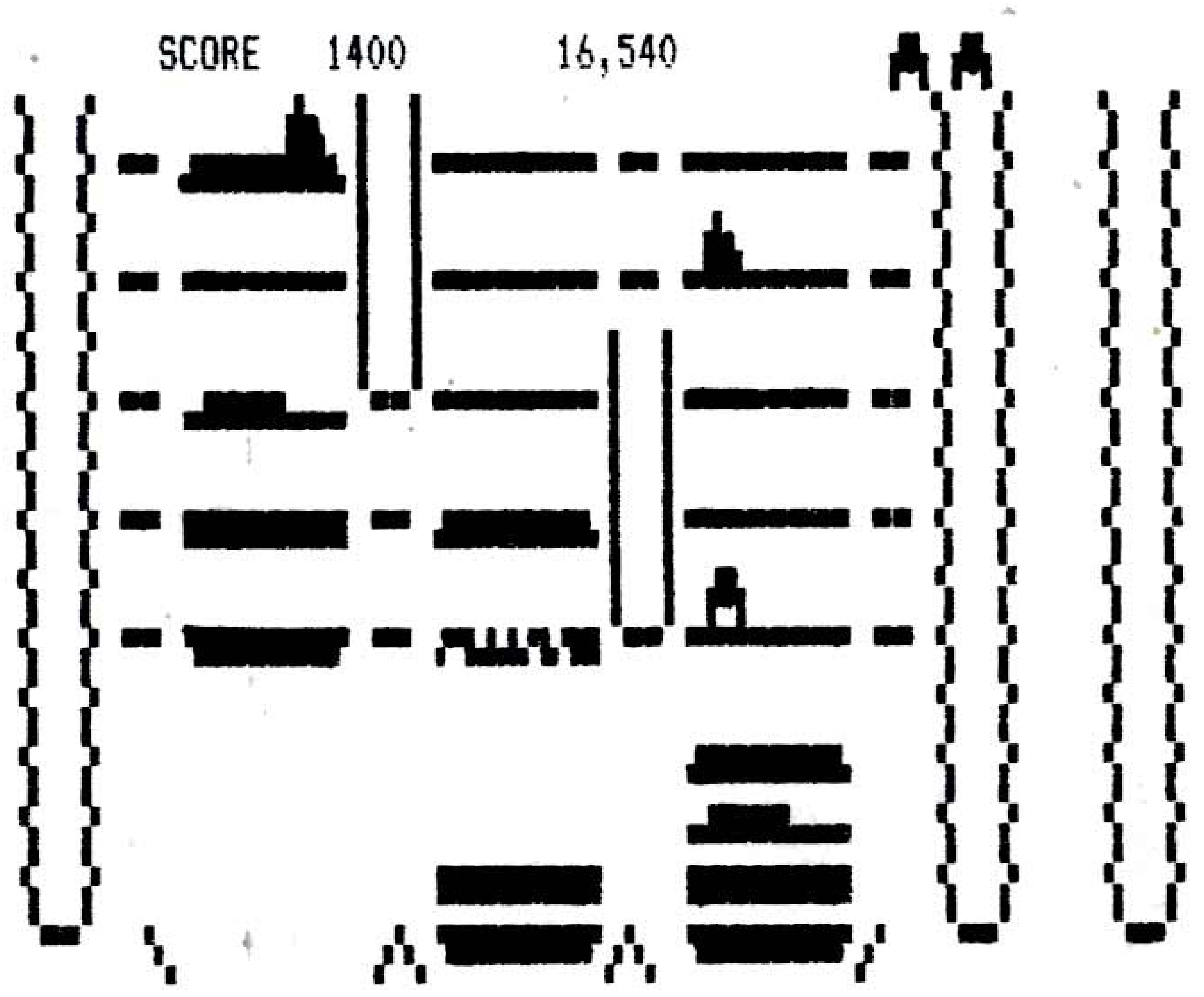
MICRO-80

P.O. BOX 213, GOODWOOD, S.A. 5034. AUSTRALIA. TELEPHONE (08) 211 7244. PRICE: AUS. $2.50, N.Z. $4.00, U.K. £1.50
Registered by Australia Post — Publication No. SBQ2207

Vol. 4, Issue 3, October 1983



HAM BURGER

Also in this issue:

PROGRAMMING:
The Midnight Printer

REVIEWS:
SK-2 Speed Up Mod. for the TRS-80
Infinite BASIC
Hi-Writer for the Peach

SOFTWARE:
Block—Peach
Boxing—Colour
Galactic Battles — Level I
Hi-Res Screen Score—Colour
Othello—Peach
System Tape Maker—Level II
XUSR Screen Fill Subroutine—Level II

• TRS-80 • SYSTEM 80 • VIDEO GENIE
• PMC-80 • HITACHI PEACH
• TRS-80 COLOUR COMPUTER

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

EDITOR:	RYSZARD WIWATOWSKI
ASSOCIATE EDITORS:	
SOFTWARE :	CHARLIE BARTLETT
HARDWARE :	EDWIN PAAY

MICRO-80 is an international magazine devoted to the Tandy TRS-80 Model I, Model III and Colour microcomputers, the Dick Smith System 80/Video Genie and the Hitachi Peach. It is available at the following prices:

	12 MONTH SUB.	SINGLE COPY
MAGAZINE ONLY	$ 26-00	$ 2-50
CASSETTE PLUS MAGAZINE	$ 65-00	$ 4-00 (cass. only)
DISK PLUS MAGAZINE	$125-00	$10-00 (disk only)

MICRO-80 is available in the United Kingdom from:
U.K. SUBSCRIPTION DEPT. 24 Woodhill Park, Pembury, Tunbridge Wells, KENT TN2 4NW

MAGAZINE ONLY	£ 16-00	£ 1-50
CASSETTE PLUS MAGAZINE	£ 43-60	£ N/A
DISK PLUS MAGAZINE	£ 75-00	£ N/A

MICRO-80 is available in New Zealand from:

MICRO PROCESSOR SERVICES, 940A Columbo Street, CHRISTCHURCH 1 N.Z. Ph 62894

MAGAZINE ONLY	NZ$ 43-00	NZ$ 4-00
CASSETTE PLUS MAGAZINE	NZ$ 89-00	NZ$ 5-00
DISK PLUS MAGAZINE	NZ$175-00	NZ$15-00

MICRO-80 is despatched from Australia by airmail to other countries at the following rates:

(12 MONTH SUB)	MAGAZINE	CASS + MAG	DISK + MAG
PAPUA NEW GUINEA	Aus$40-00	Aus$ 83-00	Aus$ 143-00
HONG KONG/SINGAPORE	Aus$44-00	Aus$ 88-00	Aus$ 148-00
INDIA/JAPAN	Aus$49-00	Aus$ 95-00	Aus$ 155-00
USA/MIDDLE EAST/CANADA	Aus$55-00	Aus$102-00	Aus$ 162-00

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, System 80/Video Genie or Peach and its peripherals. MICRO-80 is in no way connected with any of the Tandy, Dick Smith or Hitachi 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 microcomputer to earn some extra income is included in every issue.

** CONTENT **
Each month we publish at least one applications program in BASIC for each of the microcomputers we support. We also publish Utility programs in BASIC and Machine Language. We publish articles on hardware modifications, constructional articles for useful peripherals, articles on programming techniques both in Assembly Language and BASIC, new product reviews for both hardware and software and we print letters to the Editor.

** COPYRIGHT **
All the material published in this magazine is under copyright. This 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.

***** CONTENTS *****

REGULARS

PAGE

DEPARTMENTS

PROGRAMMING

REVIEWS

SOFTWARE

MICRO-80 is Registered by Australia Post - Publication No. SBQ2207

AUSTRALIAN OFFICE AND EDITOR:
MICRO-80
P.O. BOX 213
GOODWOOD, SOUTH AUSTRALIA, 5034.
TEL. (08) 211 7244

U.K. SUBSCRIPTION DEPT:
24 WOODHILL PARK, PEMBURY,
TUNBRIDGE WELLS, KENT TN2 4NW

Printed by:
Shovel & Bull Printers,
379 South Road,
MILE END, S.A. 5031.

Published in Australia by:
MICRO-80
433 Morphett Street,
Adelaide, S.A. 5000.

***** EDITORIAL *****

Most of us who own and use personal microcomputers spend much of our time waiting for something. Tandy has now released its new computers into the marketplace, or has announced their imminent appearance. If you have been waiting for this and now go out to buy one, you will find that you will be waiting again for the arrival of the new software to go with your machine.

The new Model 4 is certainly an attractive product that shows more advanced hardware and software over its predecessor. But although the Model 4 was released in the U.S. late in April, the only software available to date to make use of the new power and features, is the TRSDOS 6.0 operating system and Version 5 of Microsoft's Disk BASIC. Typically with any new machine, you are faced with an acute lack of software in the beginning between the time the computer is released and the desired software becomes available. The coming MC-10 Micro colour computer will probably start in similar circumstances, but the Model 100, being a rather different concept as far as computers go, is relatively unique in that it comes with most of the software you need already built in. Other exceptions to this rule are computers like the System-80, the Osborne, etc. that are built to take advantage of a large, already established software base.

In defence of the manufacturer, I must point out that this situation is perhaps inevitable. Software development cost, in terms of time and money, is usually much greater than that for hardware development, and it is seldom economical or practical to delay release of hardware until the software development is complete. Unfortunately, this does leave the new computer owner quite vulnerable to some enterprising but unscrupulous software vendors who inundate the market with, shall we say, inadequate software. The user finds the initial frustration produced by the lack of software compounded by the experience of a rash of poor quality software. A word of caution to those of you who may find yourselves treading this path - and perhaps an argument in favour of software libraries which will enable users to preview software first hand before buying.

We have some first hand experience of the problems associated with software development, and for those readers patiently waiting for the arrival of their new free software, I have some good news. By the time you read this, our new SOFTPACK will be ready to go into production and we expect to start sending it out by the end of this month. Over the past few months all the programs have been undergoing a thorough testing and I am glad to say that a number of bugs have been discovered. This may sound bad and has caused delays through the need to make corrections to both the programs and manual, but this inconvenience is a small price to pay for the greater quality and reliability of the final product. We would not be so bold as to claim that it is entirely bug-free, but have every confidence that it will be. I am sure that you will find that it has been well worth waiting for.

