

CASSETTE PROBLEMS

About the time we were producing last month's cassettes, our duplicator started to develop a fault. The plate holding the heads for the master cassette became looser and looser, unbeknown to us. This meant that signal strength varied and a number of bad tapes were produced. Other tapes produced at the same time were also affected. The duplicator is fully repaired now and working better than ever but, if you have a faulty tape, please send it back to us and we will replace it, free of charge.

** MELBOURNE COMPUTER SHOW **

MICRO-80 will be on stand 169 at the 8th World Computer Conference to be held in the Melbourne Exhibition Building from 14 - 19 October. We will have some interesting items on display including the SYSPAND 80 adaptor which provides your System 80 with a printer port and Tandy bus and our prototype Olivetti ET-121 conversion. You will also be able to see a Stringy Floppy demonstrated and a B51 bare drive. We will have our whole range of software with us including some interesting new programs. We are looking forward to meeting as many of our Victorian readers as possible, so come along and make yourselves known.

- 00000 -

***** 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

- 00000 -

**** G.T. BASIC **** part two of an occasional series
by Peter G. Hartley.

Last month we looked briefly at simple MATH in COMMAND MODE and NUMERIC VARIABLES. We also touched briefly on using the PRINT instruction. This month we'll have our first look at the FOR / NEXT LOOP and go into more detail on PRINT statements. We'll also have a look at DATA STATEMENTS and introduce PROGRAMS.

**** THE AMAZING FOR / NEXT LOOP ****

Well, this IS throwing the raw beginner in at the deep end. Lesson two, and already we have this funny thing called a LOOP. No, not an I.U.D., but a very powerful instruction.

Let's assume that we want to print out the "twelve times table". Remember, we are still in the COMMAND MODE, and if you were to go through all the fuss of entering the 12 separate commands to generate (12*1), (12*2), (12*3) etc., well, there wouldn't be any space left on the screen for all the answers. What a FOR / NEXT loop does is solve the problem of getting the '80 to do something for a specified number of times, even though you only tell it what to do once.....

```
FOR K=1 TO 12:PRINT K:NEXT (in Level I - F.K=1 TO 12:P.K=N.K)
```

will produce a list from 1 to 12. Now, we know that to multiply, we only have to use the "*" and the value, so lets try...

```
FOR K=1 TO 12:PRINT K*12:NEXT (in Level I- F.K.=1 TO 12:P.K*12=N.K)
```

- notice that in Level I, you have to specify the variable used in the loop when you use the NEXT. In Level II this is optional, which is rather more useful, as we will see later in the series.)

Well now, that worked fine - but it doesn't really tell you what's going on... fine for you and me, because we know what we wanted, but perhaps some little labels and notices would help, so now try...

```
CLS:PRINT"TWELVE TIMES TABLE":FOR K=1 TO 12:PRINT K;" TIMES 12 =" ;K*12:NEXT
(for Level I - P."TWELVE TIMES TABLE": F.K.=1 TO 12: P.K;" TIMES TWELVE
=" ;K*12:N.K.
```

That looks a lot better, doesn't it.

O.K. Let's take that line of instructions apart and see what we did.

```
CLS          clears the U.D.U.
:           end of instruction marker.
PRINT       just what it says.
"TWELVE TIMES TABLE"
            as soon as the '80 sees the quotes, it knows
            that it is to treat what is inside the quotes
            as something to be PRINTED - not as something
            on which to perform mathematical miracles.
            By putting it in quotes, we turn in into what is
            called a STRING.
```

```

:          end of instruction marker.
FORK=1TO12 start counting from 1, and store the value in
           variable K. Do everything from this point on
           to the instruction NEXT, once each time K is
           INCREMENTED by one. The last time is when K=12.
:          end of instruction marker.
PRINT     again...
K         it isn't in quotes, so it is a variable, so
           print the number stored in K.
:         stop printing - there is more to come on the
           same line immediately after what has just been
           printed.
"TIMES TWELVE ="
           it is in quotes, so it is a string. The last
           instruction has not ended yet, so print it.
;         stop - more to follow
K*12     find the value of K, and multiply it by 12 - and
           as we are still doing the PRINT instruction -
           print the answer.
:         end of instruction marker.
NEXT     Level II - find the last FOR statement and increment
           the appropriate variable. If it's bigger than
           the specified limit, stop and come back here.
           If it isn't bigger, start again from the first
           instruction following the FOR statement.
           Level I - increment the variable specified in the
           NEXT statement and if it is now bigger than the
           specified limit, stop and come back here.
           If it isn't bigger than the limit, start again,
           immediately after the FOR statement.

```

**** MORE THAN ONE LINE AT A TIME ****

When we use the COMMAND MODE we are very restricted in the type of problem that we can set. We can only use one multiple statement line to tell '80 what to do, and we have just used the ":" to separate the statements. In Level II, we are restricted to about 248 characters for the instruction line. In Level I the problem is even greater, because we are limited to 70 characters. Part of the solution is to put the statements into the form of a PROGRAM.

We can define a PROGRAM as being a series of instructions that are to be executed in a clearly defined sequence.

If we accept this definition, we can also see that we can in fact execute many programs by entering a whole series of separate statements while in the command mode. If you do this you will find that the '80 will remember any values that have been stored as NUMERIC VARIABLES. Unfortunately, what it will not remember is useful things like FOR / NEXT LOOPS, and other instructions that it has already executed, so the rest of the solution is to store the PROGRAM in the '80's memory.

```

Try      K=100
         PRINTK
         K=K+1
         PRINTK

```

following each instruction with an <ENTER> or a <NEW LINE>, and you will see that the '80 really does remember.


```
Try   FORK=0T010
      PRINTK
      NEXT
```

and you will also see that the '80 really does forget.

Our definition of PROGRAM said that the instructions had to be executed in a clearly defined order. To write a program into memory, we start each line with a number, to define the order.

```
Try   10 FORK=0T010
      20 PRINTK
      30 NEXT
      RUN
```

following each line with <ENTER>, as before. Now type LIST <ENTER>. Amazing! ol' '80 remembered, and if you type RUN <ENTER> it will repeat the sequence again.

Now, the numbers we used need not have been 10, 20 and 30. We could just have well have used 100, 200 and 300, or 1, 2 and 3. BUT, if we used 1, 2 and 3, and then discovered that we had left some instructions out, there would not be anywhere to insert them.

```
Try   25 PRINT"IS FOLLOWED BY";
<ENTER> LIST <ENTER> and you will see that the extra line has been inserted
into the correct place in the listing. So now you can add...
```

```
      3 CLS
      5 PRINT"THE NUMBER";
      35 PRINT"WHICH IS FOLLOWED BY";K
and   40 END
```

which tidies everything up, and presents a new surprise for the beginners. The FOR / NEXT LOOP said "K=0T010" and we have K having a value of 11. That deserves some explanation.

Most versions of BASIC would have K equalling only 10 by the time the program got to line 35. However, the MICROSOFT BASIC increments the counter BEFORE testing the its value, so after any FOR / NEXT LOOP the value of the counter will be one increment past the specified end of the loop. That is important. Sometimes it is useful - sometimes a damn nuisance. Either way, remember this well.

Last month I promised to show you how to speed up FOR / NEXT LOOPS by as much as 37%. Now this will not work in Level I, and the rule derived from this is not applicable in Level I, but Level II users should try the following, and time the runs.

```
TEST ONE
      10 FORK=0T010000:NEXT:PRINT"FINISHED"
```

```
TEST TWO
      10 DEFINTK:FORK=0T010000:NEXT:PRINT"FINISHED"
```

O.K?

```
* * * * *
* G.T.BASIC RULE TWO - WHENEVER POSSIBLE - USE AN INTEGER *
* * * * *
```

Don't worry if you are a raw beginner, and don't understand INTEGER, because we'll be discussing this term in a later article.

Now that we know how to put a program into memory, we need to know how to wipe it out again. One magic word - "NEW". Try it.

**** TIDYING UP PRINT STATEMENTS ****

Type in the following...

```
10 CLS
20 FORK=1T0100
30 PRINTK
40 NEXT
50 END
```

ending each line with <ENTER> or <NEW LINE>... and then RUN <ENTER>.

Now that looks pretty untidy, doesn't it. In fact, you cannot see any more than the last 12 numbers. Now type in a new line 30...

```
30 PRINTK,
and then RUN <ENTER>.
```

Better? What has happened is this... the comma, after the PRINT K instruction, tells the '80 to move along to the next PRINT ZONE, and then wait there for the next PRINT operation. There are eight print zones, right across the V.D.U. If you have Level II, you can move the CURSOR (that's the little line that shows where the next character will print on the screen) to the start of the next PRINT ZONE by pressing the right arrow key. If you have a System '80, you haven't got a right arrow key, and will have to press <SHIFT> <CNTRL> and <I> all at once. (This also works with the Tandy Level II)

Try typing in another version of line 30...

```
30 PRINTK;
RUN <ENTER>
```

and you will see yet another alternative. The semi-colon tells the '80 to hold its position on the V.D.U. and wait for the next PRINT instruction. In actual fact the '80 leaves a space in front of the value of any variable. This is to leave room for a negative sign, should it be required. It also leaves one space after the number, to stop the values running into one another on the screen.

Incidentally, the raw beginners amongst you have now discovered how to replace a line in a program. Just type a new line with the same number, and the old line gets replaced. The replacement does not actually occur until you press <ENTER>, so if you should change your mind about replacing the line you can either backspace the replacement line away, and do something else, or simply press <BREAK> and the '80 will ignore what you were typing. Level II users have a whole wealth of other EDITING facilities, but these will be the topic of later discussions.

**** DATA STATEMENTS AND HOW TO USE THEM ****

A DATA STATEMENT is an easy method of storing information, within a program, so that the '80 can "look up" the data each time the program is run. This saves you from having to INPUT the data each time.

******* WARNING *******

There are three known "bugs" in the MICROSOFT Level II BASIC. One of these relates to the reading of data from data statements. In some circumstances, on some Level II units only, the READ function fails to operate properly. Good practise is as follows...

IN ANY LEVEL II PROGRAM THAT READS DATA FROM DATA STATEMENTS, ALWAYS MAKE THE FIRST PROGRAM LINE READ AS FOLLOWS...

```
10 POKE 16553,255
```

THIS LINE SHOULD BE REPEATED AFTER ANY "INPUT*-1" STATEMENT IF DATA IS STILL TO BE READ OR REREAD

This will ensure that the function always works as intended.

Now try...

```
10 POKE 16553,255 (Level I users leave this line out.)
20 DATA 1, 3, 5, 7, 9, 2, 4, 6, 8, 10
30 CLS
40 PRINT"ODD NUMBERS BETWEEN 1 AND 10 ARE ";
50 FORK=1TOS
60 READA
70 PRINTA;
80 NEXT      (Level I - use N.K )
90 PRINT"EVEN NUMBERS ARE ";
100 FORK=1TOS
110 READA
120 PRINTA;
130 NEXT      (Level I - use N.K )
140 END
RUN      <ENTER>
```

Pretty neat, eh? Next time we will discuss exactly what happened with that DATA STATEMENT, and use some of what we have learned to start creating a Hex-to-Decimal conversion routine. You will also be introduced to STRING VARIABLES, and I'll try to show some of the old hands that sometimes, some of the old Level I techniques can be faster than the sophisticated logic of Level II.

Your letters, be they filled with praise, criticism or ten dollar notes, are appreciated. This series is intended to provide a service to readers, and feedback is vital if this series is to prove successful.

P.G.H.

CONTINUED FROM PAGE 13

(The graphical resolution on the TRS-80 is determined by the hardware in the machine. We have just taken delivery of a hardware modification board produced by Programma International in the U.S.A. We will shortly publish a review of this device - Ed.)

Letter from Mr. S.J. Stewart, St. James, N.S.W.

I have a System-80. How do I load machine language programs from tape and run them? I have tried "SYSTEM" "TITLE" etc. but have had no success.

(In order to load a machine language program::

1. Type SYSTEM and press <NEW LINE>

2. Answer the *? with the name of the program, press <NEW LINE>

You should then get two asterisks in the top right hand corner as for a BASIC program but the right hand asterisk turns on for 4 seconds, off for 4 seconds on again for 4 seconds and so on. When the program has finished loading, *? will again be displayed on the screen. Type in / and press <NEW LINE> when the program should commence running. If the asterisks do not appear or do not flash as described or if one changes to a C, indicating a Checksum error, then you have recorder/tape problems - Ed)

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

From: Mr. J. Grigg, West Geelong, Vic

With reference to the article in your July issue; sound effects by Ron Sully. If any of your readers are experiencing trouble with a program incorporating sound effects; that is after running it, CSAVEing it and then, at a later date, trying to run it after CLOADing. All that is necessary to fix the program is to CLOAD it as it is and Poke 17129,220 and Poke 17130,57. This works as long as Line "0" has been filled with spaces in the first place.