- 0000000000 -

***** PEEKing (UK) by Tony Edwards *****

These days seem to bring a new computer every few weeks, each better than the last. What do those of us who have an old computer do? With the advent of new computers which are all high resolution and colour the old machines are looking very dated, especially the '80 units. The obvious line to take is off with the old and on with the new, but that brings with it a number of problems. The cost of a new computer is not too high these days but it is hard to give up one's old machine which may have a number of personalised additions and lots of add-on memory. If one can give up the hardware, what about the software? Any computer user soon builds up a large collection of program tapes of well-loved games and well-used utilities. More to the point, a user builds up a store of information and techniques for his machine. When we consider changing machines do we consider the loss of all these extras?

There is now an alternative to junking an '80 and starting with a new machine. I have recently been shown a unit which simply plugs into the back of the '80 and converts it to a high resolution colour computer. The real value of this device is that all the user's programs and techniques still work because the machine is still an '80. With only the smallest changes the programs will run in colour and with a little more effort they can be run in high resolution. The cost of this device is estimated at about £100. At that price it must be a good buy at less than the cost of a new computer as it allows one to retain all those '80 add-ons.

With about 100,000 '80 machines in the U.K. there must be a large market for such an add-on. I hope to get my hands on one soon so watch this space for further reports.

- 0000000000 -

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

From: Dave Edwards - Dunedin, N.Z.

Like many owners of early System-80 machines, I have not yet got around to adding the "tab" and "clear" keys that the more recent machines have, and as a result, I have problems running some programs.

Killer Gorilla is one such program and in setting out to modify it, I discovered an apparent error in the "Level 2 ROM, Assembly Language Toolkit". On p.48, which shows the keyboard matrix, the labels of the "Less than" and "Greater than" keys seem to be transposed. The result was that the left and right movements of my modified program were reversed (not making for a high score in the game)!
For readers interested in the modification (corrected!) I used TASMON to change the necessary memory locations and to save the new version onto tape (no doubt DBUG would also work).

My version now uses "<" and ">" for left and right movement, "+" and "?" for up and down and "*" for jumping. The reason for this choice is firstly that the modification is simpler if all keys are in the same matrix row, and secondly, the chosen keys are in positions that can easily be reached with the fingers.

The memory locations to change are as follows:

Location	Original Value	Change to
746D	4ØH	2ØH
7475	67H	7FH
75B4	4ØH	2ØH
75B7	EØH	54H
75B9	CØH	44H
75BE	AØH	14H
75C3	8ØH	Ø4H
7615	4ØH	2ØH
761D	6FH	67H

Thanks for a very enjoyable magazine.

(The "less than" and "greater than" symbols are, in fact, transposed. I'm sure our readers with early model System-80's will appreciate your modifications to Killer Gorilla. - Ed.)

From: D.J. Davies - Mount Waverley, Vic.

Having just changed my TRS-80 Model I for a Model III I was pleased to see your new feature 'Form Three' in the latest edition of MICRO-80.

As expected my Model I BASIC programs run on the III without any difficulty and Tandy's machine language games 'Space Invaders' and 'Pyramid' also work but, as was pointed out in the MICRO-80 review of the Model III (Issue 15, Feb. 1981) machine language utility programs which make ROM calls do not usually work. Microsoft's Z-BUG runs with a few minor peculiarities but Editor/ Assembler Plus will not load and MICRO-80's BMON, SKEY and KEWORD (Issue 5, Apr. 1982) crash as soon as they are initialised. I should be grateful for advice from you or from readers, perhaps as articles in 'Form Three', about how to adapt these valuable utilities for the Model III.

Any information you may have about using JPC Products TC-8 high speed cassette interface with the Model III would also be appreciated.

(Those machine language programs that expect a Model 1 low RAM configuration are usually not compatible with the Level 2 Model 3. The task of adapting such programs to run on the Model 3 may require substantial changes, relocation or perhaps a complete rewrite - all of which are very time consuming. However, we would be pleased to publish any patches that our Model 3 readers send in.

We have no information concerning the interfacing of the TC-8 with the Model 3 and I would suggest contacting J.P.C. Products directly. Their address is:

12021 Paisano Ct. NE
Albuquerque, NM 87112

- Ed.)

From: Mr. D.A. Greenham - Seymour, Vic.

I am a newcomer to this game and I don't quite understand what is meant by Level 1 in Level II, sometimes printed Level 2. Your paragraph in the August '82 edition on page 36 doesn't make sense to me - maybe I'm stupid, but it says "Level 1 programs can only be loaded into a Level 1 TRS-80 if the 'level 1 in level 2' program from the MICRO-80 Software Library - Vol.1 is first loaded into your Level 2 TRS-80 or System 80". This, I can't understand. Does this mean that I can't run a Level 1 program on my Level 2 System 80 unless I feed in a "conversion" program?

On your cassette / disk index you show under Type, LII/16K, I presume this is Level 2, 16K Bytes. Sometimes you use "II" and other times "2" - a bit confusing!

Through it all I am enjoying my System 80 even though the tape counter is different from yours - do you have a "conversion" scale for "old" System 80's to "new" ones? This would be valuable.

(First of all, a BASIC interpreter is a machine language program that the CPU executes in order to interpret a higher level BASIC Program. The Level 1 TRS-80 Model 1 has a small 4K BASIC interpreter stored in ROM while the Level 2 Model 1 has a larger 12K interpreter. The Level 2 interpreter cannot execute programs written for the Level 1 interpreter and vice versa. However, since the Level 1 interpreter is a machine language program, it is possible to relocate it into the RAM of a 16K Level 2 machine and therefore, to run Level 1 programs. This is exactly what the "Level 1 in Level 2" program achieves.

The terminology 'L2/16K' (or LII/16K) means that the program is written in Level 2 BASIC and is intended for a machine with at least 16K bytes of memory.

No, we don't have a "conversion" scale but it should be a simple job for you to make one up by using the counter values given in the magazine and determining the counter positions for your own machine. - Ed.)

From: D.E.W. Gillingham - Ferny Grove, Qld.

My computer is a System 80 Mark I, of vintage just pre-Blue Label. It has the lower case character ROM but no ROM driver. I operate dual MPI B51 drives and a Tandy LP VIII printer (bought just before their sale!!!) My usual DOS is Model I TRSDOS Version 2.3. I have two problems, and one software request:

(a) After execution of utilities such as BACKUP or FORMAT under TRSDOS, the system will not return to DOS READY. I have to RESET. Suggestions or cure???

(b) I entered your Lower Case Driver Routine 1.0 from Issue 18, using NEWDOS 80 Version 2.0's EDTASM, and could not make it operate despite (fairly) careful checking of the source code. The error message suggested I was trying to load into ROM. Are you aware of any obvious problems?

(c) Model III TRSDOS provides a FORMS command giving a measure of control over a (compatible) printer, utilising the hardware capabilities of the machine. LDOS provides an even more sophisticated printer driver. What are the chances for we poor Model I or System 80 users of a relocatable printer driver providing:

. selectable form length (physical length)

. selectable lines printed per page

. selectable form width

. selectable first column for printing (left margin)

. selectable (or fixed) indent for continuation

lines in listings.

(My System 80 also hangs after completing a BACKUP or FORMAT under TRSDOS. The reason why the system hangs is that these particular utilities use the HALT instruction (76H) to reboot the DOS when they have finished. This works fine on a Tandy Model 1 but not on the System 80. This is due to a minor hardware difference between the two. On the Model 1, the Halt pin on the Z-80 is physically connected to the RESET pin and when the HALT instruction is executed, the HALT line is pulled low, resetting the CPU. On the System 80 however, the HALT pin is not connected to the RESET line so that after the execution of the HALT instruction, the CPU is halted - you have to reset it manually. The problem can be cured by changing the HALT instruction to a RST Ø (C7H). However, there may be more than one of these in each of these utilities (unlikely) or there may be other 76H's that are parts of other instrutions or data (more likely) and must not be changed. You would need to patch these files on disk. As you also have NEWDOS80 V2.0 why not try this:

1) Use SUPERZAP to find all the 76H's in the files FORMAT/CMD and BACKUP/CMD, making a note of where they occur. If you're lucky there will be only the one in each file and you can change it to C7H.

2) If there is more than one, then you'll need to proceed by trial and error. Change one of them, boot TRSDOS and try to FORMAT a disk. If it works, you've found it - if not, then cross that one off the list and try another and repeat the test.

The answer to the second question is partly tied up with the discussion of lower case modifications in last month's Group One column and some other relevant comments appear there this month.

As for your last request, I can say the chances are quite good now that MICRO-80 readers are aware of your needs and know that we'll be only too happy to publish such a utility. - Ed.)

From: Brendan Moroni - Bendigo, Vic.

I am writing this letter in the hope that you or one of your readers may be able to help me. I have a System 80 with no expansion unit and have Radio Shack TRS-80 software called "In Memory Information". In this program it has the ability to send information to the printer, but because of the fact that it addresses it to the wrong port, I don't get anything. This program has a strange way of loading (I think so anyway). You type SYSTEM and answer the first prompt with the name then it starts to load stopping after a short time giving you the second prompt, which is answered with a slash. This starts the loading again and when it finishes it goes into auto run. I have tried to load it into Microsoft Editor/Assembler-Plus with no result and as I am only learning, I would appreciate any help I can get. Also could you please explain why you put programs on the cassette in System type and also EDTASM type. Please keep up the good work you do in printing MICRO-80.

(Your conclusion regarding printer addressing is quite correct. In the June '82 issue, we published a number of patches for some programs so that they could drive the System 80 printer, but we have none for this particular program. Perhaps some other reader can assist here?

The "strange way" of loading you describe is quite common and is intended to provide some protection against unauthorised copying. The "In Memory Information" program is most probably stored on tape in a form that cannot be read by the usual ROM routines. Therefore, a special program loader is provided on the tape immediately before the main program which is loaded first by the SYSTEM command. When control is passed to it, it loads and auto-starts the main program.

To understand the difference between SYSTEM-type and EDTASM-type files requires some knowledge of both machine language and assembly language programming and a complete discussion of these is out of place here. However, only SYSTEM-type (or object) files can be loaded by the SYSTEM command and only EDTASM-type (or source) files can be loaded by Editor/Assembler. You cannot load an object file into an Editor/Assembler. Because the object file is generated from the source file by an Assembler, we provide both files on cassette and disk editions for those readers who may wish to modify these programs. - Ed.)

From: Stuart McMinn - Pascoe Vale, Vic.

I am prompted to writing this letter after reading a similar type of letter published in the August 1982 issue of MICRO-80. Similar to the reader who wrote in on this occasion, I also have been waiting a fair while to see my program in print.

It seems unfair to the people who submit these programs as they tend to put in a lot of effort to achieve their goal when writing a program, only to have to wait such a long time for it to come into print. People submit their programs to MICRO-80 in good faith, hoping that it will be printed as soon as possible. If people knew of the long delay involved between actual submission and publication, they would seek elsewhere to get their program published.

In my case, I have been waiting now for thirteen months (not long when compared to Mr. Merrylees), for my program to be published. During this time I submitted an article to MICRO-80 on adding a low cost numeric keypad for the System-80. The time between submission and publication was approximately three months. (Understandable for hardware, as it would seem unlikely that you would be inundated with reams of articles on the subject of hardware from readers). Nevertheless, I am disappointed with respect to this, as I received no notification as to whether the article was accepted or not, whereas the opposite is true in the case of software. It was not until it was published in the August 1982 issue that I was even aware that it had even been accepted for publication.

As an editorial comment to Mr. Merrylees' letter you stated that you have many programs in waiting for publication, and I quote: "We can't do a lot about it, short of making each issue about 500 pages long to clear the backlog of programs." Would it not be fairer to all concerned then, if you published more programs per month than the usual (to quote your own advertising in APC) "6 new programs per issue"?

Please excuse the abruptness of my criticisms, but I find that others have also found themselves in a similar situation with regards to software and/or articles submitted to your magazine. I, as did Mr. Merrylees, would like to see this letter published in MICRO-80.

(I will admit that you should have been notified of your article's fate before seeing it in print and shall endeavour to avoid the repetition of such an occurrence. However, I must reiterate the comments of our Software Editor with respect to programs submitted for publication. The suggestion of publishing more programs each month would necessitate reducing the other content in the magazine (not desirable) or increasing the magazine's size (limited by cost). This subject has been mentioned from time to time in the magazine and most regular readers would be aware of the delays involved. To provide program authors greater flexibility, our exclusive rights to publish programs are now valid for a period of only twelve months (previously indefinite). At the end of this time, if the program has not yet been published, both the program author(s) and MICRO-80 have the option of extending the period further or terminating the agreement. We have many readers of differing tastes and try to provide something for all. Therefore, the amount of time a program spends in waiting for publication is determined by its nature, quality and age. I refer you to the piece by our Software Editor, Charlie Bartlett, in the October '82 edition for further explanation. - Ed.)

From: Howard Lee - Armadale, W.A.

I have just received the latest issue of MICRO-80 Vol.3 Issue 12 July 1983 and this is the first time that I have been driven to sit down and write to you about the standard of material, articles, and software in your publication. In short, it is of a disgracefully low standard and has been getting worse culminating in this latest issue.

It is unbelievable that the magazine reviews an astrology program which has been on the market since Noah was a boy when there is so much good new material being written for our computers. Exactly how many people do you think need a program to play anagrams? Well, let me tell you, it won't be many; the same applies to E=MC2 and a Horse Performance Guide.

I think that you are forgetting a basic rule in retailing any product and that is that you have to sell your item and if your magazine carries on in this inane and sorry fashion, it will eventually lose all of its readers. I have many friends and associates who subscribe to MICRO-80 and I can say for certain that if you do not improve very soon, you will lose a lot of subscriptions.

I know that the standard answer to complaints of my type is to blame a lack of good articles and to offer to the writer that he or she should contribute something to the magazine. I feel that an answer of that type is beneath you. Anyone who subscribes to MICRO-80 does so because he or she expects it to produce a certain standard of articles - if it fails to do this, it is not worth having.

(By its omission, I take it that the Golf program met with your approval. Believe it or not, we welcome all comments from our readers and hold the view that everyone is entitled to his/her own opinion. - Ed.)

From: W. Sweet - Morwell, Vic.

I would like to take issue with your reviewer regarding Pyramid 2000. I have this program and it gathers dust. Your reviewer implies that this is a lack of ingenuity. This seems overbearing. If you can play chess then surely a game program should tax you about the same, not be a trial and error thing. If you need a pencil and paper, then all you really have is an old fashioned board game with the rule book and dice hidden in the computer. And remember, we pay cold cash for these things. It also seems to me that the inventors of this and similar programs overvalue their genius. You can buy a good record or book for much less. And writing a book takes talent to do it well. The prices of software invite piracy.

Perhaps for us who lack ingenuity, someone could publish the first few moves of Asylum and Pyramid, just to get us going.

It is dangerous for enthusiasts to assume that people will agree with the status quo. Computers have made a very small market penetration (1.4 per school) compared with other innovations. Complexity for its own sake is no merit.

- 0000000000 -

DEPARTMENTS

***** KALEIDOSCOPE *****

After a little thinking and some trial and error, our Software Editor came up with relatively simple solutions to make two of the longer programs we have published work on a 16K system. For 'Sinking the Enemy Navy', you need only reduce the number of graphics pages allocated to one, before loading and running the program. This is done by typing: PCLEAR 1 (enter). The 'Star Trek' program requires the same step and as well the instructions at the end of the program (lines 112Ø to 137Ø) should be deleted or just not typed in at all. Line 111Ø should then be changed to:

```
111Ø GOTO 9Ø
```

In the past we have neglected to mention that the printer on which the Colour Computer listings are produced does not print the up-arrow character that is displayed on the screen but instead, prints the left square bracket symbol ([). Whenever you come across this symbol in a printed listing, you must type the up-arrow key on the keyboard. This happens with all printers that use the standard ASCII character set which assigns the left-square bracked symbol to the value 91 (instead of the up-arrow that Tandy use).

Our two programs this month are also the work of our Software Editor - Boxing (one of his first efforts) and a Hi-Res Screen Score subroutine that you can incorporate into your own programs. This demonstrates one approach to producing text on the hi-res graphics screen that you may adapt to other purposes.

THE SPEED STORY

We, as well as other magazines covering the Colour Computer have published "Speed-up POKES", but without much explanation. After a close study of the relevant application notes, the full story proves to be both enlightening and interesting.

In the Colour Computer, a Synchronous Address Multiplexer (SAM) chip is used to control the memory of the system and provide other basic hardware functions such as controlling the processor clock rate. The SAM has two bits (designated RO and R1) in its control register that determine the RATEMODE, which may be one of SLOW, Address Dependent or FAST. In the SLOW mode, the operating frequency is equal to the crystal frequency divided by 16 (or about 0.9MHz). In the Address Dependent mode, the operating frequency remains the same for RAM accesses but doubles for ROM accesses and internal processor operations (the crystal frequency is divided by 8 giving about 1.8MHz). In the FAST Mode, all operations are at the 1.8MHz clock frequency.

However, there is a catch to using the FAST mode. The SAM also controls the Video Display Generator's (VDG) access to display memory and carries out the refreshing of dynamic RAMs (necessary if they are not to lose the information that they store). In the FAST mode neither VDG display nor RAM refresh are available. You may be able to tolerate a messy or absent screen display but unless the RAMs are periodically refreshed all your data (including BASIC and machine language programs) will be lost. Secondly, since all input and output routines are in ROM and almost certainly contain time-dependent program loops, all I/O operations in both Address Dependent and FAST modes will be unreliable. This means that you can make your programs run faster provided you use the faster rate modes with care and follow a few simple rules:

1. All input/output must be in SLOW Mode.

2. The FAST Mode is perhaps avoided altogether since using it reliably for a portion of a program depends upon the length of real time for which that portion will be executed.

The SAM control register itself is accessed in a rather interesting way. Each bit in the control register is assigned two unique consecutive memory addresses and writing to the even address clears the bit (bit = Ø) while writing to the odd address sets the bit (bit = 1). The following table summarises how the rate mode bits are accessed:

Address (decimal)	Address (hexadecimal)	Action on Write
65494	$FFD6	RØ = Ø
65495	$FFD7	RØ = 1
65496	$FFD8	R1 = Ø
65497	$FFD9	R1 = 1

The data value written to these addresses is totally ignored and can be anything in the range Ø-255.

The rate mode selection is given below:

RATE MODE	RØ	R1
SLOW	Ø	Ø
Address dependent	1	Ø
FAST	Ø	1
FAST	1	1

The rate mode may be freely changed with the following exception:

Changing from FAST mode directly to SLOW mode is not allowed except by a hardware reset.

The following is a suggestion on how to make use of the rate modes to speed up a BASIC program:

```
POKE65495,Ø : REM SELECT A.D.MODE
Print titles to screen, etc.
POKE65494,Ø : REM GO BACK TO SLOW FOR I/O
Read Data File from cassette, etc.
POKE65495,Ø : REM BACK TO A.D.MODE
Perform data manipulation and calculations, etc.
POKE65494,Ø : REM GO SLOW FOR PRINTING.
Print results.  Write new file, etc.
END : REM LEAVE IT IN SLOW.
```

If you want to experiment with the FAST mode, try this:

```
        POKE65497,Ø : REM GO TO FAST MODE
        Do what has to be done.
        Don't spend too much time here or everything will go away.
        POKE65495,Ø ; REM MAKE SURE WE PASS BACK TO A.D.MODE
        POKE65496,Ø ; REM EXIT FAST TO A.D.
        POKE65494,Ø : REM BACK TO SLOW
```

With a little patience and by trial and error, you can probably put these suggestions to good use in your own programs. We would be happy to hear about any discoveries you may make and look forward to publishing some speedy programs.

- 0000000000 -

***** PEACH BOWL *****

For our Peach readers this month we present two games of skill - Othello and Block - both written by a Peach owner. Othello presents a contest of strategy for two players and is based on the board game of the same name, whilst Block involves a combination of manual dexterity, strategy and some element of chance.

Before you turn to the back to type in the Othello program, I must point out that an error appears in the listing in lines 44Ø and 45Ø. As the author points out in his description, these lines contain the graphic block character or CHR$(254) which cannot be entered from the keyboard. Alas, our printer cannot print this character either, but behaves rather strangely as you will notice if you refer to the listing. As a suggestion, use the asterisk character instead and enter in the two lines in question as follows:

```
44Ø LOCATE 4,22 : PRINT CHR$(26);"*"; : GOTO 46Ø
45Ø LOCATE 1,22 : PRINT CHR$(26);:LOCATE 76,22:PRINT"*";
```

While on the subject of errors, some readers have reported problems with the 'Sinking the Enemy Navy' program published in the July '83 issue. The statement at the end of line 2Ø should read 'SCREEN Ø,,1' with two commas and not one as published.