The sound effects program inserts machine code into the information space; in this machine code there are some 0's. This confuses the Basic Interpreter into thinking there are more lines in the program as 0 is recognised as an end of line marker.

It seems that upon CSAVEing the program, Loc. 17129 and 17130 are set to the first location after the first 0 encountered. Thus when we CLOAD again there is confusion.

I hope this information will be of help to someone as I spent about 5 hours all told in finding it out. (by experimentation).

From: R.D.Binzer, Glen Waverley, Vic.

I am responding to your August 1980 issue with reference to the "Open Letter to Tandy (Australia) by Peter Hartley.

With due respect to Peter Hartley, his letter offers little other than an outpouring of complaint with no and I repeat no, reference to his directions to Tandy in the first instance. All too often these days in all media areas there are instances of complaints with, in a lot of cases, no description of how the complaint arose and it seems easy to air a complaint, (instance - publication of his letter) without necessarily knowing whether the customer or the supplier was at fault.

We would all like something for nothing???, but the justification for whinging if we cannot get it can be debated.

Peter answers his own initial question in the affirmative, possibly because that suits his argument. However, I feel reasonably sure that had the vehicle referred to been recently purchased and was within its service period, it could quite easily have been referred back to him without repair, or alternatively repaired only upon his explicit instructions.

In these days of Consumer Protection and Trade Practice regulations there would be few if any suppliers repair items that had been altered, or modified without explicit instructions from the customer.

I do agree that if, as Peter describes, the brief and discourteous attention he received is accurate, then Tandy is certainly at fault, but again I repeat that Peter offers his complaint without reference to how the complaint arose in the first instance.

In defence of Tandy, I offer that in the approximate 2-1/2 years it has taken me to "develop" from the original 4K Level 1 to 48K Level 11 Twin Disc, with the progressive add-in and conversion of memory, I have always found that the service, attention and help offered by both Tandy store and technical personnel to be courteous and attentive and could only commend Tandy's attitude.

A recent "Glitch", which proved difficult to trace, offered on-site service. loaned components to complete a particular job on time and I could not have asked for any more courteous or attentive service and final correction of the problem.

I do trust that Peter's problem has been overcome, perhaps he struck a bad day - we all have them - we all make mistakes - this may have been one of Tandy's, but on my experience I could only doubt.

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

May I make a further suggestion, that your readers may find helpful.

From time to time there are a series of sub-routines that repeat from program to program. It might be of assistance if you could have a SUB-ROUTINE section that folk could contribute to.

For instance, recently I was looking for a way to make a blinking cursor. The answer to this could be one of the sub-routines. I've also been looking for a way to SET a graphics unit that is not a 'block' but could be a "+" or a ".", so that graphs could be drawn with these. This would make the graphs easier to read where there are 2 or 3 on the screen - as in the Biorhythm program. Neither of these has been solved by me - but some of your experienced men/women may have. Also when are you going to have programs available on disk for those of us without a recorder. Having to borrow a recorder is a bit of a hassle.

(We have put your suggestions into Readers Requests. We hope to be able to supply programs on disk within 2-3 months - Ed.)

Letter from+ Mr. J.G. Descovich, Caloundra, Qld.

I have fitted your 'Lower Case Modification' to my TRS-80, LII, 16K and am very pleased with the results. I am also looking forward to Eddie Paay's shift key reversal. However, the access program gave a little trouble at first, perhaps some others may be saved the frustration. My machine stopped execution with an O.V. Error in line 60. After dithering about a bit, checking the program, re-reading instructions, etc. I finally tried inserting line 15 ON ERROR GOTO70 as a last ditch stand. It worked and stays in my program. Keep up your good work, thanking you.

From: K.L. Dansie, Broken Hill, NSW.

I note that a number of users have loading problems with the System 80.

Having investigated some local users' problems, with various makes of microcomputers I would like to offer the following.

Apart from non-compatible commercial tapes the commonest causes of failure to load are:-

(a) failure to advance tapes past the leader before commencing to CSAVE. I can find no reference to this point in the System 80 Manuals. In view of the prevalence of tapes with leaders this vital operatin cannot be stressed too strongly.

(b) failure to erase up to and slightly beyond the proposed starting point when recording over a previous program. This is dealt with in Section 6.1 of the System 80 Users Manual. Using this "overlapping" technique there seems to be no necessity for the use of the bulk eraser recommended by David Lien. (Basic Computer Language - Radio Shack).

One of the local '80 machines suffered from a dry-joint in the inbuilt recorder. The others appear to be working satisfactorily.

The most puzzling loading problem was with an external Philips N2227 recorder, an excellent little machine which recorded O.K. but refused to load. After juggling volume and tone controls, examining wave-forms of output and tearing out lumps of hair, careful examination of the recorder circuit revealed that the output plug was reversed, the active lead being connected to the outer pin with the centre one earthed.

Consequently, on connecting the recorder/System 80 harness the signal was effectively earthed through the recorder input plug. An adaptor to reverse the output leads cured the trouble. But, N2227 owners, beware!

I am not connected with Dick Smith in any way but am merely an enthusiastic user of the System 80 which I have now mated with an Itoh 8300P Printer. Results so far are very encouraging.

INSTRUCTIONS FOR SAVING SARGON II ON ESF WAFER by John Castle

1. Switch on machine.
 2. Using SYSTEM, load the boot loader (Sargon). When second "*" appears, push reset button to return to BASIC.
 3. Enter - POKE 17565,0: POKE 17566,0: POKE 17567,0. This will cause RESET button to act normally after Sargon is loaded.
 4. Enter SYSTEM; Type /17428. This will now load the rest of the program.
 5. After it is loaded, press RESET button - BASIC.
 6. SYSTEM followed by /12346 - initiate ESF.
 - 7.* Sargon II uses RESET button to return to Sargon. If you wish to preserve this then:-
Enter POKE 17565,34: POKE 17566,7: POKE 17561,64.
 8. To save on wafer (20'):-
nSAVE 1, 17129, 15572, 17562
 9. Only RUN Sargon II after it is loaded (SAVED).
- * This step is optional.

From Mr. R.N.Wall, Melbourne.

Can you use the disk drive controller interface that is supplied with the TRS-80 when using the 8 inch disk drive, or is a special interface required for driving 8 inch disk drives?

(Some additional hardware is required to run an 8 inch disk drive on a standard Tandy expansion interface. These are available in the U.S.A. and are generally mounted in the cable to the disk drive - Ed.)

From Mr. P. Chapman, Auckland, N.Z.

The statement at the top of page 26 in the June Issue of MICRO-80, that one cannot effect the Level I + II modification if Level I is resident in two chips, is not correct. This is done by proceeding exactly as laid down in the magazine except that both chips are placed in Z34 socket soldering one above the other in pick-a-back fashion. This works fine as I have proved in my own TRS-80.

(Thank you Mr. Chapman. Readers please note, we have not tried this modification ourselves and therefore can accept no responsibility for its performance. - Ed.)

From G.L. Paterson, P.O. Box 437, Atheron, Qld. 4883.

Could MICRO-80 provide an open "vetting" service to analyse and improve programs written by inexperienced readers, especially in the technical fields. If this is not in your area, can you recommend anyone who might be able to take on such tasks for we micro users?

(I am afraid that MICRO-80 is not in a position to offer such a service. Perhaps some of our readers could, however. Please contact Mr. Paterson direct - Ed.)

Letter from Mr. R.D. Hall, North Geelong, Vic.

Could you tell me if there is any way of producing better graphics on the TRS-80, e.g. more realistic curves instead of square patterns? Is it software controllable, e.g. with a higher level language such as Pascal, or is it a hardware problem? If it is hardware, do you think it possible to modify with better or more chips?

***** BLOCK MOVE *****
by E. R. Paay

After reading the title of this article some of you may ask what this is all about. Well, picture this: You just received your latest issue of Micro-80 in the post and found a crash-hot machine language program which makes the cursor flash and displays rude comments while it runs BASIC programs etc. Just think of it - all your mates will be asking how you did it. So you load your favourite monitor program in memory to type in ten pages of hex from a hex dump in the magazine and to dump it to tape afterwards. But then it happens! After carefully typing in five pages suddenly the cassette relay buzzes the screen fills with funny characters, the keyboard and reset button are locked out and the only thing you can do is power up again. You just sit there wondering what happened. Then studying the hex dump closely it suddenly strikes you, part of the hex dump resides in the same place as your monitor and you have started to over-write part of the monitor you were using to enter the program in memory, hence the crash. This poses a problem, you need the monitor to dump the program to tape after typing it in but you can't use it, if the program is typed in at the location it has to run from. The answer is use a block move! The Z80 cpu has a block move instruction as part of its instruction set which can be used to advantage.

The basic idea is to type the program in at a different location from that at which it will be executed and to add a small block move routine to the beginning or end of the actual program. Then the whole program, block move and all, is saved to tape using the start of the block move routine as the entry point. Whenever the program is executed, the block move routine will move the program to its proper location and then jump to the correct entry point to execute it. To summarize then, all we need do is set up the CPU registers with the new start address, the old start address and the program length, execute a block move and then jump to the entry point of the program.

Let us look at an actual example. Assume that we have to type in the machine language program below and then save it to tape but the only free space starts at 5000H. First, the program might look like this:

```
7000 01 00 00 CD 60 00 3E 2A CD 33 00 C3 00 70
```

In assembly language it looks like this:

```
100          ORG    7000
200          LD     BC,0
300          CALL  60H
400 ENTRY    LD     A,'*'
500          CALL  33H
600          JP     7000H
700          END
```

The start is at 7000H, the entry point is at 7006H and the length of the program is 14 bytes. All this program does is print an asterix at one second intervals (couldn't think of anything better!). In order to use the block move instruction we have to load the HL register pair with the new start address, the DE register pair with the normal start address and the BC register pair with the program length, this takes nine bytes:- three for each load. The block move instruction is two bytes long and the jump is three bytes long, giving a total of 14 bytes (E hex). If we now type the above program in, starting at 5000 hex and add our block move routine to the end of it, it would look like this:

```
5000 01 00 00 CD 60 00 3E 2A CD 33 00 C3 00 70 21 00
5010 50 11 00 70 01 0E 00 ED 80 C3 06 70
```

In assembly language :

```
100      ORG      5000H
110      LD       BC,0
120      CALL    60H
130 ENTRY LD      A,'*'
140      CALL    33H
150      JP       7000H
160 ; APPENDAGE STARTS HERE !
170      LD       HL,5000H ; NEW START ADRS
180      LD       DE,7000H ; PROPER START ADRS
190      LD       BC,0EH   ; PROGRAM LENGTH = 14 BYTES
200      LDIR     ; BLOCK MOVE
210      JP       7006H   ; JUMP TO ENTRY POINT
220      END
```

In this case the block move routine has been appended to the end of the original program but it is also possible to put it at the start of the program with no change necessary.

- 00000 -

PROGRAMMERS

Wanted: Free-lance programmers familiar with TRS-80 Level II BASIC to write customized software packages for dynamic electronics retailer.

If you are a University student, tutor, lecturer, doctor, lawyer, computer programmer or microcomputer buff and if you own a TRS-80/SYSTEM 80 and/or enjoy programming in BASIC, why not take this opportunity to make some extra money in your leisure time by writing various software programmes for us? Give me a call - you have nothing to lose but plenty to gain.

Please direct enquiries to:



**MARK SIM
DICK SMITH
ELECTRONICS
Phone (02) 888 3200**

***** '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. Owner 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 between 9-4

EASTERN SUBURBS - 2

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

BUG-80 (Burwood Users Group for '80 Cassette
Systems)

Contact: Rob Callander, 288-2983 after 6 p.m.

DARWIN:
N.T.5789.

Contact: Tony Domigan, P.O. Box 39086,Winnellie,

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

- 00000 -

So you need
advice about
Computers...



DICK SMITH'S
NEW
HOTLINE SERVICE

is just
a phone call away.