Also of special interest to our Peach readers in this month's issue, we feature a review of Hi-Writer, a word processing program for the Peach.

- 0000000000 -

***** GROUP ONE *****

For our Level 1 users this month, Galactic Battles and for Level 2, Hamburger, a System Tape Maker utility and XUSR Screen Fill Subroutine. For those of you who have the MICRO-80 Stickeroo Joystick, Hamburger has been modified to accept either joystick or keyboard input. To see how it's done, take a look at the subroutine at the end of the program. The keyboard is read first in the PEEK statement. If a key is being pressed, the routine returns with a value, otherwise it scans the joystick port. If the joystick is not being used or not present, a zero is returned otherwise a direction value equal to the key value for that direction is returned.

32-CHARACTER MODE IN THE SYSTEM-80

As a System-80 owner myself, I was not aware for a long time that the "PRINT CHR$(23);" on my machine produced a less impressive result than on the Tandy Model 1. On a System-80, what you get is normal-sized characters that are double-spaced on the screen, whereas on a Model 1 you get double-sized characters. The double-sized characters on the System-80 are not accessible under software control. Way back in the August and September, 1980 issues, we published a hardware modification developed by Eddy Paay that would allow software to access double-sized characters. However, this did not make the two compatible with respect to the 32-character mode.

The following hardware modification, contributed by Don McKenzie, produces the TRS-80 type 32-character mode on the System-80.

Components required: NIL

All wiring is done on interface card.

1.	Cut Z29/11	Connect Z29/11 to GND.
2.	Cut Z29/2	Connect Z29/2 to Z29/3

3. Cut Z29/14 Connect Z29/14 to Z29/13
4. Cut Z29/5 Connect Z29/5 to Z29/6
5. Cut Z37/14 Connect Z37/14 to Z37/13
6. Cut Z37/11 Connect Z37/11 to Z37/10

Remove R3 4.7K Resistor (near Q10)
Remove R29 4.7K Resistor (near Z7)

MODSEL SIGNAL

.... Connect Z6/6 to Z7/4
Connect Z7/4 to Z37/1
Connect Z6/5 to Z5/7 (To Z12/11 on boards that have 2114 Video Ram, e.g. Mark 2.)

OPTIONAL

.... Isolate S1 (video-cut switch) and page switch.
This will give you two spare switches.
These switches are not fitted in the later Mark 2, Blue label type computers.

Please Note that installation of this modification will void your warranty. This modification is not recommended for those readers with little or no experience in electronics or soldering, and is installed at your own risk.

TAPE ON DOSPLUS

A few months ago, one of our readers reported some difficulty in transferring a machine language game from cassette tape to disk. The utility in question was TAPE on DOSPLUS 3.5, which appears to be the same as that supplied with 3.4. As you may or may not know, the ROM SYSTEM tape loader is not too particular about the file name supplied. For example, if the name of the program on tape is actually "PROG1", then you can supply any portion of the filename and the loader only checks the portion given against what it finds on tape. So you could respond to the SYSTEM prompt with "P", "PR", "PRO", etc. and the filename match would be successful.

If you try this approach with the TAPE utility you will find that the filename is successfully recognized if and only if you supply the full filename (less any trailing blanks). That is, TAPE would only load the above program if you use "PROG1" as the filename. For the particular program in question, REARGUARD, the loading instructions specified using "R" as the filename and under Level 2 BASIC this worked fine. The TAPE utility refused to recognize the program and would not read it into memory. After repeating the process a few times, fiddling with the volume control, reading the manual, etc., etc. still no success. I decided to go to Disk BASIC and see what filename was actually recorded on the tape because I was running out of ideas.

You can read the filename of any machine language program on tape by running the following BASIC program:

```
1Ø CLEAR3ØØ:CMD"T":INPUT#-1,A$:PRINTMID$(A$,2,6)
```

Although the precise filename has faded from my memory, let's say the result in this case was:

```
REAR
```

Returning to DOS and the TAPE utility and trying this filename produced the desired results - the program was read into memory and then saved to disk.

NEWDOS AND LOWER-CASE

While investigating the lower-case question last month, I made an interesting discovery regarding NEWDOS8Ø. I used LMOFFSET to transfer the lower-case driver routines to the disk files LCD/CMD and LCDB/CMD with no offset so that they would load to Screen RAM. Attempting to load or execute either driver while bit 6 in Video RAM was not fully functional resulted in part of the program being loaded, then an abrupt halt and the message:

```
TRIED TO LOAD READ ONLY MEMORY
```

After modifying the hardware so that Video RAM was just like ordinary RAM, the drivers would load and run as expected. The only logical conclusion is that the NEWDOS8Ø system loader checks each byte as it stores it into memory and aborts the load when a byte fails to store correctly - a sensible precaution considering the possible consequences of executing a machine language program that is loaded into faulty memory. My compliments to the authors of NEWDOS8Ø for their thoroughness.

MORE ON SYSTEM-80 LOWER CASE

The System-80 went through three apparently distinct versions. First, the early Mark I black label model was released which had no volume control or VU meter, had no right arrow or Clear keys and had the up, down and left arrow keys labelled Esc, Ctrl and Back Sp respectively. Later, this model was revised to include the volume control and VU meter, right arrow and Clear keys were added and arrow key tops were used. Lastly, the Blue Label System-80 was released with all the features of the revised black label plus a built-in speaker, lower case and a monitor ROM.

Internally, however, the versions were not quite so distinct - particularly with respect to the interface boards. The early version interface board used seven 2102 (1K x 1 bit) memory chips and an upper-case only 2513 character generator chip. The Lower Case Modification Kits supplied by MICRO-80 are designed to fit these boards. These same interface boards were also used in the revised Mark I for some time. For some reason (about which we can only speculate), the later Mark I's were fitted with new interface boards that used two 2114 (1K x 4 bit) memory chips and had provision for two character generator chips (but with only one fitted) - the same 2513 or a 52116 with both upper- and lower-case. The Blue Label interface board was similar to this but only came with the 52116 character generator.

Now, if you have a later model Mark I and have been considering installing a lower-case modification, before you send off for one, take a closer look at your machine - it may be that you've had lower-case all the time and didn't know about it! Some of the later Mark I's were fitted with the new interface boards but upper-case only character generators while others had the new interface board with bother upper- and lower-case. How can you tell what's in yours?

Well, there are two ways. The first is to open up your '80 and actually take a look at exactly what's installed in your machine. The other is to run the little test program published last month in this section. If you get numbers that indicate you have funny 8-bit Video RAM and you also see lower-case characters, then you're lucky - you already have lower-case and all you need is the lower-case driver software. If you have funny 8-bit Video RAM but only upper case characters, then all you need is another character generator and the driver routine. Otherwise, you're out of luck and will need to send away for a bit (or is that 'kit'?)

Note that the MICRO-80 Lower Case Driver routines will not work with System-80's of this type (including the Blue Label) because none of them have 8-bit Video RAM in the true sense of the word. However, the new interface boards can be modified to provide true 8-bit RAM by cutting the trace from pin 6 of Z8 to pin 13 of Z5 and by linking pins 13 and 14 of Z5.

POKES FOR LEVEL 2

There are times when you may find some of the following hints useful, particularly if you are new to Level 2. These were sent in by Andrew White:

To disable the LIST command type:

```
POKE 16863,95 : POKE 16864,2Ø4 : POKE 16865,6
```

To disable the SYSTEM command type:

```
POKE 16866,195 : POKE 16867,2Ø4 : POKE 16868,6
```

To load an unknown SYSTEM tape, type:

```
POKE 16777,2Ø6 : POKE 16778,2
LOAD
```

To see if any key is being pressed use:

```
5K=PEEK(14591) : IF K=Ø THEN 5
```

(You must also disable the BREAK key to keep your program running).

To make the BREAK key like Shift @ type:

```
POKE 16396,62 : POKE 16397,96 : POKE 16398,2Ø1
```

- 0000000000 -

***** FORM THREE *****

Level 2 Model 3 users will need to modify the System Tape Maker utility as it uses the Model 1 exit to BASIC that does not work on the Model 3 (JP Ø66CH). In line 15Ø of the source listing, change the Ø666H to 1A19H and reassemble the program. Alternatively, if you are entering the program from the HEX dump listing, then change the data at the end of the first line from -

7EBØ ...C3CCØ6 CD 8Ø to 7EBØ ...C3191ACD8Ø

The other programs should require no changes.

- 0000000000 -

PROGRAMMING

***** THE MIDNIGHT PRINTER *****

by S.H. Liggins

Writing and testing programs in the small hours require development of special techniques to ensure domestic harmony - particularly where a printer is involved. When I first started the 'Little Athletics' suite of programs (MICRO-80, May 1982), I didn't even HAVE a printer, and when I did get one, it was far too noisy to run in the middle of the night.

The programs were designed so that every report was an exact copy of lines previously displayed on the screen, and two subroutines plus an initialisation routine could handle all the printing, and be tested at my convenience.

(a) Initialisation

```
VD% = 15360                             Video Address
VP% = 0                                 Variable Pointer
AD% = 0                                 Poking Address
DIM PR$(15)                             Dummy Printer array
FOR I = TO 15                           For each video line
  PR$(I) = " "                            Set up a dummy variable
  VP% = VARPTR(PR$(I))                    Gets its pointer
  POKE VP%,64                             Give it a new length of 64
  AD% = VD% + I*64                        Get the address of the video line
  POKE VP%+1,AD% AND 255                  Make the variable point to the video line
  POKE VP%+2,INT(AD%/256)
NEXT I
```

(b) Print a single line. (In my case I always use the second bottom line).

```
LPRINT PR$(14) : RETURN
```

I expanded this subroutine to include tests for the printer being on line, line counting, page overflows, etc.

(c) Print the whole screen.

```
FOR I = 0 TO 15 : LPRINT PR$(I) : NEXT I : RETURN
```

Because all the 'formatting' was done on the screen, (PRINT, USINGs, TABs, centering, etc.) I didn't need the printer at all for testing once I knew the subroutines worked. The program initially asked if a printer was to be used, and set a flag which was tested at the start of each printing subroutine.

The use of 'dummy' variables which pointed to the screen also opened up new possibilities including a solution to the perennial 'save/restore the display' problem.

1. Storing the screen in memory. DIM PS$(15).

```
FOR I = 0 TO 15 : PS$(I) = PR$(I) : NEXT I
```

2. Returning it to the screen.

```
CLS : FOR I = 0 TO 15 : PRINT PS$(I); : NEXT I
```

3. Storing the screen on disk. (This will work if there are no quotes characters on the display).

```
CLS : QU$ = CHR$(34) : OPEN "O",1,"VIDEO"
FOR I = 0 TO 15: PRINT #1,QU$;PR$(I);QU$;: NEXT I
CLOSE 1
```

4. Restoring the screen from disk.

```
CLS : OPEN "I",1,"VIDEO"
FOR I = 0 TO 15 : INPUT#1,W$: PRINT W$; : NEXT I
CLOSE 1
```

5. Storing and recovering the screen on tape would presumably be virtually the same as for disk. (I haven't tested it).

6. Rolling top half of screen down one line.

```
FOR I = 7 TO 1 STEP -1
     PRINT @ (I*64),PR$(I-1)
NEXT I " PRINT @0,CHR$(30);
```

7. Creating a run-time menu from a CMD(D:O) BASIC command (MODEL III TRSDOS or equivalent in NEWDOS).

P.S. Make your own small printer stand for less than $1.00

Materials: 2 household bricks
wall paper remnants (optional)

Method: Stand bricks on their edges (wrapped in the wallpaper if the aesthetics of your installation are important!) Stack the paper between the bricks. Sit the printer on the bricks.

- 0000000000 -

REVIEWS

We are interested in publishing reviews of commercially available hardware and software products for the computers that we support. If it's a good product our readers will appreciate knowing about it and, if it's not, they will appreciate it even more! We pay for the reviewsthat we publish in the magazine. For further information, please write to the MICRO-80 Editorial Office for a copy of our guidelines for writing reviews.

***** AMUST 80-DT PRINTER *****

A product review by Carl Cranstone

The first computer in the Cranstone household was the Level I 4K computer. This was followed by an upgrade to 16K, an upgrade to Level II, the purchase of a printer and finally Disk Drives.

The printer was satisfactory until the need for something better persuaded me to look around. The previous printer was the Tandy line-printer VII which was not appreciated by the other members of the family when lengthy printouts were in progress during their favourite TV programs!

Enter the need for a quiet printer.

Meanwhile...the green characters and flashing cursor on the Model I screen started to give me nightmares. (Only because I had seen colour computers hit the market!)

So, I decided to take the plunge and the Cranstone household will now get a LNW-80. But....I thought...the silver case of the line-printer will not match the case of the LNW-80 (white). This seemed like a good opportunity to buy a new printer as well!

The printer arrived first and hence I am writing this review.

The AMUST 80-DT printer (DT-80 for short) is sold by MICRO-80 and is virtually the same as an EPSON MX80 III. It comes well packaged in a large box weighing about 6 kilograms (which I had to carry home!). The average weight of the printer is approximately 5.3 kilograms. With the printer came a 48 page user manual, ribbon cartridge and wire paper rack.

1. THE MANUAL

The manual covers all of the functions of the printer giving examples of most. Some of the examples will not work correctly on the TRS-80 because of the LPRINT routine in the TRS-80 ROM. The ROM routine will not accept some values such as CHR$(0), but you can overcome this by POKEing the value directly to the printer at address 14312 (37E8H).

The manual is confusing! Apart from spelling errors, the manual appears to have been written by a Japanese person who doesn't know velly much English. Complete sentences are nearly total gibberish but after reading them 2 or 3 times, they make some sense. Here is a quote from the manual which tells you how to remove the top section of the printer:

> "Hold the both side of upper case and gently lift and push back towards back and upper way so the portion of case below the tractor mechanism got by the tractor blocks - to make it easy, move the right side tractor block to left as far as it can be pushed over".

Sea what I mean? Apart from the flaws, it is a good manual (better than Tandy Manuals!). It covers just about everything with heaps of printout samples, how to install paper and ribbon cartridge and more.

2. THE RIBBON CARTRIDGE

The cartridge supplied is fairly long (approximately 250mm) and is the type which looks like shiny plastic on one side and carbon paper on the other side (except it isn't carbon paper and can print over the same place more than once).

MICRO-80 supplies these cartridges (approx. $14) and for about the same price can reload the cartridge with nylon ribbon.

The ribbon cartridge supplied gives a fine clean print (no furriness) and looks almost letter quality.

3. THE WIRE PAPER RACK

The paper is fed underneath the paper rack with the printout piling up on the rack. The wire rack itself is coated with a white plastic and appears to be fairly strong. (Though I wouldn't trust it with 16K disassemblies!). The paper comes out of the printer and hits the upturned end of the rack causing the paper to fold and form a neat pile, (which is better than having it coat the living room floor!).

4. THE PRINTER

The printing method of the printer is an impact dot matrix. The size of the characters is 2.1mm (W) by 2.4mm (H) in a 7 x 8 dot matrix. The printer prints at 80cps and is quiet in the normal print mode. (The print speed slows down as low as 50cps with higher density printing). The printer is bi-directional, i.e. the print head moves from left to right and then from right to left. It also has a logic seeking feature, i.e. takes the shortest path to print information. The printer has friction feed as well as traction feed and is very easy to load with paper.

The power switch is located on the right hand side of the printer with the paper feed knob on the left. On the top right hand front side of the printer are 3 buttons and 4 LEDS (light emitting diodes - rectangular). The LEDS show Power On, Read, Paper Out and On Line. The buttons are for On Line, Form Feed and Line Feed.

The DT-80 uses a standard Centronics parallel interface with optional RS-2323C support. The data transfer rate is 4000 cps maximum.

The manual tells you how to set the internal switches which are a factory option (in other words you don't get it unless you ask for it). The internal buzzer is also a factory option.

CHARACTER CODES AND FONTS.

The printer is capable of printing up to 142 columns per line with the power on state being 80 columns.

The print types are Normal, Compressed and Double-width.

The Normal prints at 80 columns per line (c/l), the Compressed at 142 c/l and the Double-width at 40 c/l. You can print these characters in italics as well as emphasized and double-strike. With double-strike you can make the characters very dark and almost letter quality. The double-strike I believe prints the character on the same spot twice and the emphasized prints the character once and then prints it again with the dots just to the right of the first. You can also print subscripts and superscripts.

The DT-80 has an underlining feature which gives a true underline and not dashes.

A very useful feature is a backspace. You can only backspace in the printer buffer, i.e. if you print a character '0' then a backspace followed by '/' you can make slashed zeroes!

The way of changing these formats of print and other options is done by POKEing the values to address 14312 or via the LPRINTCHR$(command.

The control code or printer code is CHR$(27). To select the different types of print you must send this code to the printer followed by another code or codes.

The DT-80 will run EPSON MX80 III programs without any modifications (probably because the DT-80 is a look-alike).

I have included some sample printouts including a sphere which was printed in the 1280 dots/line graphics mode. The program to print it is a GRAPHTRAX program which appeared in the American publication 80-Microcomputing.

As for things like screen dumps....tough bickies...you have to write a program for it. (Hint: using some of the special characters...see samples).

SUMMARY

The DT-80 printer is A MUST (pardon the pun) and is good value for money. ($699 from MICRO-80).

If you are considering buying a printer for the first time or you are considering upgrading to a better printer then this one is well worth looking at.

I use my printer mainly for debugging programs, letter writing (you can even print on envelopes!!), and printing copies of all my valued programs.

The dimensions of the printer are 377mm (14.8") (W) by 295mm (11.6") (D) by 125 mm (4.9") (H). These measurements are the printer itself without the paper rack connected (add about 300mm to Depth).

- 0000000000 -

***** THE "HI-WRITER" WORD PROCESSING PROGRAM FOR THE HITACHI "PEACH" *****

A Review by Gordon F. Gross, D.Sc.

HI-WRITER is a wordprocessing program of considerable sophistication and flexibility. There are facilities for automatic dating, and even timing, of documents and, with the frequently updated lists of equipment, etc. which we maintain, the former is especially useful to check if you are looking at the latest printout. Other important facilities are the ability to scroll through the whole document, to see it laid out on screen as it is going to be printed with the reversible ESC F function, the ability to put text in the margin, to automatically number paragraphs and to set individual paragraphs to print wider or narrower than the rest, direct inserts from keyboard, underlining, bold printing, superscripting and subscripting, global search and replace, and shifting text from one part of a document to another and so on through quite a range of useful features. The display is most informative on a colour monitor but it is just as easy to see what is going on a monochrome display, with an appropriately modified disk.

Certain of the features offered may not be realized if you are using a printer other than the recommended Diablo. For example, we are using an Itoh F10-40P and cannot use the bold print, superscript or subscript options, although there is a way around the two latter. All of the facilities which I have employed, save one , work well. There are, however, so many of them that finding them in the manual can be a bit of a chore if you are not continually using the program. The manual is now well laid out with tagged pages to head the various sections and with a summary of the function codes at the end.

HI-WRITER is a segmented program and needs to call in overlays below its included DISK BASIC (which it loads first) for its various functions, e.g. whether to EDIT, PRINT or employ the FILER. In the FILER further overlays appear to be called in to perform various tasks like renaming files, naming or renaming the data disk, deleting files, copying files, setting the PF keys from PF5 onwards to small bits of repetitive text, etc. (The FILER is the one area not well covered in the manual although sufficiently explained on screen). HI-WRITER therefore requires two disk drives and the HI-WRITER disk must remain the whole time (except for some disk copying routines) in drive Ø and the data disk in drive 1. The data disk can be changed when exiting from EDIT (after SAVING), PRINT or FILER to the main menu as it re-examines the disk, or when going from the main menu to the FILER.

When HI-WRITER boots up after DISK BASIC has been loaded, a Cybernetics Notice appears on the screen and here you set the current date (it and the time can also be set or altered in the FILER). Then the main menu appears on screen with the disk name, number of free bytes on the disk and the names of all the files on it together with a series of options across the bottom, e.g. CREATE, EDIT, PRINT, MAIL (i.e. PRINT - a main file incorporating details of a sequence of addresses, etc. from a second file). On this, and also the PRINT and FILER menus, choices are made by obvious movements of the cursor keys and RETURN.