Computers are fairly complex products and customers who have bought them tend to need assistance, advice and support from time to time.
A special "Hotline" service is available to Dick Smith System 80 and Sorcerer computer and peripherals customers. Up-to-the-minute help and advice is available just by phoning (02) 888 2002.
Your consultant, Steven Engel, will be pleased to assist with Dick Smith computer enquiries and will forward comprehensive literature if required.
This new "Hotline" provides a level of service and support without equal in Australia's small computer industry.

special hotline no.
(02) 888-2002

DICK SMITH ELECTRONICS

SYDNEY ● MELBOURNE ● BRISBANE ● ADELAIDE
CANBERRA ● PERTH ● WOLLONGONG



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

This month we are going to take a look at the ability of the ESF to @SAVE and @LOAD areas of memory to wafer, (any area of memory !!!), lets see you do that with a cassette.

Load in this small demonstration program and (@SAVE 1) the program before you try to run it, as the program will try to @SAVE 2, and if you have not @SAVED 1 first you are going to get nowhere.

```
10 CLS
20 FOR X=15488TO16383:POKEX,191:NEXT
25 PRINT@0,"JUST A SECOND, I'M ";
30 @SAVE2,15488,896
40 CLS
45 PRINT@0,"JUST A SECOND, I'M ";
50 @LOAD2
60 PRINT@0,CHR$(205);:PRINT@12,"THERE, IT'S ALL ";
70 REMEMBER THIS IS WHERE YOU WOULD @LOAD3 YOUR MAIN PROGRAM
80 GOTO80
```

Study the listing, there is as much to be learned from the listing as there is from running the program.

Line 20 - whites-out most of the screen, the top of the screen being kept clear for messages.

Line 30 - instructs the ESF to @SAVE an area of memory, only in this case the area we are going to save is the Video Memory, rather than a block of RAM, as you would normally do.

(25 - are a crafty way of using part of the message that the ESF puts
Lines(45 out.
(60

Line 40 - clears the screen so that when line 50 is executed we can see that Video Memory is loaded direct from the wafer.

No doubt, you can see lots of possibilities on the horizon. For instance, program instructions loaded direct to the screen, without any of your valuable RAM space being used or graphic displays loaded one after the other. I said it last month and i'm going to say it again.

ITS NOT..."JUST A FAST CASSETTE DECK " So if you haven't got one yet, just look at what you're missing!!

- 00000 -

***** SPECIAL OFFER - EXATRON STRINGY FLOPPY \$352.50 incl. p&p *****

All Exatron Stringy Floppies sold by MICRO-80 will include the special chained version of HOUSEHOLD ACCOUNTS, developed by Charlie Bartlett. When used on the ESF, this program is powerful enough to perform many of the accounting functions in a small business. Remember, the ESF comes complete with a comprehensive manual, a 2 way bus-extender cable, its own power supply and 10 wafers of mixed length. One wafer contains the Data Input/Output program and another the HOUSEHOLD ACCOUNTS program.

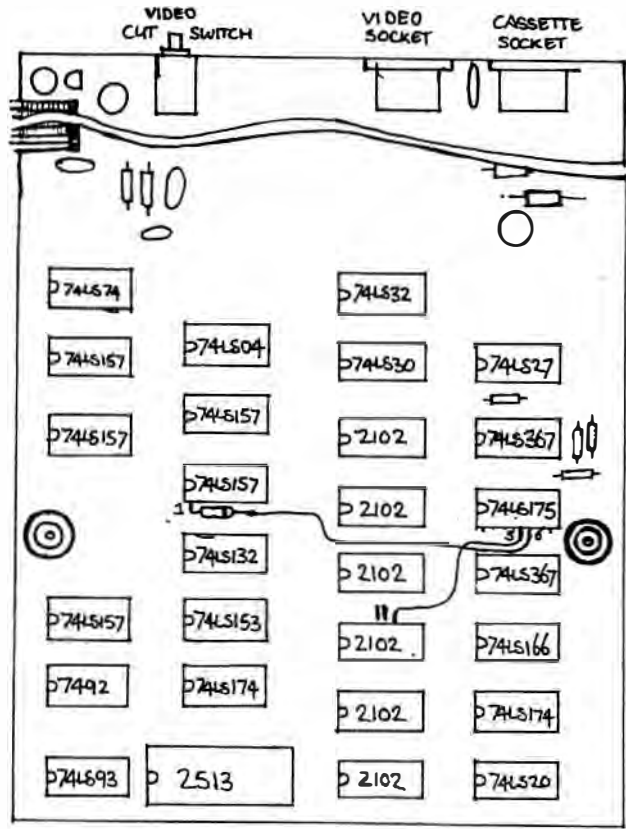
CAN'T MAKE UP YOUR MIND ABOUT THE ESF?

Then send in \$5.00 for a copy of the manual. We will refund your \$5.00 IN FULL when you purchase an ESF.

**** SYSTEM 80 - 32 CHARACTER PER LINE UNDER SOFTWARE CONTROL ****
******* CORRECTION *******

There was an error in last month's article. The lead running from pin 5 of the 74LS175 to the 2102 should have been shown going to pin 11 not pin 10. See revised diagram below.

In some machines, this modification causes the computer to go straight into large character mode after switch on. It is then necessary to type:
 OUT 255,0 <NEW LINE>
 to revert to standard character size. Next month we will describe how you can add a switch to control this function.



SYSTEM 80 LAYOUT

******* BOOT *******
 by E. R. Paay

When a disk system is first turned on, the Disk Operating System automatically loads into memory and initializes itself (commonly known as booting DOS). Some of our readers may have wondered how this happens. As a matter of fact, the routine for loading the initial DOS system file named BOOT/SYS is contained in the level II ROM. This routine is located from 069FH to 06CBH. From 0002H, the interpreter jumps to 0674H, the start of the initialization routine. At 0696H, a test is made to see if there is a disk drive. If not, initialization continues at 75H. If the floppy disk controller (FDC) is present, initialization continues at 069FH. To make clear what happens at 069FH I have prepared a disassembled listing to which I have added comments, this is shown below.

BOOT/SYS is a small program 255 bytes long (and not just a table of data as some people seem to think), which loads the system file called SYS0/SYS. In order to understand the process fully, knowledge of digital electronics and of devices such as the FDC is required, I have included sufficient comments however to understand how the program itself works.

```

0696 3AEC37 LD A,(37ECH) ;FLOPPY DISK CONTROLLER
0699 3C INC A ;PRESENT ?
069A FE02 CP 02H ;
069C DA7500 JP C,75H ;IF NOT, INITIALIZE LEV II
069F 3E01 LD A,01H ;
06A1 32E137 LD (37E1H),A ;TURN DRIVE 0 ON.
06A4 21EC37 LD HL,37ECH ;POINT HL TO FDC STATUS
;ADDRESS (RD) OR COMMAND
;REGISTER ADDRESS (WR)
06A7 11EF37 LD DE,37EFH ;POINT DE TO ADDRESS OF FDC
;DATA REGISTER.
06AA 3603 LD (HL),03H ;LOAD COMMAND REG. IN FDC
;WITH CODE FOR 40 MSEC
;HEAD STEPPING RATE.
06AC 010000 LD BC,0 ;
06AF CD6000 CALL 60H ;DELAY ONE SECOND TO ALLOW
;DRIVE MOTOR TO SETTLE DOWN
06B2 CB46 BIT 0,(HL) ;WAIT TILL DRIVE SIGNALS
;THAT IT IS "NOT BUSY"
06B6 AF XOR A ;CLEAR A REG.
06B7 32EE37 LD (37EEH),A ;SET SECTOR REGISTER IN FDC
;TO ZERO. (TRACK 0,
;SECTOR 0) THIS IS WHERE
;BOOT/SYS RESIDES ON DISK
;ADDRESS WHERE BOOT/SYS
;WILL BE LOADED.
06BA 010042 LD BC,4200H ;
06BD 3E8C LD A,8CH ;
06BF 77 LD (HL),A ;LOAD COMMAND REG. (8CH
;TELLS THE FDC TO READ A
;SINGLE RECORD, USING IBM
;FORMAT, TO LOAD HEAD ON
;DISK AND DELAY 10 MSEC.
06C0 CB4E BIT 1,(HL) ;
06C2 28FC JR Z,6C0H ;WAIT TILL DATA READY
;SIGNAL PRESENT.
06C4 1A LD A,(DE) ;GET THE BYTE READ FROM
;DISK BY THE FDC AND PLACE
;IT IN REG A.
06C5 02 LD (BC),A ;NOW PUT IT IN MEMORY.
06C6 0C INC C ;INCREMENT MEMORY POINTER
06C7 20F7 JR NZ,6C0H ;LOOP TILL 255 BYTES READ
06C9 C30042 JP 4200H ;JUMP TO BOOT/SYS PROGRAM

```

- 00000 -

+++++++ HOW TO GO BEYOND 16K WITHOUT GOING BANKRUPT ++++++

A Review of the LNW RESEARCH Expansion Interface Kit

--- By Brian H. CHRISTENSEN P.O. Box 140, WODEN A.C.T. 2606 ---

After twelve months as a happy user of a 16K Level II system I recently decided to make the great leap forward to a disk system. The shock of TANDY's price tag of \$919 for a 32K Expansion Interface, needed before I could even think about a disk drive, sent me searching through overseas magazines in search of an alternative. I eventually found an advertisement offering a bare pc board for an expansion interface, with a construction manual, for the paltry sum of US\$69.95. The offer was made by LNW Research, of P.O. Box 16216, Irvine, California, 92713, U.S.A. to whom I promptly wrote for further details.

The first surprise was the time taken for a reply to be received - a little over a week. The reply gave further details of the board including the fact that it has provision for an on-board RS232C interface - a considerable bonus since TANDY charge \$139.95 for this feature. The cost of shipping by airmail to Australia was quoted as US\$10.05, making the total cost \$90.00. Encouraged thus far, I decided to take the plunge and promptly sent off a bank draft. In less than another two weeks I had in my hand a well-made double-sided 292mm x 245mm board and a 70 page manual. I then commenced to acquire the 66 integrated circuit chips and other components necessary to convert the board into an expansion interface. To make this easier, the manual provides separate parts lists for (a) power supplies, (b) memory expansion, (c) floppy disk controller, (d) printer interface, (e) serial interface controller (RS232C), and (f) real time clock. This enables the constructor to build only those features he requires. The cost of components will vary with the source of supply, but the following is a fair guide to what I paid for each section of the interface:-

Power Supplies	\$30 plus transformer
Disk Controller	\$78
Memory Expansion	\$26 plus memory chips
Printer Interface	\$16 (if the disk controller is built this drops to about \$4)
Serial Interface	\$41
Dual Cassette Interface	\$2 plus relay and hardware

Note that 200 ns chips (4116 or equivalent) are necessary for the memory expansion. Slower chips will not work in the interface. If you already have some slower 4116 chips, try putting them in the CPU and using the TANDY chips in the interface. This worked for me. For the power supply

transformer, LNW Research suggest that you use the standard TANDY TRS 80 power supply. However, since I understand TANDY (Australia) are asking about \$50 for this item, I chose to supply the two voltages required (17 VAC at 1A and 20VDC rectified but not smoothed at 350 ma) using two Ferguson low profile transformers at about \$10 each (types PL15/40VA and PL40/40VA) and a couple of EM402 diodes.

So much for the costs. How easy is it to get the interface built and working? Although I have held a ham licence since 1963 and have put together a few small ic projects (eg. digital clocks), I do not consider myself an expert with a soldering iron and I have certainly never tackled anything as complex as the interface before. I made sure I had a light, controlled-temperature soldering iron, a supply of good quality fine solder and a well-lit place to work. The actual construction of the board took me just under six hours. I think most people could do it in four. The board is clearly marked with the position of all components and anything unusual is explained in the manual. Although sockets are not necessary for the chips, LNW Research recommend their use and I commend their advice. The only test equipment specified is a voltmeter to ensure you have the right voltages in the right places, before you plug in the chips. When I had finished I plugged in the 16K of memory chips I had on hand and waited for results. Happily no smoke issued forth but the system went into a wait state (or a loop?) while trying to establish memory size. Swapping the memory chips as described above, I was rewarded with a proper memory size. However, my joy was shortlived as I soon discovered that BASIC programs which use high memory for string storage would just not work. A long and careful inspection of the 1000-odd solder points looking for dry joints proved fruitless. Eventually I traced the problem to the 40-connector cable joining the interface to the CPU. In my desire to save money, I had made this up myself instead of buying it ready-made, and a slow tedious job it was. Despite my care, one of the connections had broken, apparently in an address line. When this was fixed the problem disappeared. The next test was the parallel printer interface. As soon as I plugged in the ANADIX and typed in an LPRINT command, the printer sprang into life indicating that I had no problems in that area. The crucial test, I felt, would be the disk controller. Not owning a disk drive, I had to wait a few days before I could borrow one. With fear and trepidation, I plugged in the drive, powered up and was greeted by the magic words "DOS READY". So far I have not been able to test the RS232C serial interface but I have no reason to believe that it will cause any more trouble than the other sections. The dual cassette

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

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 80. Use 35, 40, 77 or 80 track 5" mini disk drives or 8" disk drives, OR ANY COMBINATION!
- * A 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.
- * CDE function; simultaneous striking of the C, D and E 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 like 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.