In EDIT you can brng up or dispose of a "TAB LINE" which has preset tabs, which you can change at will, and which indicates the number of bytes left for your document and your current format settings. As your document has to occupy a space below the ROM BASIC, DISK BASIC, HI-WRITER overlay area and a copy buffer, documents must be about 8,000 bytes or less. However, one document can be chained to another to call on the disk if you need to for continuous printing by the "@CHAIN:1: next file name" function. The latest settings on the "TAB LINE" are not remembered from one document call to the next, but they can be changed through a document anyway. However, settings for total line length, left margin and paragraph indent are recorded on disk. Default values are set on CREATing a new document and these can be altered at will whether in EDIT or PRINT and the latest settings are always valid.

No wordprocessing program, particularly in early versions, will be free from criticism or from the odd 'bug' and the following points are meant to be helpful for future releases. The present version uses the tilde accent (~) to push text to the right (e.g. the date) and this symbol cannot be output (unlike the much used caret (^)). In my opinion, CHR$(253) should have been used on PF1 to push text to the right. Like CHR$(254) which HI-WRITER uses for marking text to be searched for or shifted, it is not required to be output. In the HI-WRITER initialization PF6-10 should be programmed to produce `{|}~ none of which can be output at present. These function keys can be redefined in the FILER to repetitious pieces of text by the user if these symbols are not required. (For those requiring the first four now, go to the main menu, push the ABORT Button to go into BASIC, after CLS define the function keys as KEY 6, CHR$(96), etc. and then sneak back into HI-WRITER through the back door by LOAD "0:FILER" and RUN).

Many professional people need these extra five codes, sometimes when using the special symbols on other than ASCII daisywheels. Both in our earlier and later disks, trying to go to PRINT from the ESC X list of options in two out of five cases fails to close the file, which PRINT then refuses to print. The faulty file can be copied and the copy printed. The original can then be deleted and the copy renamed, but it is a lot of messing around.

Summing up - good marks overall but one 'bug' needing attention. There is a sorting option available as an extra but a spelling check package and a communications option for interfacing to phototypsetting equipment would be useful extras.

- 0000000000 -

***** INFINITE BASIC *****

A Review by J.F. Lamich.

This is a program produced by Racet Computers, which is designed to provide machine language routines to either replace slow BASIC routines, or to provide routines which are difficult to produce efficiently in BASIC. An additional module called "Infinite Business" is also available to provide some routines useful in a "business" environment. With the two modules, 100 routines are available.

The routines to be used for a particular BASIC program are selected from those available, saved on tape as a System module which is loaded when that program is to be run. Memory space is conserved as only those routines needed for that program are placed in memory.

The tape contains a loader, a matrix module, a string module and a file creator. There is also a disk loader to create a disk file. The general procedure is to firstly select which routines are required, load the loader program and input the routine names. Prompts are given to guide the user. The other modules are searched by the program and the required routines selected. The procedure is rather slow because it is on cassette. It takes about 15 minutes to create a file. The saving grace is that it takes the same time to create a one routine or a multi routine file.

Once the modules have been examined, the file is created and can either be saved on cassette (or disk) or run as is. The best idea is to save it so that it can be re-used. Memory size must be set to use it, and the relevant parameters are displayed on the screen (in hex) so that memory size can easily be calculated. Of course, the more routines included, the more memory used and less space available for the program. However, the advantages of speed and, often, less BASIC programming compensate for this.

Incidentally, for ESF owners, copies of the operating module can be made using the copy cassette facility on the ESF Monitor, but I have not been able to copy it onto a wafer - it seems to save parts of the Monitor as well. I'd be glad to learn the procedure if anyone manages to do it. Anyway, the operating files are relatively short and can be loaded from tape in two or three minutes.

When the file is loaded, the user types PRINT USR (1). A copyright message appears and the program is ready for use. Once it has been thus initialised, the USR address can be used for other routines if required.

The commands now available use an ampersand (&) as the instruction token, and most require a dummy argument, e.g. IR=&SRTV(S$,1,N) or J=&SDHL(0,0,25). Some of these can also be used as direct commands, e.g. the latter command will draw an horizontal line starting at 0,0 being 25 positions long. Error messages are provided. If an attempt is used to call an Infinite Business command without the module loaded, an L3 Error will result. If an incorrect command is used, an FC or SN error will result.

The following set out only some of the facilities available.

Matrix module.

This module allows arithmetic, scalar arithmetic, array copying, mathematics, input/output, shaping, and a fetch and store function for two byte integers. For example, a number of arrays can be quickly added together by copying the first array to an empty array and then accumulating to that array, e.g. suppose the user wants to add array A to array B. In BASIC this would be done like this:

```
200 FOR I = 1 TO N
210 D(I)=A(I)+B(I)+C(I)
220 NEXT I
```

This is done by omitting the loop and just copying A to D then adding B to D and C to D. The copy can be omitted if the contents of the individual arrays are not needed, by just adding B to A and then C to A. A loop is still necessary to total the columns of the array, but it will still calculate much faster than a BASIC loop.

The String module contains various string operation routines including two excellent sorts.

The program is almost worth the price for these alone. The first is used on a single multi-position array (e.g. the type of array in the Household Accounts program). The second sorts multi-arrays e.g. A$,B$,C$. Unfortunately there is no routine to sort multi-dimensioned arrays of the form A$(5,4).

The sort is made on a key, e.g. S$="A$,B$,C$", which can be defined in the program or input from the keyboard. It works in a number of formats, single ascending or descending with carries (it sorts one nominated array and carries the equivalent row in the other arrays), multiple ascending or descending with carries (if it finds equals in the first, it sorts the next nominated in the specified order and carries), or either of these without carries. The last would be rarely used as the rows would go out of order.

The sorts are very fast. A single array of 2000 names takes about 30 seconds. A set of 10 arrays with 200 elements each takes about 15 seconds. A single array with 200 names takes about 4 seconds. The sort operates using ASCII code numbers and this requires care in input as upper case and lower case codes are in different parts of the code. A shift entry on a non-lower case machine gives a lower case code but displays in upper case on the screen, and the resultant sort will look all wrong, although it will correctly follow the rules. For the same reason, a converted machine will demonstrate that the sort is in the general order - ALL CAPITALS, mixed case, all lower case. Therefore, consistency in entry of items is essential. For the same reason, it is important to specify the order in S$ in upper case.

This module also contains a number of other useful routines, e.g. to generate random strings for testing formats, to manipulate strings by copying, counting, reversing, rotating and searching.

Screen display functions.

These draw and erase vertical and horizontal (but not diagonal) lines, and also scroll the screen up, down, left and right. The latter scroll off the screen but not back on again, and are useful for creating moving bar graphs.

They only draw a vertical line from the top, so to make sense of such a graph requires inverting the screen or the operator. Horizontal bar graphs are somewhat easier to comprehend and more closely approximate bar graphs produced on printers.

The Business module allows packed decimal arithmetic, search and insert in strings, printer control, and hash code generation. To some extent, these compliment the other modules but the printer control functions are particularly useful.

The Manuals are quite good. These give a description of the function, a summary and program examples, but require quite a lot of study and experimentation to work out and utilise their potential.

For those who want to use the power of the system faster and more efficiently, particularly in Data Base Management or mathematical applications, this is a useful program to look at. Although it is rather daunting at first sight, careful study and practise should result in great benefits to the user from a very good set of programs.

- 0000000000 -

***** MUMFORD MICRO SYSTEMS SK-2 SPEED-UP MODIFICATION FOR THE TRS-80 *****

A Review by Peter R. Smith.

What is it?

The SK-2 Modification is a piece of electronic hardware which gives your TRS-80 a number of speed options:

(a) run at normal (1.77MHz) clock speed

(b) run at slow (0.88MHz) clock speed

(c) run at fast (2.55MHz) clock speed

(d) various hardware and/or software 'switches' to control the three speeds above

(e) with some additional components you MAY be able to run at a super-fast speed (3.55MHz).

What do you get?

The kit comes in a small plastic bag and consists of six pages of step-by-step instructions and diagrams, a printed circuit board with four IC's (all socket mounted), hookup wire and a double-sided adhesive foam mounting strip.

What else do you need?

Items you must supply yourself include solder, a low wattage (preferably temperature-controlled) soldering iron and a couple of uninterrupted hours. Depending on which of the options you wish to install with the kit, you may also need a small LED, one or two SPDT or SPST switches and some more hookup wire, but these are not necessary if only software speed control is required. The manufacturers have tried to keep the kit price down by providing only the essential components and letting those who want options purchase the extras separately (or find them from their 'junk boxes').

What do YOU have to do?

Firstly, and most importantly, read all the documentation - Twice. It is well written, concise and easy to follow. It is then worth reading again and giving careful consideration to the following:

(i) Do you want (or need) the slow speed (0.88MHz) option?

(This slows down LISTs and is claimed to eliminate keybounce). In my case, not having any keybounce problems and seeing no need for slower LISTs (that would still be too fast to read without the use of Shift @), I decided against this option.

(ii) Do you want visible indication of non-standard speed?

I find one of the major benefits of this modification is that programs will CLOAD and CSAVE at the fast (2.55MHz) speed with no problems. This cuts the CSAVE and CLOAD time down by one-third. (However, tapes recorded at "normal" speed will not load with the computer running fast and vice-versa). A visual indicator of non-standard speed is very useful and I decided to install this option.

(iii) Do you want hardware only switching, software only or a combination of the two?

I quickly decided that hardware switching would be essential for the CLOAD and CSAVE operations. Software switching is accomplished by POKEing into an unused bit of port 255, which is also used to control the cassette and double width (CHR$(23)) features. The software option needs the addition of two wires to the TRS-80 main PCB, the hardware only option needs a SPST switch, the combined option needs the two wire attachment and a SPDT centre-off switch.

(iv) Do you want to attempt to run at double speed?

If you do, you will lose the half speed option. The notes provided point out that there is a high probability that any particular TRS-80 will NOT WORK at double speed. Asking for a quart from a pint-pot (for those of us who haven't metricated yet) can be just too much. As a result, I would strongly recommend installing the 50% upgrade first and ensuring that it runs with no trouble before even considering the 100% up-grade. I took this approach. After a few happy days at fast speed, I tried the 100% increase. For this, you need a 74LS08 IC (Tandy Part No. 276-1908) and you must make some changes to the SK-2 board itself. These are not difficult and should the 100% speed increase not work, you have not done any damage to either the TRS-80 or the SK-2 board. As I said, I tried it but to no avail - my TRS-80 is not one of those prepared to exert itself at double rate.

(v) Do you have an expansion interface and disk drives?

If you do, there is a modification described in the notes that will automatically slow down the TRS-80 whilst disk I/O is taking place.

What are the disadvantages?

(i) If you get keybounce at normal speeds it is likely to be worse at high speed and you will need to use a software de-bounce routine.

(ii) If you have an early TRS-80 and expansion interface you may need to install some hard-wired changes (described in the notes) to improve reliability at higher speeds.

(iii) Programs that rely on the internal speed of the TRS-80 (such as Music and Sound-Generating programs) need to be run at slow speed.

What are the advantages?

(i) From my point of view the major advantage is the decreased CSAVE and CLOAD time. The hours I have saved have paid for the kit several times over.

(ii) Programs that "number-crunch" run faster, graphics displays are faster and so on.

(iii) The manufacturers claim that if you lose characters with Electric Pencil when the last word is dropped from one line to the next, then this will be reduced. My experience is with SCRIPSIT which I find can get characters wrong when it runs with the fast speed on.

Who makes it, where can you get one, and how much?

The manufacturer is Mumford Micro Systems - Box 435, Summerland, California, 93067.

I bought mine from De Forest Software - Station St., Nunawarding, Victoria.

It costs $35 which is high if you consider only the value of components in it but, in my view, a reasonable price when the development, documentation and distribution costs are taken into account.

For anyone wishing to get more time out of the TRS-80 I would certainly recommend this kit.

- 0000000000 -

***** DOUBLE PRECISION SUBROUTINES FROM TANDY *****

A Review by D. Grigg.

I was writing a program to work out the positions of the planets of the solar system at any required date (using the methods outline in an excellent little book 'Practical Astronomy With Your Calculator' by Peter Duffett-Smith), when I realised that the trigonometrical subroutines built into Microsoft BASIC were nowhere near accurate enough for my purposes. Five decimal places of accuracy, if I was lucky, was all I could expect, and it wasn't good enough.

So when I saw a package called 'Double Precision Subroutines' in my local Tandy Store, I bought it. I was a little disappointed to find out, after I had taken the package home, that the routines were in BASIC. Somehow I had imagined a carefully crafted bit of machine language which would mate neatly with the Microsoft. Instead, the package includes a set of BASIC routines which you have to graft on to your programs. This is okay if you are writing a new program from scratch, but a little annoying if you want to add double precision to an existing program.

Still, the package includes a full print-out of the routines, and a good explanation of how they work. And they certainly do work, accurate to at least sixteen decimal places. The routines in the package are sine, cosine, arctangent, square root, natural logs, exponentials and a double precision value of pi.

Being in BASIC, though, these routines are relatively slow. And they are not all that easy to use. Whereas when you only want a single precision value, you might write Y=SIN(X) and have the result in a trice, using the Tandy routines for a double precision value, you would have to write Z=X:GOSUB 40400:Y=Z2 to achieve the same thing.

However, as I have said, the routines do work, and they enabled me to finish my planetary positions program to my own satisfaction. If you are developing a program that needs high accuracy, it might be worth buying this package. But then, given a little more effort, you could probably develop such routines in BASIC yourself.

SUMMARY

DOCUMENTATION	Very good
EASE OF LOADING	Fair
EASE OF USE	Moderate
COST	$14.95
VALUE FOR MONEY	Fair to poor
AVAILABLE FROM	Tandy Computer Stores.

- 0000000000 -

***** MYSTERY FUN HOUSE ADVENTURE *****

A Review by Bob Jacobs.

'Mystery Fun House' is the seventh in a series of programs by Scott Adams, and is similar to previous 'Adventures' in that the computer is your puppet controlled by two-word instructions drawn from a vocabulary of 120 words. There are no graphics, but the program describes where you are and what you see. The game is written in machine language, so although it's large and very complex, it takes only a split second to respond to your command. Another advantage is that it is difficult to cheat by listing the program to find the solution.

In 'Mystery Fun House' your task is to explore an amusement park, and recover secret plans hidden there by its owner, before the bouncer throws you out at midnight. The game is a challenge right from the start, since you must first work out how to get into the park before the adventure can really start. This had me sweating for half an hour!

The magical mirror room, maze, and shooting gallery are just some of the many rooms and passages, so it is essential to draw a map as you go along. But beware of red herrings! You can waste many hours pondering over rooms or objects that have no importance.

MICRO-80 PRODUCTS CATALOGUE

This catalogue contains a selection from the wide range of peripherals, interfaces, computers and software carried by MICRO-80 for your computer. If you don't see the item you want, contact us, we probably have it anyway!

MICRO-80 has been supplying customers throughout Australia and the Pacific region by mail-order for 2½ years. Our customers find this a simple and efficient way to do business. You may place your order by telephone or by mailing the order form from any issue of MICRO-80 magazine. Generally, it takes about one week from receipt of order until despatch. You should allow 2-3 days for your letter to reach us and 7-10 days for the parcel to reach you, making a total turnaround time of 2½-3 weeks.

WARRANTY AND SERVICE

All hardware products carry a 90 day parts and labour warranty either from the manufacturer/distributor or from MICRO-80 Pty Ltd. In many cases, warranty servicing can be arranged in your own city, otherwise goods will be repaired by our own team of technicians in our Adelaide workshops.

TRADE-INS AND TERMS

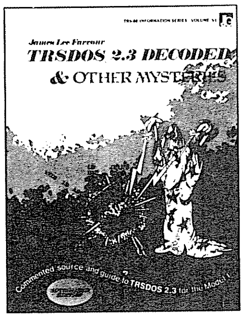
MICRO-80 can accept your existing equipment as a trade-in on new equipment. We can also arrange consumer mortgage financing or leasing on larger hardware purchases. Contact us for details.

BOOKS

TRSDOS 2.3 DECODED AND OTHER MYSTERIES

by James Farnour

The TRSDOS operating system explained with a fully commented source code listing, a description of the command line interpreter and the subroutines used by the DOS commands, the file management system, the error message processor and much, much more. Excellent value at $39.95 (plus $1.20 p. & p.)



BASIC FASTER AND BETTER AND OTHER MYSTERIES

by Lewis Rosenfelder

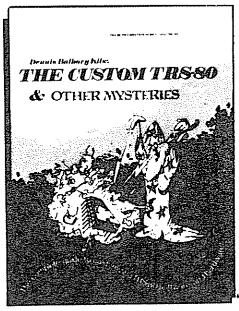
Basic is not nearly as slow as most programmers think. *Basic Faster and Better* shows you how to super charge your BASIC with almost 300 pages of fast, functions and subroutines. You won't find any trivial poorly designed "check-book balancing" programs in this book — it's packed with *useful* programs. Tutorial for the beginner, instructive for the advanced, and invaluable for the professional, this book doesn't just talk . . . it shows how! Basic Faster and Better is $39.95 (plus $1.20 p. & p.)

BOOKS

THE CUSTOM TRS-80 AND OTHER MYSTERIES

by Dennis Bathory Kitsz

Ever wanted to do things to your TRS-80 that Radio Shack said couldn't be done? How about reverse video, high resolution graphics, and audible keystrokes? Now enough? How about turning an 8-track into a mass storage device, making music, controlling a synthesiser, individual reverse characters, and a real-time clock just to name a few? The *Custom TRS-80 and Other Mysteries* is packed with more than 290 pages of practical information and can be yours for only $39.95 (plus $1.20 p. & p.)



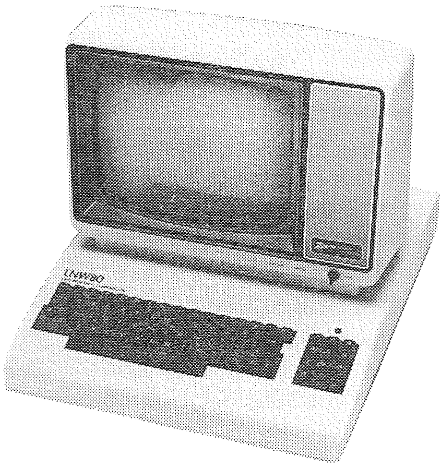
MACHINE LANGUAGE DISK I/O and OTHER MYSTERIES

by Michael Wagner

The guide to machine language disk software for TRS-80 Models 1 and 3 that explains what the floppy drive system is all about, disk controller chips, read/write access, head step and seek commands, forced-interrupt command, disk files, how records are stored on disk, creating a file, etc. You also get a disk formatter program, a program to calculate passwords and much more for only $39.95 (plus $1.20 p. & p.)

Michael J. Wagner
MACHINE LANGUAGE DISK I/O
& OTHER MYSTERIES
Machine Language Disk I/O for the TRS-80 models I and III

Introducing the NEW LNW80 MODEL 2



The new LNW Model 2 is not just a microcomputer but a complete computer package that includes excellent hardware, extensive systems software and a range of application software. Manufactured by LNW Research Corporation, the LNW Model 2 features:

HARDWARE:

- 4MHz Z80A microprocessor with 96K user RAM.
- 16K x 6 bits Graphics RAM, expandable to 64K (four pages).
- Printer, RS-232-C and cassette interfaces.
- 12K BASIC in ROM (Level 2 compatible).
- Support for single/double sided, single/double density, 5¼" or 8" disk drives.
- Full TRS-80 Model 1 compatibility.
- Hi-res Colour (RGB) and B & W video outputs.
- Four Hi-Res Graphics Modes:
  — B & W 128 x 48
  — B & W 480 x 192
  — 8 colour 128 x 192
  — 8 colour 480 x 192
- Three text displays Modes:
  — 64 char x 16 lines
  — 80 char x 16 lines
  — 80 char x 24 lines

SYSTEMS SOFTWARE:

LNWBASIC 4.0 — an extension to disk BASIC that allows full use of the LNW80's graphics capabilities through the addition of powerful graphics commands such as CIRCLE, COLOR, DRAW, etc.

DOSPLUS 3.4 — the fast, reliable and easy-to-use operating system that provides all the file control and disk management you need for maximum benefit from your disk drives as well as an enhanced disk BASIC.