** 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

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.

***** 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.

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

We have changed our brand of diskettes but not our price. We now sell genuine

BAS F DISKETTES

SINGLE SIDE/SINGLE DENSITY \$45.00 per box of 10 incl. p&p.

DOUBLE SIDE/DOUBLE DENSITY \$59.00 per box of 10 incl. p&p.

***** 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). Tapes are specially recorded at high levels to ensure satisfactory loading into SYSTEM 80's.

** 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 will provide 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.
- * S-KEY allows text, basic commands and/or graphics to be defined to shifted keys. This will make 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 graphic symbols directly into PRINT statements.
- * S-KEY will allow 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 entry's 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 50 c. p&p
THE ULTIMATE HIGH MEMORY BASIC MONITOR L2/16K and up.

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 afford NOT to have BMON?

** EDUCATIONAL **

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

Give the 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 NOW 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 everyday or occasional use, 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 reach you to type rapidly and accurately, and a massive cassette data dump to control your progress. The computer checks for accuracy, and sets times 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 **

UBOAT \$7.50 plus 50c. p&p.

Real time simulation at its best! Comes with working sonar-screen and periscope, a full rack of torpedos, plenty of targets, working fuel and battery meters, helpful Mothership for high-sea provisioning, 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 each 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) contains 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.

***** GREEN SCREEN SIMULATOR *****
\$19.95 incl p&p

The GREEN SCREEN SIMULATOR is made from a deep green perspex, cut and curved 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 a new (rounded) style monitor when ordering.

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

Use ORDER FORM on page 45

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

MPI is the second largest manufacturer of disk drives in the world. They use the same form of head control as on 8" drives and consequently, they have the fastest track-to-track access time available: 5 msec. All their drives are capable of single or double density operation. (Double-density operation requires the installation of a special PC board in the expansion interfacs. 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. To all intents and purposes, a dual-head drive behaves like two single-head drives but is much cheaper.

Our MPI drives are supplied bare, set up to operate with the TRS-80. They can be used bare or a simple cabinet made up from sheet metal. 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
Ring first to check availability.

***** BOOKS *****

LEVEL II ROM REFERENCE MANUAL \$24.95 plus \$1.20 p&p.

Over 70 pages packed full of useful information and sample programs. Applies to both TRS-80 and SYSTEM 80.

TRS-80 DISK AND OTHER MYSTERIES \$24.95 plus \$1.20 p&p.

The hottest selling TRS-80 book in the 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.

LEARNING LEVEL II \$16.95 plus \$1.20 p&p.

The second hottest selling TRS-80/ SYSTEM 80 book in the U.S.A. Written by David Lien, the author of the TRS-80 Level I handbook, this book teaches you, step-by-step, how to get the most from your Level II machine.

INTRODUCTION TO T-BUG \$12.50 plus \$1.20 p&p.

This book describes in detail how to use Tandy's T-BUG monitor program. Each command is explained and discussed in detail with examples. A must for the T-BUG user.

**** 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)
 \$7.95 + 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) \$7.95 + 50c. p&p
 4 Space games including ULTRA-TREK, ROMULAN, STARWARS and STARLANES. Fast, real-time graphics.

STRATEGY GAMES (L2/16K) \$7.95 + 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) \$7.95 + 50c. p&p
 A set of 6 utility programs which allow you to draw BAR GRAPHS, GRAPH CARTESIAN COORDINATES, carry out POLAR GRAPHING, PARAMETRIC GRAPHING, LINEAR REGRESSION and PARABOLIC REGRESSION.

**** PROGRAMS BY MICROSOFT ****

EDITOR ASSEMBLER PLUS (L2/16K)
 \$39.95 plus \$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 September. 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.

******* 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.

interface merely provides a signal to drive a changeover relay. There is no provision for the relay on the board itself. If you don't want dual cassettes you can use the relay to sound a buzzer, turn a coffee pot on and off, or whatever.

The construction manual does not give any suggestion as to how the interface should be housed. I chose a metal box 419 x 280 x 134 mm (Adaptive Electronics model M502) which is the same length as the keyboard and can fit under the video monitor. The height is a bit excessive but leaves room for other boards which I intend to add. There is room at one end of the box for the power supply transformers, as shown in Fig.1. Since I wanted to place the disk drives and printer to the right of the monitor, I chose to connect the interface to the CPU using the scheme shown in Fig. 1. There is a limit of 300mm on the length of the connecting cable. If you want to use flat connecting cable, it would be possible to turn the board over and use the configuration in Fig. 2.

So, there it is. I am extremely happy with my new expansion interface. Its construction should present no difficulties to anyone who has had at least some experience in integrated circuit projects. Provided you tackle the project with care and the proper equipment (a 200 watt iron is NOT recommended) you should have the same success as I had, and save the price of your first disk drive into the bargain. I would be happy to answer any queries on the project, but a stamped addressed envelope would be appreciated.

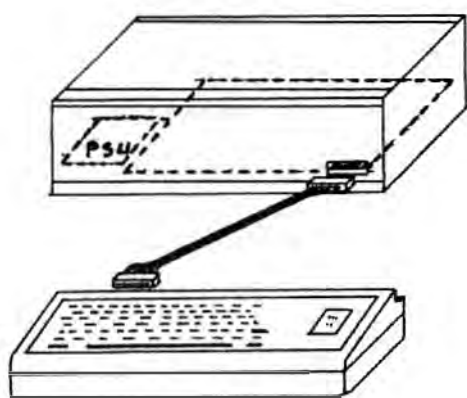


FIGURE 1

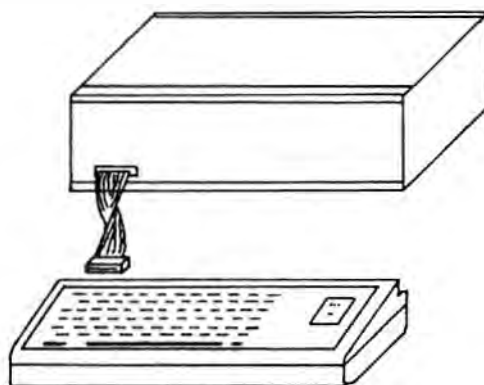


FIGURE 2

```
10 REM."ESCAPEE" COPYRIGHT A.W.HOLLAND 23SEP79
20 CLS:Y=1:N=0:H=0
30 P.T.(24),"*** ESCAPEE ***
40 I."DO YOU REQUIRE INSTRUCTIONS";T:IFT=0G.320
50 IFT<>1G.40
60 P.A.64:P.A.64,"WHEN INSTRUCTIONS APPEAR HOLD DOWN [
70 P."KEY UNTIL YOU ARE READY TO GO ON.":F.K=1T05000:N.K
80 P.A.64:P.
90 P.A.64,"YOU ARE A POLICEMAN CHASING AN ESCAPED PRISONER.
100 P."YOU HAVE CORNERED HIM IN AN UNDERGROUND GARAGE."
110 P."THE GARAGE HAS 64 PILLARS (SEE PLAN BELOW)":GOS.700:GOS.770
120 P.A.64:P.:P.
130 P.A.64,"THE ESCAPEE IS HIDING BEHIND 1 OF THE PILLARS.
140 P."IF YOU CAN GUESS WHICH PILLAR HE IS ";
150 P."BEHIND, YOU WILL CATCH HIM":GOS.770:P.A.64:P.
160 P.A.64,"EACH PILLAR IS NUMBERED -
170 P."EXAMPLE - 6,2 MEANS THE PILLAR IN COLUMN 6 ROW 2 (SEE BELOW)
180 F.K=1T010:P.AT558,"*";AT565,".";A.572,"":F.P=1T0500:N.P
190 P.A.558,".";A.565,".";A.572,"":F.P=1T0100:N.P:N.K
200 P.A.64:P.
210 P.A.64,"AFTER EACH GUESS I WILL TELL YOU EITHER HOW MANY
220 P."COLUMNS OR ROWS AWAY THE ESCAPEE IS FROM YOUR GUESS.
230 GOS.770:P.A.64:P.:P.:P.
240 P.A.64,"THERE IS 1 CATCH -
250 P."AFTER EACH GUESS THE ESCAPEE MOVES.
260 P."HE MOVES 1 PILLAR HORIZONTALLY OR VERTICALLY ON THE PLAN.
270 GOS.770:P.A.64:P.:P.
310 G.330
320 GOS.700
330 A=RND(8):B=RND(8)
340 P.A.64,"GUESS WHICH PILLAR THE ESCAPEE IS HIDING BEHIND (COL,ROW)";
350 I.R,S:IF(R<1)+(R>8)+(S<1)+(S>8)G.340
355 H=H+1:CLS
360 IF(A=R)*(B=S)G.600
370 C=A-R:IFC<0C=-C
380 D=B-S:IFD<0D=-D
385 IFC=0G.430
386 IFD=0G.400
390 F=RND(2):ONFG.400,430
400 IFC=1P.A.128,"ESCAPEE IS 1 COLUMN FROM YOUR LAST GUESS.
410 IFC=1G.460
420 P.A.128,"ESCAPEE IS";C;"COLUMNS FROM YOUR LAST GUESS.":G.460
430 IFD=1P.A.128,"ESCAPEE IS 1 ROW FROM YOUR LAST GUESS.
440 IFD=1G.460
450 P.A.128,"ESCAPEE IS";D;"ROWS FROM YOUR LAST GUESS.
460 P.A.24,"*** ESCAPEE ***":GOS.700
465 P.A.452+R*7+(S-1)*64,"*";
470 E=RND(4):ONEG.400,510,540,570
480 A=A+1:IFA=9A=A-1:G.470
500 G.340
510 A=A-1:IFA=0A=A+1:G.470
530 G.340
540 B=B+1:IFB=9B=B-1:G.470
560 G.340
570 B=B-1:IF B=0 B=B+1:G.470
590 G.340
600 P."YOU CAUGHT HIM !!! CONGRATULATIONS
610 P."IT TOOK YOU";H;"GUESSES
```

```

620 I. "DO YOU WANT ANOTHER GAME";T:IFT=16.10
630 IFT<>0G.620
640 CLS:P. "OKAY SEE YOU LATER. ":GOS.770:CLS
650 E.
700 P.A.348, "COLUMNS";A.641, "ROWS
720 N=1:F.M=394T0443S.7:P.A.M,N:N=N+1:N.M
730 N=1:F.M=454T0902S.64:P.A.M,N:N=N+1:N.M
740 M=459:N=508:F.K=1T08:F.L=MTONS.7:P.A.L, ".":N.L
750 M=M+64:N=N+64:N.K
760 N=0:RET.
770 F.K=1T02000:N.K:RET.

```

```

1 REM ***** THE WORLD - LEVEL I *****
2 REM ** WRITTEN BY R. A. GEHRIG
3 REM ** 9 CAROLINE ST KINGSGROVE
4 REM ** COPYRIGHT MICRO-80 **
10 C.
60 REST.
100 P.AT91, "THE WORLD";
120 F.I=1T0145:READL,R,Y
130 F.B=LTOR:R.(B,Y):S.(B,Y):N.B
140 N.I
150 F.I=1T010000:N.I:G.10
1000 D.4,12,9,38,51,9,2,19,10,41,50,10,7,28,11,40,49,11,84,89,11,8,25
1010 D.12,39,44,12,79,104,12,8,20,13,39,40,13,62,69,13,77,114,13,7,22
1020 D.14,26,31,14,59,62,14,64,115,14,6,32,15,54,54,15,58,62,15,66,108
1030 D.15,113,113,15,5,27,16,51,51,16,53,54,16,59,59,16,64,106,16,113
1040 D.114,16,3,28,17,57,109,17,3,24,18,54,110,18,4,21,19,51,55,19,59
1050 D.59,19,62,66,19,71,74,19,77,78,19,80,110,19,112,112,19,3,18,20,51
1060 D.54,20,61,61,20,63,74,20,77,108,20,113,113,20,3,3,21,5,10,21,16
1070 D.17,21,66,75,21,78,108,21,107,108,21,112,113,21,3,3,22,5,8,22,16
1080 D.17,22,50,58,22,70,104,22,111,114,22,5,8,23,48,75,23,77,107,23,6
1090 D.13,24,18,19,24,21,21,24,46,68,24,71,75,24,78,107,24,10
1100 D.13,25,46,69,25,71,79,25,84,106,25,12,13,26,46,70,26,73,79,26,87
1110 D.92,26,96,105,26,14,24,27,46,72,27,74,76,27,87,90,27,97,102,27
1120 D.109,109,27,17,27,28,47,76,28,87,90,28,99,103,28,109,109,28,16,28
1130 D.29,56,76,29,88,90,29,99,99,29,101,102,29,15,29,30,57,74,30,99
1140 D.100,30,110,112,30,15,34,31,57,72,31,100,100,31,106,107,31,16,35
1150 D.32,57,70,32,104,107,32,17,33,33,59,70,33,115,118,33,19,33,34,58
1160 D.70,34,117,120,34,20,33,35,58,70,35,74,75,35,120,121,35,20,31,36
1170 D.59,67,36,73,74,36,113,115,36,118,119,36,20,29,37,59,67,37,73,73
1180 D.37,109,115,37,117,119,37,21,28,38,60,65,38,104,119,38,21,25,39
1190 D.60,63,39,104,119,39,21,27,40,103,120,40,21,24,41,103,120,41,21
1200 D.24,42,102,105,42,110,118,42,22,24,43,112,116,43,23,26,44,125,125
1210 D.44,112,113,45,124,126,45,118,123,46

```

```

1 ***** CUP '80 - LEVEL I *****
BY: SPENCER GEORGE.

```

```

5 F.R=1T050:X=R.(100):N.R
10 F.X=1T016:A(X)=0:N.X
30 C.
31 F.J=1T016
32 P.A.(J-1)*64,J;
33 N.J
40 X=R.(16)
50 A(X)=A(X)+1
60 IFA(X)=60P.A.(X-1)*64+10, " NUMBER ";X;" IS THE WINNER ";;G.200
70 T=A(X)+(X-1)*64+3
110 P.A.T,"*";
120 G.40
200 G.200

```

```

1 '      ***** CUP '80 - LEVEL II *****
        BY:  SPENCER GEORGE,
            14/47 YERRIN STREET,
            BALWYN, VIC. 3103
            (03) 836 4225

```

```

5 RANDOM
10 CLEAR 2000
20 DIM L(16)
30 CLS
31 FOR J = 1 TO 16
32 PRINT@ (J-1) * 64, J;
33 NEXT J
40 X = RND(16)
50 L(X) = L(X) + 1
60 IF L(X) = 60 PRINT@ (X-1)*64 + 10, " NUMBER " ; X ; " IS THE WINNER " ;: G
OTO 200
70 T = L(X) + 15360 + (X-1) * 64 + 3
110 POKE T, 42
120 GOTO 40
200 GOTO 200

```

**** LOTTO PREDICTOR**** 32K AND DISK (options for cassette) by Peter G. Hartley

This is the first international edition of MICRO-80 (you'll probably find the Editor in the middle of Leicester Square shouting "MICRO-80 guv? Only a quid?") so I'd better explain to my fellow countrymen back "home" just what LOTTO is - because to them it's probably still an old Naval name for Housie-housie or BINGO.

Every Englishman knows that Australians will bet on anything - even two flies crawling up a window. In South Australia and Victoria each week, under the myopic gaze of the Television Audience, a massive perspex barrel spins, selecting six or seven numbered ping-pong balls (which the Aussies insist on calling marbles). We all grab our copy coupons and cross our fingers.

The object is to have marked the first six out of the barrel - in which remote instance the lucky coupon holder will "shout all his mates" (purchase a glass of beer for each of his friends present in the local licensed premises), retire from gainful employment and purchase a lottery ticket or three to celebrate.

Now, the odds against winning are fairly remote - though not as remote as getting 8 score-draws out of 8 on the "footy-pools". Those of you with a mathematical bent may care to work out the odds using the following formula.

$$O = (T * T-1 * T-2 * \dots * T-(N-1))$$

$$(1 * 2 * \dots * N)$$

where T = TOTAL number of possible numbers

N = NUMBER of numbers required

O = a snowflake in hell's chance to ONE

For forty numbers and picking six - it's about one in 3.8 million!

Now, no matter how well run such a "lottery" may be, there's always some idiot around who goes looking for a pattern. I'm just such an idiot, and the scheme used in this program actually worked with some small degree of success in the early days of the South Australian Cross Lotto, when there were only 36 numbers instead of the present 40.

Assumption number one - patterns tend to repeat

Assumption number two - every number has an equal chance each week, regardless of previous events

Assumption number three - eventually it will all even out, and in the year 3000 or so every number will have been drawn an equal number of times - and every pattern will equal every other pattern.

The pattern that I sought was that of numbers being drawn in any one draw being prediction pointers for numbers drawn the following week. Having built up a data base of these predictions, we then take care of the second and third "assumptions" by mixing the most predicted numbers with those least predicted.

The structure of the data base is a 40 by 40 array (L). (L for logical? Ed.) The size is determined by the number of possible numbers. If there are only 36 numbers, it would be 36 by 36. In L(X,Y) X is one of the numbers drawn last week, and Y is any of the possible numbers. The data stored there represents the number of occurrences when Y has been drawn in the week following the drawing of X.

With six numbers drawn there are 6*6 pieces of data to file away each week. Each of the previous week's numbers are assumed to predict each of the most recent draw's numbers.

Once the data file has been opened, and the data loaded, the results of the previous "most recent draw" are read into W(1) through W(6), in line 130.

Lines 150 through 270 allow the operator to input the results of the next draw into array X(1) through X(6)

It is now a simple matter to increment the appropriate parts of the data base with the 36 new predictions. This occurs in lines 270 to 310.

Next we load array U with the levels of prediction recorded for the six most recently drawn numbers - still in array X (Lines 330 to 380) and these are bubble-sorted into their order of merit (lines 390 to 450).

The remainder of the program simply mixes your selections of most and least predicted values and generates random mixes of these for your coupon entries.

The now updated data is filed back to disk, and the contents of array X are written in for reading back as array W in one week's time.

Because of the size of the data-base, this program doesn't really lend itself to use with cassette based systems, but the REMs in the listing explain how the more determined of you can do it anyway....

The error-trap in lines 50 and 570 is only used with disk systems for the very first pass through the program - when there is no file on disk to read.

Line 320 is only required to allow the building up of a data base from past records, and should be deleted when you start running the program on a week by week basis. Once your data-base has been established, when the loop established by line 320 asks for more data, press <BREAK>, type in GOTO 690 <ENTER> and the data base will be filed away....

Readers may care to develop this further - perhaps by adding extra dimensions to the arrays so as to allow predictions based of the previous week but one, the previous week but two, etc.

Over twenty years ago I forked out forty pounds (a great deal of money in those distant days) for a vaguely similar system for use on the football pools. As it actually worked, I look forward to being able to present the BASIC version of that in a future edition of MICRO-80.

Happy Dreaming!

P.G.H.

```

1 '   COPYRIGHT (C) 1980 PETER G. HARTLEY.
   57 MAIN AVENUE, FREWVILLE, S.A. 5063.
   TELEPHONE 08 - 79 4061.

2 '
3 '   MICRO-80, SEPTEMBER 1980
4 '
10 DEFINTA-Z
20 CLS
30 PRINT"LOTTO PREDICTION SYSTEM"
40 DIML(40,40),V(1,40),Z(40) 'VALUES DETERMINED BY NUMBER OF MARBLES
50 ONERRORGOTO790 'THIS LINE ONLY REQUIRED ON FIRST PASS
   THROUGH WITH NO DATA FILE ON DISK
60 OPEN"1",1,"LOTTO/TXT" 'CASSETTE USERS CHANGE TO
   60 PRINT"READY CASSETTE?"
   AND DELETE LINE 50
70 ONERRORGOTO0 'SEE REM LINE 50
80 FORK1=1T040 'THIS SEQUENCE IS INEFFICIENT WITH CASSETTE
90 FORK2=1T040 'SYSTEMS, FOR WHICH DELETE LINE 90
   AND CHANGE 100 TO...
100 INPUT#1,L(K1,K2) '100 INPUT#-1,L(K1,1),L(K1,2),L(K1,3),
   L(K1,4),L(K1,5)...L(K1,40)
110 NEXT:NEXT
120 FORK=1T06 'NUMBER DRAWN WEEKLY
130 INPUT#1,W(K) 'LAST WEEK'S NUMBERS
   FOR CASSETTE CHANGE TO
   130 INPUT#-1,W(1),W(2),W(3),... to W(6)
   AND DELETE LINES 120, 140 AND 150
140 NEXT
150 CLOSE
160 CLS
170 FORK=1T06 'NUMBER DRAWN WEEKLY
180 IFK=1GOTO230
190 CLS
200 FORL=1TOK-1
210 PRINT"#*L"... *X(L)
220 NEXT
230 PRINT"MARBLE #*K;:INPUT"NUMBER WAS ";X(K)
240 NEXT
250 CLS
260 PRINT"UPDATING DATA"
270 FORK1=1T06 'NUMBER DRAWN WEEKLY
280 FORK2=1T06 'NUMBER DRAWN WEEKLY.
290 L(W(K1),X(K2))=L(W(K1),X(K2))+1
300 NEXT
310 NEXT

```



```

320 FORK3=1T06:W(K3)=X(K3):NEXT:GOTO160 'USE THIS LINE ONLY WHILE BUILDING
UP YOUR DATA BASE
330 FORKK=1T040:U(0,KK)=KK:NEXT
340 FORK1=1T06
350 FORK2=1T040
360 U(1,K2)=U(1,K2)+L(X(K1),K2)
370 NEXT
380 NEXT
390 FORK1=1T039          ' INEFFICIENT OLD BUBBLE-SORT STARTS HERE
400 P=K1
410 FORK2=K1+1T040
420 IFU(1,K2)>U(1,P)THENP=K2
430 NEXT
440 IFF<>K1:THENUU=U(0,P):
      U(0,P)=U(0,K1):
      U(0,K1)=UU:
      UU=U(1,P):
      U(1,P)=U(1,K1):
      U(1,K1)=UU
450 NEXT
460 INPUT"HOW MANY (A) MOST PREDICTED
AND
HOW MANY (B) LEAST PREDICTED   A , B  ";A,B
470 T=A+B
480 CLS
490 PRINT"MOST PREDICTED GROUP..."
500 FORK1=1T0A:PRINTU(0,K1),:S=S+1:Z(S)=U(0,K1):NEXT
510 PRINT
520 PRINT"LEAST PREDICTED GROUP..."
530 FORK2=40T041-BSTEP-1:PRINTU(0,K2),:S=S+1:Z(S)=U(0,K2):NEXT
540 PRINT
550 INPUT"HOW MANY NUMBERS PER ENTRY?";TN
560 IFTN>T GOTOSS0
570 INPUT"HOW MANY ENTRIES";TE
580 CLS
590 PRINT"STAND BY..."
600 DIMR(TN)
610 FORK1=1TOTE
620 FORK2=1TOTN
630 R(K2)=RND(T)
640 IFK2<>1
      FORK3=1TOK2-1:
          IFR(K3)=R(K2)GOTO630
650     NEXT
660 NEXT
670 FORK2=1TOTN:PRINTZ(R(K2));:NEXT:PRINT" ",
680 NEXT
690 OPEN"0",1,"LOTTO/TXT":'FOR CASSETTE USE CHANGE TO
      690 INPUT"READY CASSETTE ";D$
700 FORK1=1T040
710 FORK2=1T040:'      SEE REM ON LINE 750
720 PRINT#1,L(K1,K2)
730 NEXT
740 NEXT
750 FORK1=1T06          'CASSETTE USERS ARE DIRECTED TO STUDY
760 PRINT#1,X(K1)      'REM FOR LINES 120 THRU 140
770 NEXT
780 PRINT"RUN ENDED"
790 RESUME690          ' ERROR TRAP
                        THIS CREATES A DATA FILE ON YOUR FIRST
                        PASS THROUGH THE PROGRAM

```

TRIANGLE

KEITH NEIGHBOUR

Given any combination of three known values of the sides and angles of any triangle, the other three values are determined by this program. (The case of three given angles does not provide a valid solution, of course.)

A triangle is displayed on the screen purely as a diagram to prompt the correct inputting of the given values, and also to identify the answers with the remaining appropriate sides and/or angles.

Input prompting is clockwise on the diagram and the computer will ask for each of the six possible values in turn, identifying the position on the triangle by "???" with each request.

When the request is for the value of a side or an angle which is not known (one of the required answers) hit ENTER to receive the next prompt.

It does not matter if you allot an obtuse angle (greater than 90°) to a position shown on the diagram as an acute angle (less than 90°), or a short side to a diagrammatic long side, or vice versa.

The correct relative positions of the input data is the only important factor.

You cannot input more than three lots of data, as the execution program will automatically take off after the third input. However if you input less than three values in a complete cycle of prompts you will be told to start again.

Whilst the display triangle is diagrammatic (i.e. not to angular or dimensional scale for a particular case), it will automatically be re-drawn if necessary to conform to a correct short side to long side relationship (left and right sides). There is a reason for this apart from making it look a little more correct, which is explained later.

There are five possible combinations of three given values to solve any triangle. These are:

- (1) Three sides,
- (2) Two Angles and the included Side,
- (3) One Side and the following (clockwise) two Angles,
- (4) Two Sides and the included Angle,
- (5) Two Sides and an adjacent Angle.

Combination (5) is a special case and is the main reason for re-drawing the long and short sides where appropriate. For this particular combination there are two possible solutions in some cases. You needn't worry about it, as the computer will recognize these special cases when they occur, and will answer and draw both solutions.

Also for combination 5 the given input angle must be allotted to either the left or right base angle (not the apex), and the two given sides must be allocated to their correct positions relative to the base angle used. In other words the base in the diagram cannot be regarded as one of the two given sides.

However for combinations (1), (2), (3) or (4) there is no restriction. That is, there are three configurations for fitting any one given set of these combinations on the diagrammatic triangle (rotating the same relative positions around the triangle) and it does not matter which one you use.

When the answers are displayed against the diagram, all angles are identified by an asterisk (*) before their values. All angles are input and answered in Degrees (not radians)

(Calculations are Single Precision or Double Precision selectable.)

```

10 ' * TRIANGLE *
    BY KEITH NEIGHBOUR
20 CLS: CLEAR 200:F1=0:F3=0:F4=1:DEFINT C-H,J,Z
30 P1$="101010":Q1$="110001":Q2$="011100":Q3$="000111"
40 R1$="110100":R2$="001101":R3$="010011"
50 T1$="111000":T2$="001110":T3$="100011"
60 U1$="101100":U4$="101001"
65 M0$="(FOR DEG,MINS,SECS HIT 'M' (USE FORMAT D/MM/SS) ELSE HIT ENTER)"
70 R1=57.29578:PI=3.141593:B2$=STRING$(7,128)
80 GOSUB 230 :GOSUB 800
90 PRINT@0,"ENTER DATA (3 ONLY) CLOCKWISE AS PER ??. FOR NO DATA HIT ENTER";
100 A1=0:A2=0:A3=0:P=0:Q=0:R=0:D=0:B$=STRING$(63,128):B1$=STRING$(10,128)
102 PRINT@896,"FOR DOUBLE PRECISION HIT 'D' ELSE HIT ANY OTHER KEY";:C$=""
103 C$=INKEY$:IF C$=""GOTO 103
104 IF C$="D" THEN F5=1:P$="( D.P.)":DEFDBL A,K,L,M,P,Q,R,S,T,X:R1=57.2957795130
8232:PI=3.14159265358979323:X9=PI/2:GOTO 106
105 P$="( S.P.)":DEFSNG A,K,L,M,P,Q,R,S,T,X
106 PRINT@896,B$;:PRINT@184,P$;:PRINT@896,M0$;
107 M5$=INKEY$:IF M5$=""GOTO 107
110 S1=0:PRINT@463,"??";:PRINT@896,B$;:PRINT@896,"SIDE S1";:INPUT S1:IF S1=0 THE
N A$="0":GOTO 120 ELSE PRINT@460,B2$;:PRINT@460,CSNG(S1);:A$="1":D=D+1
120 A0$="":PRINT@294,"??";:PRINT@896,B$;:PRINT@896,"ANGLE A1";:INPUT A0$:IF A0$=
"" THEN A1=0:A$=A$+"0":GOTO 130 ELSE PRINT@291,B2$;:GOSUB 5000:A1=A0:PRINT@291,
CSNG(A1);:A$=A$+"1":D=D+1
130 S2=0:PRINT@503,"??";:PRINT@896,B$;:PRINT@960,B$;:PRINT@896,"SIDE S2";:INPUT
S2:IF S2=0 THEN A$=A$+"0":GOTO 150 ELSE PRINT@500,B2$;:PRINT@500,CSNG(S2);:A$=A
$+"1":D=D+1
140 IF D=3 THEN A$=A$+"000":GOTO 250
150 A0$="":PRINT@819,"??";:PRINT@896,B$;:PRINT@896,"ANGLE A2";:INPUT A0$:IF A0$=
"" THEN A2=0:A$=A$+"0":GOTO 170 ELSE PRINT@816,B2$;:GOSUB 5000:A2=A0:PRINT@816
,CSNG(A2);:A$=A$+"1":D=D+1
160 IF D=3 THEN A$=A$+"00":GOTO 250
170 S3=0:PRINT@797,"??";:PRINT@896,B$;:PRINT@896,"SIDE S3";:INPUT S3:IF S3=0 THE
N A$=A$+"0":GOTO 190 ELSE PRINT@794,B2$;:PRINT@794,CSNG(S3);:A$=A$+"1":D=D+1
180 IF D=3 THEN A$=A$+"0":GOTO 250
190 A3=0:PRINT@780,"??";:PRINT@896,B$;:PRINT@896,"ANGLE A3";:INPUT A3:IF A0$=""
THEN A3=0:A$=A$+"0":GOTO 200 ELSE PRINT@777,B2$;:GOSUB 5000:A3=A0:PRINT@777,CSN
G(A3);:A$=A$+"1":D=D+1
200 IF D<3 THEN PRINT@896,B$;:PRINT@896,"NOT ENOUGH DATA. HIT ENTER TO START AGA
IN";:GOTO 220
210 GOTO 250
220 INPUT Z:GOTO 20
230 J1=39:J2=41:H=3:V=65:C0=176:C1=140:C2=131:C3=139:C4=180:C5=130:C6=173:C7=144
:E1=460:E2=291:E3=500:E4=542:RETURN
240 J1=23:J2=22:H=-3:V=62:C0=131:C1=140:C2=176:C3=184:C4=135:C5=160:C6=158:C7=12
9:E1=451:E2=276:E3=492:E4=537:RETURN
250 PRINT@0,B$;:IF F5=1 THEN PRINT@896,"          C O M P U T I N G . . . . . IN DOU
BLE PRECISION";
255 A1=A1/R1:A2=A2/R1:A3=A3/R1
260 IF A$=P1$ THEN K=S1:L=S2:M=S3:GOSUB 620 :A3=P:A2=R:A1=Q:GOTO 430
270 IF A$=Q1$ THEN K=S1:P=A3:Q=A1:GOSUB 670 :A2=R:S2=L:S3=M:GOTO 430
280 IF A$=Q2$ THEN K=S2:P=A1:Q=A2:GOSUB 670 :A3=R:S3=L:S1=M:GOTO 430
290 IF A$=Q3$ THEN K=S3:P=A2:Q=A3:GOSUB 670 :A1=R:S1=L:S2=M:GOTO 430
300 IF A$=R1$ THEN K=S1:Q=A1:R=A2:GOSUB 710 :A3=P:GOSUB 680 :S2=L:S3=M:GOTO 43
0
310 IF A$=R2$ THEN K=S2:Q=A2:R=A3:GOSUB 710 :A1=P:GOSUB 680 :S3=L:S1=M:GOTO 43
0

```

```

320 IF A$=R3$ THEN K=S3:Q=A3:R=A1:GOSUB 710 :A2=P:GOSUB 680 :S1=L:S2=M:GOTO 430
330 IF A$=T1$ THEN K=S1:L=S2:Q=A1:GOSUB 730 :S3=M:GOSUB 620 :A2=R:A3=P:GOTO 430
340 IF A$=T2$ THEN K=S2:L=S3:Q=A2:GOSUB 730 :S1=M:GOSUB 620 :A1=P:A3=R:GOTO 430
350 IF A$=T3$ THEN K=S3:L=S1:Q=A3:GOSUB 730 :S2=M:GOSUB 620 :A1=R:A2=P:GOTO 430
360 IF A$=U1$ THEN K=S1:L=S2:R=A2:GOSUB 750 :A3=P:A1=Q:GOSUB 670 :S3=M:F2=0:GOTO 390
370 IF A$=U4$ THEN K=S2:L=S1:R=A3:GOSUB 750 :A2=P:A1=Q:GOSUB 670 :S3=M:F2=1:GOTO 380
380 IF L<K GOTO 430 ELSE GOTO 400
390 IF L<K GOTO 430
400 P1=PI-P:Q1=P-R:Q2=Q:Q=Q1:M2=M:GOSUB 730 :M1=M:M=M2:Q=Q2:IF F2=1 GOTO 420
410 J3=23:V1=65:H1=-3:D0=137:D1=144:D2=130:D3=164:D4=135:D5=156:E5=87:F1=1:GOTO 430
420 J3=40:V1=63:H1=3:D0=160:D1=134:D2=152:D3=129:D4=172:D5=139:E5=104:F3=1:GOTO 430
430 IF S2>S1 THEN CLS:GOSUB 240 :GOSUB 800
440 IF F3=1 GOSUB S70
450 PRINT@396,B$;:PRINT@794,B1$;
460 PRINT@E1,B2$;:PRINT@E3,B2$;:PRINT@922,B2$;:PRINT@E2,B2$;:PRINT@815,B2$;:PRINT@777,B2$;
470 PRINT@E1,CSNG(S1);:PRINT@E3,CSNG(S2);:PRINT@922,CSNG(S3);
480 A1=A1*R1:A2=A2*R1:A3=A3*R1
490 PRINT@E2,CSNG(A1);:PRINT@815,CSNG(A2);:PRINT@777,CSNG(A3);:PRINT@E2,"*";:PRINT@815,"*";:PRINT@777,"*";
500 IF F3=1 GOSUB 590
510 IF F1=1 GOSUB 560
515 IF F5=1 THEN X=P:GOSUB 10030:P2=X2 ELSE P2=SIN(P)
520 A=(K*M*P2)/2:PRINT@E4,"A=";CSNG(A);
530 PRINT@972,"( FOR FURTHER EXERCISES HIT ANY KEY )";
540 IF INKEY$="" GOTO 540
550 GOTO 20
560 PRINT@287,CSNG(Q1*R1);:PRINT@935,"OR ";CSNG(M1);:PRINT@744,CSNG(P1*R1);
570 PRINT@287,"*";:PRINT@744,"*";
580 GOSUB 615:A0=(K*M1*P3)/2:PRINT@670,"A=";CSNG(A0);:RETURN
590 PRINT@157,CSNG(Q1*R1);:PRINT@901,CSNG(M1);:PRINT@917,"OR";
600 PRINT@786,CSNG(P1*R1);:PRINT@157,"*";:PRINT@786,"*";
610 GOSUB 615:A0=(K*M1*P3)/2:PRINT@656,"A=";CSNG(A0);:RETURN
615 IF F5=1 THEN X=P1:GOSUB 10090:P3=X2 ELSE P3=SIN(P1):RETURN
620 IF F5=1 GOTO 4000 ELSE T=(K+L+M)/2
630 IF P=0 THEN X=SQR(T*(T-L)/(K*M)):GOSUB 780 :P=2*X
640 IF R=0 THEN X=SQR(T*(T-K)/(L*M)):GOSUB 780 :R=2*X
650 IF Q=0 THEN X=-COS(P+R):GOSUB 780 :Q=X
660 RETURN
670 IF F5=1 GOTO 4050 ELSE X=-COS(P+Q):GOSUB 780 :R=X
680 L=K*SIN(P)/SIN(R)
690 M=K*COS(P)+L*COS(R)
700 RETURN
710 IF F5=1 GOTO 4090 ELSE X=-COS(Q+R):GOSUB 780 :P=X
720 RETURN
730 IF F5=1 GOTO 4120 ELSE M=SQR(K[2+L[2-2*K*L*COS(Q)])
740 RETURN
750 IF F5=1 GOTO 4140 ELSE X=L/K*SIN(R):GOSUB 730 :P=X
760 X=-COS(R+P):GOSUB 780 :Q=X
770 RETURN

```

```

780 IF -X*X+1<0 GOTO 1060 ELSE X=(-ATN(X/SQR(-X*X+1))+1.5708):RETURN
790 IF -X*X+1<0 GOTO 1060 ELSE X=ATN(X/SQR(-X*X+1)):RETURN
800 FOR N=1 TO 12
810 N1=N*64+J1
820 PRINT@N1,CHR$(C0);CHR$(C1);CHR$(C2);
830 J1=J1-H
840 NEXT N
850 FOR N=1 TO 12 STEP 2
860 N1=N*64+J2
870 PRINT@N1,CHR$(C3);CHR$(C4);
880 PRINT@N1+V,CHR$(C5);CHR$(C6);CHR$(C7);
890 J2=J2+H
900 NEXT N
910 FOR N=837 TO 891
920 PRINT@N,CHR$(131);
930 NEXT N
940 IF F1=1 GOTO 970
950 IF F4=1 THEN F4=0:GOSUB 1050
960 RETURN
970 FOR N=1 TO 12 STEP 2
980 N1=N*64+J3
990 PRINT@N1,CHR$(D0);CHR$(D1);
1000 PRINT@N1+V1,CHR$(D2);CHR$(D5);
1010 J3=J3-H1
1020 NEXT N
1030 PRINT@E5,CHR$(D4);CHR$(D5);
1040 RETURN
1050 PRINT@ 460,"S1";:PRINT@291,"A1";:PRINT@500,"S2";:PRINT@816,"A2";:PRINT@794,
"S3";:PRINT@777,"A3";:RETURN
1060 PRINT@896,B$;:PRINT@896,"IMPOSSIBLE ! HIT ENTER TO START AGAIN .....";
:INPUT Z:GOTO 20
4000 T=(K+L+M)/2
4010 IF P=0 THEN X=T*(T-L)/(K*M):GOSUB 10190:X=X3:GOSUB 4200:P=2*X
4020 IF R=0 THEN X=T*(T-K)/(L*M):GOSUB 10190:X=X3:GOSUB 4200:R=2*X
4030 IF Q=0 THEN X=P+R:GOSUB 10080:X=-X2:GOSUB 4200:Q=X
4040 RETURN
4050 X=P+Q:GOSUB 10080:X=-X2:GOSUB 4200:R=X
4060 X=P:GOSUB 10090:X5=X2:X=R:GOSUB 10090:X6=X2:L=K*X5/X6
4070 X=P:GOSUB 10080:X5=X2:X=R:GOSUB 10080:X6=X2:M=K*X5+L*X6
4080 RETURN
4090 X=Q+R:GOSUB 10080:X=-X2:GOSUB 4200:P=X
4100 RETURN
4120 X=Q:GOSUB 10080:M=K*K+L*L-2*K*L*X2:X=M:GOSUB 10190:M=X3
4130 RETURN
4140 X=R:GOSUB 10090:X=L/K*X2:GOSUB 4210:P=X
4150 X=R+P:GOSUB 10080:X=-X2:GOSUB 4200:Q=X
4160 RETURN
4200 IF -X*X+1<0 GOTO 1060 ELSE X5=X:X=-X*X+1:GOSUB 10190:X=X5/X3:GOSUB 10140:X=
X9-X2:RETURN
4210 IF -X*X+1<0 GOTO 1060 ELSE X5=X:X=-X*X+1:GOSUB 10190:X=X5/X3:GOSUB 10140:X=
X2:RETURN
5000 IF MS$(>"M" THEN A0=VAL(A0$):RETURN ELSE N2=LEN(A0$):D6=N2-6
5020 D$=LEFT$(A0$,D6)
5030 M$=MID$(A0$,D6+2,2)
5040 S$=RIGHT$(A0$,2)
5050 A0=VAL(D$)+VAL(M$)/60+VAL(S$)/3600:RETURN
10080 X=X9-X

```

```

10090 G0=0:G2=SGN(X):X2=X9+XS:X=ABS(X):X=X-INT(X/(X2+X2))*(X2+X2):IFX>X2THENX=X-
X2:G2=-G2
10100 IFX>X9THENX=X2-X
10110 IFABS(X)<.063THEN10120ELSEX=X/3:G0=G0+1:GOTO10110
10120 X2=-X*X:X2=((X2/42+1)*X2/20+1)*X2/6+1)*X:IFG0=0THEN10130ELSEFORG=1TOG0:X2
=(3-4*X2*X2)*X2:NEXT
10130 X2=X2*G2:RETURN
10140 G0=0:G1=0:G2=SGN(X):X=ABS(X):IFX>1X=1/X:G1=1
10150 IFX<.077THEN10160ELSEX2=X:X=X*X+1:GOSUB10190:X=X2/(X3+1):G0=G0+1:GOTO10150

10160 X3=X:G3=-11:GOSUB10200:IFG0=0THEN10170ELSEFORG=1TOG0:X=X+X:NEXT
10170 X2=X:IF G1=1 X2=X9-X2
10180 X2=X2*G2:RETURN
10190 X3=SQR(X):X3=(X3+X/X3)/2:X3=(X3+X/X3)/2:RETURN
10200 X2=X*X:FORG=3TOABS(G3)STEP2:X3=SGN(G3)*X3*X2:X=X+X2/G:NEXT:RETURN

```

```

1 ' ***** THE WORLD - LEVEL II *****
10 ' *** COPYRIGHT MICRO-80 - 1980
30 DATA-91,84,72,69,-1,87,79,82,76,68,-93,140,143,175,191,191,189,188,188,188,17
6,176,176,176,144,-4,131,187,191,191,191,191,135,-16,176,176,176,-22,160
40 DATA191,191,191,191,191,191,183,147,131,176,176,176,-3,138,135,131,129,-5,160
,176,156,188,188,188,176,176,176,184,190,191,191,191,191,191,191,191
50 DATA191,191,191,191,191,189,188,188,188,188,180,-7,160,184,191,191,191,191,19
1,191,191,191,191,191,147,131,129,-8,136,136,133,160,187,179,177
60 DATA188,191,191,191,191,191,191,191,191,191,191,191,191,191,131,191,191,191,1
91,191,191,183,177,-1,138,132,-7,162,191,191,191,191,191,191,191,159
70 DATA143,131,129,-12,168,188,159,131,139,163,175,191,183,179,187,191,151,171,1
83,191,191,191,191,191,191,191,191,191,191,191,191,191,191,159,133,164
80 DATA-8,138,170,191,151,129,-2,143,-15,176,100,100,100,100,180,176,176,176,179
,179,191,191,191,172,191,191,191,191,191,191,191,191,191,191,191,191,191
90 DATA183,179,129,136,143,132,-9,131,131,143,191,-2,131,130,-12,191,191,191,191
,191,191,191,191,191,191,189,154,175,191,188,191,131,131,143,175,191
100 DATA191,159,143,191,191,191,191,191,135,-17,131,187,191,191,191,189,188,144,
-8,139,143,143,143,143,191,191,191,191,191,191,191,191,189,191,149,-4,138
110 DATA191,149,-2,130,171,175,157,-2,138,-16,138,191,191,191,191,191,191,191,18
8,188,180,-10,170,191,191,191,191,191,191,159,135,129,-11,130,133,-1,176
120 DATA188,-1,131,129,-15,130,139,191,191,191,191,191,191,191,-12,190,191,191,1
91,191,191,149,-1,176,-19,130,139,141,180,-13,175,191,191,191,159,131,-13,138
130 DATA191,191,191,143,-2,138,129,-14,176,176,184,188,190,191,184,191,-14,170,1
91,159,140,-16,131,131,-19,168,191,191,191,191,191,191,191,191,143,-13,130:DATA1
75,181,144,-37,131,131,-2,131,143,143,135,129,-2,160,-57,131,-2,140,140,140,131,
129
140 ONERRORGOTO160:CLS:M=15360:POKE16553,255:RESTORE
150 READA:IFAK0THENM=M-A:GOTO150ELSEPOKEM,A:M=M+1:GOTO150
160 IFERR=6THENRESUME170ELSEONERRORGOTO0
170 GOTO170

```

** SOLVER 4 FOR 16K MACHINES **

Bob Sunners sent us a program called Solver which simulates the TI PROGRAMMER calculator and does much more besides. He also provided us with a simplified version which fits in 4K and the same version with high line numbers so that it can be co-resident with other BASIC programs. Bob's documentation was first class and an example to us all. We have, therefore, included it exactly as he presented it to us.

The smaller version of SOLVER, this program is compatible with 4K computers as a sole program, or with 16K+ computers as a co-resident program for ready access during programming operations on a separate subject.

For the 4K operator SOLVA4 will occupy the entire memory, and can carry out the following functions -

- * Perform arithmetic in either decimal or hexadecimal.
- * Signed floating point arithmetic in decimal for normal calculator operations.
- * Conversion of integers between decimal and hexadecimal.
- * Parenthesis - available at up to 10 processing levels to allow you to dictate the order of operation of a mathematical sequence.
- * Display format - the screen will display the following information -

		X DEC	H HEX	SPACE	Q QUIT			
(OPEN) CLOS	/ DIV	* MULT	+ ADD	- SUBT	= EQU	> SUBT	
0 0000	1 0001	2 0010	3 0011	4 0100	5 0101	6 0110	7 0111	
8 1000	9 1001	A 1010	B 1011	C 1100	D 1101	E 1110	F 1111	

DECI

The top four lines indicate the keys to be used in the calculations. Line 5 indicates the current base mode.

- X DEC : Converts base to Decimal. Decimal mode is default condition on power-up.
- H HEX : Converts latest displayed figure or total to Hexadecimal. Shows value as negative if decimal integer is negative.
- SPACE : Clears entered calculations and returns calculator to 'Ready' state.
- Q QUIT : Ends program and returns computer to 'Ready'.
- ← : Not indicated on display. Left arrow will backspace over an incorrect numeric entry. Has no effect if pressed after an operand entry.
- (OPEN : Opens up to 10 levels of parenthesis to isolate expressions for execution in a desired sequence.
-) CLOS : Closes parenthesis. All calculations on that level will be performed before the next entry is made. If left off before "=" is entered, levels will automatically be correctly closed.
- / DIV : Divides the current subtotal by the next entered value
- * MULT : Multiplies the current subtotal by the next entered value
- + ADD : Adds the current subtotal to the next entered value.

- SUB : Subtracts the next entered value from the current subtotal.
- = EQU : Completes all previously entered operations and displays the result.
- > SUBT : Displays the current subtotal without affecting the calculations.
- 0 to F : Numerical value keys - only keys 0 - 9 will function in the Decimal base.
Adjacent to each number is the binary value of that number in the Hexadecimal base. Logical AND, OR, NOT, ONE'S AND TWO'S COMPLEMENT AND SHIFTS can be performed easily on paper from any Hex results displayed. Hex to Binary or reverse can be quickly calculated on paper from these displays.
Note that totals are stored as double precision decimal floating point signed numbers and that conversion to HEX integers does not affect the re-converted results in DEC.

Additional features of this programme are graphics characters representing HEX or DEC which are inserted automatically into the calculation chain display to provide visual recognition of the point at which conversion occurred.

The first entry must be positive. A negative first entry is possible by entering "0", "-" in sequence.

If an incorrect entry is made, pressing the left-arrow (backspace) key before any symbol key clears the incorrect number without affecting any calculations in progress.

When an unwanted "+" or "-" or "*" or "/" key is pressed, simply press the correct key and continue.

Conversions are straight forward:

Example: $328_{10} = ?_{16}$

Enter	Display
X	
328	328
H	>148<

Means $328_{10} = 148_{16}$

Arithmetic operations may be combined with conversions:

Example: $FA_{16} + 73_{10} = ?_{16}$

Enter	Display
H	
FA	FA
X	>250<
+ 73	+ 73
=	323
H	>143<

Means $FA_{16} + 73_{10} = 143_{16}$

Pressing any number key following the "=" key will add that number to the resulting answer. To recommence new calculations after the "=" key, the SPACE key must first be used.

SOLVER. 16K

This larger version of SOLVA4, apart from its many extra functions, differs slightly from the SOLVA4 operations.

- SHIFT D : Converts to Decimal base mode. Decimal base is default condition on power-up
- SHIFT H : Converts to Hexadecimal mode.
- SHIFT B : Converts to Binary mode.
- SHIFT O : Converts to Octal mode.
On conversion to each of these bases, the latest displayed value is redisplayed in the new mode. Operations can be carried on continuously from mode to mode.
- SHIFT M : Clears previous memory content and loads the latest displayed figure into memory.
- SHIFT P : Adds the latest displayed figure into memory. If latest display is a negative total, it adds this into memory as a negative value.
- SHIFT R : Displays the memory total without affecting the contents of memory. Automatically converts the recalled amount to the current base.
- SHIFT K : Stores a constant value and operand for repetitive use. After the constant and its operand are stored, the entry of any number followed by "=" automatically executes the operation stored by the K key. pressing SHIFT K a second time clears this function.
- + - * / = () : Keys operate as with SOLVA4.
- @ +/- : Converts decimal values to opposite sign. When in any other base, pressing this key produces the Two's Complement value of any displayed figure or total.
- 0 to F : Operate as for SOLVA4.
Values 0 through 9 operate in Decimal.
Values 0 through F operate in Hexadecimal.
Values 0 through 7 operate in Octal.
Values 0 and 1 operate in Binary.
- Q QUIT : Ends calculator program.
- > SBTO : Returns the current subtotal without affecting the calculations.

- , REST : Restores the total of the previous cleared operation.
- ← BACK : Backspaces over the latest numerical entry.
Has no effect if the last entry was an operand.
- : 1'S : One's Complement Key - instructs the program to immediately convert the displayed number to its one's complement. Equivalent to the logical NOT function. Operates in all bases.
- ; OR : Or's the next entry with the current subtotal.
operates in all bases.
- & AND : And's the next entry with the current subtotal.
Operates in all bases.
- ? XOR : Xor's the next entry with the current subtotal.
Operates in all bases.
- < SHFT : Perform a logical shift on the current subtotal.
- For a Left shift, enter "<" followed by a positive value indicating the number of zero bits to be added to the right of the binary value to be shifted.
- For a right shift, enter ">" followed by "-" and then a value indicating the number of bits to be truncated from the right of the target value.
- All shifts are carried out bit by bit and so only zeroes are added to the target value to perform a logical shift.
- Operates in all bases.
- SPACE : Erases the display but does not erase the constant or memory functions.
- ' NEW FORMAT : Converts the operation mode of the decimal display between 16 bits and unformatted modes. Z-80 or 16 bit format is the default mode on power-up, which displays decimal values and only carries out operations between -32768 and + 32767.
- ERROR TRAPPING : An error-trapping routine is included which recognises any of the following entry mistakes, and indicates its recognition to the user by flashing the offending entry and automatically removing it from screen:-
- * "/" entry followed by "0".
 - * ")" entry followed by any number.
 - * "{" entry followed by an operation symbol other than "(" or ">".
 - * ")" entry which would indicate closing of a lower level of parenthesis than has been opened by "(" entries.
 - * Operational symbol followed by ")".
 - * Entry of any symbol other than "(" or ">" at the commencement of a calculation.

OPERATING HINTS : Current totals and operations are stored and performed as signed floating point decimal values, which are converted to integer when changing bases from decimal.

All values are stored and operated on as double precision.

Parenthesis specify the order in which operations are to be carried out. Operations in higher levels are executed first and their results combined with the next lower level for execution. Pressing "=" before all closing parentheses are entered will not create an error - the program will execute each level from highest to lowest before displaying the results.

This software has been tested and to my best knowledge performs accurately if operated as described above. Should any problems arise with the program please advise by letter stating the nature of the fault and the routine which led to the fault to:

Robert A. Sunners,
26 Suncroft Street
Mt. Gravatt 4122
Qld. 07 349 2598

60000 REM#####

SOLVA4 FOR 16K COMPUTERS..
(C) ROBERT A SUNNERS 1980
26 SUNCROFT ST MT GRAVATT
4122 BRISBANE 07 349 2598

```

60010 REM#####
60020 PRINT@960,CHR$(31);:PRINT,"DO YOU REQUIRE FULL DISPLAY (Y/N)";:GOSUB60030:
IFP$="N"THENPA=832:PB=960:GOSUB60380:GOTO60070ELSE60040
60030 P$=INKEY$:IFP$=""THEN60030ELSERETURN
60040 CLS:PRINT@23,"* S O L V A 4 *":PA=576:PB=960:GOSUB60380:PL$=" "
60050 PRINT@640,"";:PRINT,"X DEC. H HEX. SPACE Q QUIT",,"( OPEN";PL$;" ) CLOS
";PL$;"/ DIV. ";PL$;" * MULT";PL$;"+ ADD. ";PL$;"- SUB. ";PL$;"= EQU. ";PL$; "> SUBT"
60060 PRINT"0 0000 1 0001 2 0010 3 0011 4 0100 5 0101 6 0110 7 0111 8 10
00 9 1001 A 1010 B 1011 C 1100 D 1101 E 1110 F 1111"
60070 CLEAR200:DEFDBLQ:CR$="0123456789ABCDEF":PRINT@989," DECI ";:FP=1
60080 PA$=STRING$(58," "):PI$="":P$="":PRINT@899,PA$;:FORX=0TOFL:PI(X)=0:FF(X)=0
:FL(X)=0:QT(X)=0:NEXT:FL=0:FQ=1:FA=1:FI=1
    
```

```

60090 REM                               INPUT SEQUENCE
60100 P$=INKEY$: IFFP$="" THEN60100ELSE IFFP<>1ANDP$="." THEN60100ELSE IFFP$="Q" THENCLS
:END
60110 IFFP=1ANDASC(P$)>62ANDP$<>"H"ANDP$<>"X" THEN60100
60120 IF(P$="." )OR(ASC(P$)>47ANDASC(P$)<58 )OR(ASC(P$)>64ANDASC(P$)<71 ) THENPI$=PI
$+P$: IFFA=1 THENP$=" "+P$: GOSUB60300: FA=0: GOT060100ELSE GOSUB60300: GOT060100
60130 IFFP$=CHR$(32) THEN60080ELSE IFFP$=CHR$(8) THENPA$=STRING$(LEN(PI$), " ") +LEFT$(
PA$, LEN(PA$)-LEN(PI$)): PI$=" ": PRINT@899, PA$; : GOT060100
60140 IFFP$="X" ORP$="H" THEN60320: GOT060100
60150 IFFP$<>" "ANDP$<>" "ANDP$<>"-"ANDP$<>"*ANDP$<>"/ANDP$<>"("&"ANDP$<>")"ANDP$
<>">" THENPI$=" ": GOT060100ELSE GOSUB60300
60160 REM                               SELECT VALUE
60170 FA=0: IFFP$="( " THEN60210ELSE IFF(PI$="" ) THEN60200ELSE IFFP>1 THENFS=FP: FD=1: PN$=
PI$: GOSUB60420: PI(FL)=VAL(PN$) ELSE PI(FL)=VAL(PI$)
60180 REM                               CALCULATIONS
60190 IFFF(FL)=0ANDQT(FL)=0 THENQT(FL)=QT(FL)+PI(FL) ELSE IFFF(FL)=1 THENQT(FL)=QT(FL)
/PI(FL) ELSE IFFF(FL)=2 THENQT(FL)=QT(FL)*PI(FL) ELSE IFFF(FL)=3 THENQT(FL)=QT(FL)+P
I(FL) ELSE IFFF(FL)=4 THENQT(FL)=QT(FL)-PI(FL)
60200 FF(FL)=0: PI$=" ": IFFP$="/" THENFF(FL)=1: GOT060100ELSE IFFP$="*" THENFF(FL)=2: GOT
060100ELSE IFFP$="+" THENFF(FL)=3: GOT060100ELSE IFFP$="-" THENFF(FL)=4: GOT060100
60210 IFFP$="( " THENPI$=" ": IFFQT(FL)>0 THENFL(FL)=1: FL=FL+1: GOT060100ELSE FL=FL+1: GO
060100
60220 GOSUB60410: IFFP=2 THENFS=1: FD=2: GOSUB60420
60230 PT$=PN$: IFFI=-1 THENPT$="-" +PT$
60240 IFFP$=">" THENPN$=PT$+"<": GOSUB60300: GOT060100
60250 REM                               EQUALS OPERATION
60260 IFFP$="=" THENIFFP=1 THENPT$=LEFT$(PT$, 16)
60270 IFFP$="<=" THENIFFL>0 THENGOSUB60280: P$=" ": GOT060190ELSE P$=PT$: GOSUB60300: FF(
FL)=0: FA=1: GOT060100ELSE IFFP$=")" THENGOSUB60280: P$="[" : GOT060190ELSE IFFP$="[" THEN6
0100
60280 IFFL=0 THENRETURN ELSE PI$=" ": FF(FL)=0: PI(FL)=0: FL(FL)=0: FL=FL-1: PI(FL)=QT(FL
+1): QT(FL+1)=0: RETURN
60290 REM                               DISPLAY OPERATION
60300 PA$=RIGHT$(PA$, LEN(PA$)-LEN(P$))+P$: PRINT@899, PA$; : IFFP$="" THENPN$="0": GOT06
0300ELSE RETURN
60310 REM                               CHANGE BASE
60320 IFFP$="X" THENFQ=1: P$=CHR$(170): GOSUB60300: PRINT@989, " DECI "; ELSE IFFP$="H" TH
ENFQ=2: P$=CHR$(162): GOSUB60300: PRINT@989, " HEXA ";
60330 IFFQ=FP THEN60100ELSE IFFQ=1 THENIFQT(FL)<>0 THENGOSUB60410: GOSUB60350: PT$=PN
:P$=">": PI$=PN$: GOSUB60300: FP=1: GOT060240
60340 IFQT(FL)<>0 THENFS=1: FD=2: GOSUB60410: GOSUB60420: GOSUB60350: PT$=PN$: P$=">": P
I$=PN$: GOSUB60300: FP=2: GOT060240ELSE FS=FP: FD=FQ: GOSUB60400: PN$=PI$: GOSUB60420: GO
SUB60350: PT$=PN$: P$=">": PI$=PN$: GOSUB60300: FP=FQ: GOT060240
60350 GOSUB60360: IFFI=-1 THENPN$="-" +PN$: RETURN ELSE RETURN
60360 IFSGN(QT(FL))<>0 THENQT(FL)=-QT(FL): RETURN ELSE RETURN
60370 REM                               DRAW LINES
60380 PRINT@PA, CHR$(31); : FORX=3 TO60: PRINT@PA+X, CHR$(140); : PRINT@PB+X, CHR$(140); :
NEXT: RETURN
60390 REM                               REMOVE SIGN
60400 IFLEFT$(PI$, 1)="-" THENPI$=RIGHT$(PI$, LEN(PI$)-1): RETURN ELSE RETURN
60410 FI=SGN(QT(FL)): PN$=RIGHT$(STR$(QT(FL)), LEN(STR$(QT(FL)))-1): RETURN
60420 IFFS=FD THENRETURN ELSE IFFS=1 THENGOSUB60470: RETURN ELSE GOSUB60440: RETURN
60430 REM                               HEX TO DEC
60440 CD=0: CA=1: FORX=1 TOLEN(PN$): CB$=MID$(RIGHT$(PN$, X), 1, 1): IFCB$<"A" THENCB=VAL
(CB$): GOT060450ELSE CB=ASC(CB$)-55
60450 CD=CB*CA+CD: CA=CA*16: NEXT: PN$=RIGHT$(STR$(CD), LEN(STR$(CD))-1): RETURN
60460 REM                               DEC TO HEX
60470 CA$="": CD=VAL(PN$)
60480 CD=CD/16: CA=(CD-INT(CD))*16: CA$=MID$(CR$, CA+1, 1)+CA$: CD=INT(CD): IF CD=0 THEN
FN$=CA$: RETURN ELSE60480

```

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

We're sorry KBFIX mysteriously disappeared just before we went to press. We will publish it next month plus at least the following programs:

BIG LETTERS (L1)

Reproduce letters and words in a 4 cm x 4 cm format. This program produces 3 lines with up to 8 letters each.

CONCENTRATION (L2)

Inspired by those TV shows, in this game the screen displays a row of cards, facing away from you. You must select (guess?) the two identical cards to score. For 1 or 2 players.

RESISTOR CODES (L2)

If you are like us and have difficulty remembering whether 5 is green or purple (or is it yellow?) then this is the program for you.

SPACE INVADERS (L1)

Who said you couldn't write a Space Invaders game in Level 1. This version has all the aliens you could want and saves all those 20 cents!

TANK BATTLE (L2)

Drive your tank through the maze until you can get a clean shot at your opponent.

SYS COPY (L2 ml)

This program loads machine language programs into your computer then displays the Start, End and Entry points so that you can punch out a copy using T BUG, EIMON etc.

tick where appropriate

To MICRO-80

Please consider the enclosed program for...

() Publication in MICRO-80

() Publication on disk or cassette only

() 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, Editor, 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 - padlocks are suggested - and enclose stamps or postage if you want your cassette or disk returned.

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