LNW-CP/M — the CP/M operating system opens the door to a whole new world of software. LNW-CP/M was designed to be compatible with application programs written for CP/M 2.2 and provides the user with a 61K system

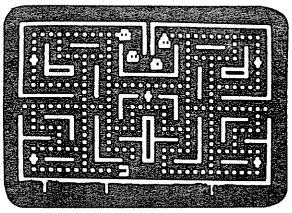
APPLICATION SOFTWARE:

1. LNW Small Business and Professional Accounting Series — *including General Ledger, Accounts Receivable, Accounts Payable and Payroll (U.S.A. conventions).*
2. Electric Spreadsheet — *for financial planning and forecasting.*
3. Electric Pencil — *an easy-to-use word processor.*
4. Microterm — *an intelligent terminal program for communications.*
5. Chart-Ex — *allows you to transform your data into pie, line or bar charts on hi-res display or printer with bit graphics.*

The LNW80 Model 2 is perfect for the serious hobbyist or businessman seeking a higher performance, more reliable computer to replace his TRS-80 Model 1 without sacrificing a huge investment in software and programming experience.

LNW80 Model 2 computer	$2,850
(complete except for disk drives and monitor)	
HI-RES Green Phosphor Monitor	$265
Super HI-RES Hitachi RGB Monitor	$1,250
Two Single-sided 40 Track Double Density Disk Drives	$825
(in cabinet with power supply and cable)	

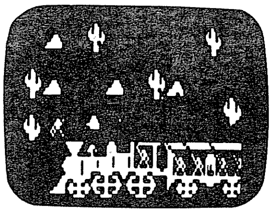
All prices include Sales Tax and are subject to change without notice. Prices are FOB Adelaide. Add $20 road freight anywhere in Australia. All equipment carries MICRO-80's Australia-wide 90 day warranty covering parts and labour.



SCARFMAN

This incredibly popular game craze now runs on your TRS-80! It's eat or be eaten. You run Scarfman around the maze, gobbling up everything in your path. Try to eat it all before nasty monsters devour you. Excellent high speed machine language action game from the Cornsoft Group. With sound.

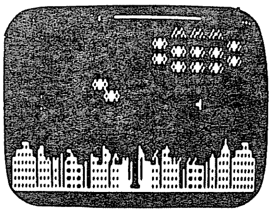
Price: $17.95



THE WILD WEST

It's up to you to keep the West beautiful with Outlaws and renegade Indians on all sides. Even the train has been captured by Outlaws with all the payroll on board. Can you clean up the Wild West?

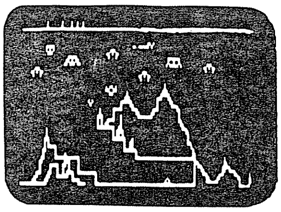
Price: $26.50



SPACE ATTACK

Steady your nerves, keep a sharp lookout, and prepare for battle to save your city. Fiendish aliens are all around, and if they destroy the city you lose.

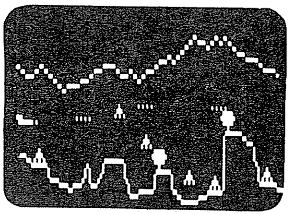
Price: $26.50



STRIKE FORCE

As the primary defender of a world of cities under deadly alien attack, your weaponry is the latest rapid fire missiles, long range radar, and incendiary ''star shells.'' Your force field can absorb only a limited number of impacts. A complex game of strategy, skill and reflexes from Melbourne House.

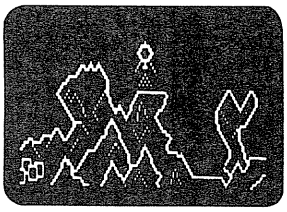
Price: $26.50



PENETRATOR

Soar swiftly over jagged landscape, swooping high and low to avoid obstacles and enemy missiles attacks. With miles of wild terrain and tunnels to penetrate, you're well armed with bombs and multiple forward missile capability. From Melbourne House. Features sound, trainer mode and customizing program.

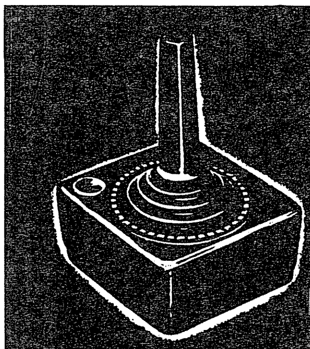
Price: $36.50



LUNAR LANDER

As a vast panoramic moonscape scrolls by, select one of many landing sights. The more perilous the spot, the more points scored -- if you land safely. You control LEM main engines and side thrusters. One of the best uses of TRS-80 graphics we have ever seen. From Adventure International. With sound.

Price: $26.50



STICKEROO JOYSTICK INTERFACE

for the TRS-80 MODEL I and SYSTEM 80

FROM **$32.00**

ADD $2.00 p. & p.

CONVERT YOUR COMPUTER INTO AN ARCADE GAMES MACHINE

Micro-80's Stickeroo Interface Features:

•Compatible with Joysticks for Atari, Vic-20 and most video games •Saves your keyboard from abuse •Compatible with programs from leading US software houses: Big Five, Cornsoft, Melbourne House, Adventure International •Adds a whole new dimension of pleasure and fun to your favourite games •Will be supported in MICRO-80 •Can be used with your own basic or ML Programs •Comes complete, ready to plug in and use •Absolutely no modifications required to your computer

Due to popular demand, Stickeroo Interface is now available separately so you can use the Joystick of your choice.

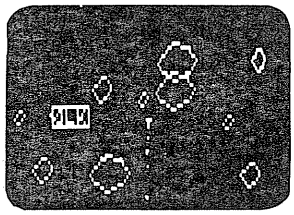
PRICE INCLUDES ... STICKEROO + INSTRUCTIONS + DEMO PROGRAM LISTING
PLEASE SPECIFY TRS-80 MODEL I OR SYSTEM 80 WHEN ORDERING

The Stickeroo Interface plugs in to the expansion edge connector and may not be suitable for expanded systems.

PISTOL GRIP JOYSTICK WITH FIRE BUTTON

$25 + $2 p & p (No p & p required if ordered with Stickeroo Interface)

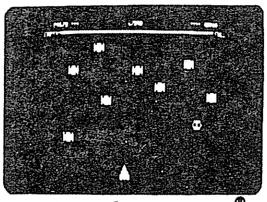
ALL GAMES ADVERTISED ON THIS PAGE ARE STICKEROO COMPATIBLE



SUPER NOVA

Asteroids float ominously around the screen. You must destroy the asteroids before they destroy you! (Big asteroids break into little ones). Your ship will respond to thrust, rotate, hyperspace and fire. Watch out for that saucer with the laser! As reviewed in May 1981 Byte Magazine.

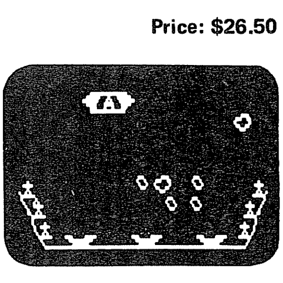
Price: $26.50



COSMIC FIGHTER®

Your ship comes out of hyperspace under a convoy of aliens. You destroy every one. But another set appears. These seem more intelligent. You eliminate them too. Your fuel supply is diminishing. You must destroy two more sets before you can dock. The space station is now on your scanner... With sound!

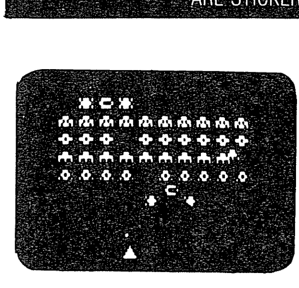
Price: $26.50



METEOR MISSION II

As you look down on your view, astronauts cry out for rescue. You must maneuver through the asteroids and meteors. (Can you get back to the space station?) Fire lasers to destroy the asteroids, but watch out, there could be an alien Flagship lurking. Includes sound effects!

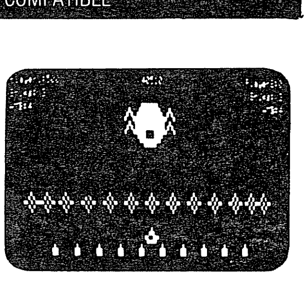
Price: $26.50



GALAXY INVASION

The sound of the klaxon is calling you! Invaders have been spotted warping toward Earth. You shift right and left as you fire your lasers. A few break formation and fly straight at you! You place your finger on the fire button knowing that this shot must connect! With sound effects!

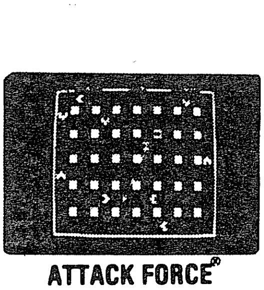
Price: $26.50



DEFENSE COMMAND

The invaders are back! Alone, you defend the all important nuclear fuel canisters from the repeated attacks of thieving aliens, repeatedly. An alien passes your guard, snatches a canister and flys straight off. Quick! You have one last chance to blast him from the sky! With sound and voice.

Price: $26.50



ATTACK FORCE®

As your ship appears on the bottom of the maze, eight alien ships appear on the top, all traveling directly at you! You move toward them and fire missiles. But the more aliens you destroy, the faster the remaining ones become. If you get too good you must endure the ''Flagship''... With sound effects!

Price: $26.50

FOR YOUR ENTERTAINMENT

MICRO-80 now offers you the widest range possible in entertainment software. These programs are supplied on cassette for the Level II/16K TRS-80 Model I/III (except as noted). They are also suitable for the System 80 but sound may not be available unless a hardware modification has been fitted to reverse the roles of recorders #1 and #2.

DEFENCE PENETRATOR $20.95

DEFENCE PENETRATOR is based on one of the most popular arcade favourites of all time with smooth graphics and sound effects. With realistic scrolling planetscape it's the best game yet.

BATTLE STATION $21.50

The aim of the game is to defend your space station against the attack of four alien space ships.

MORGOTH $20.95

Morgoth is a unique action packed adventure allowing you to wander through the enchanted dominion of Morgoth and collect the lost treasures of KAZARD KALLAHAN. But Beware! You must escape before the satanic Morgoth is aroused and seeks yea!

KILLER BEETLES $21.50

The aim of the game is to dig traps. When a beetle falls in you must fill it in to bury them, before they can catch you.

STAR CRESTA $20.95

Star Cresta takes you beyond the limits of your computer and into the Cosmic void itself! Beware! Iron clad concentration and lightning relfexes are required to destroy the evil empress.

JUNGLE RAIDERS $21.50

The aim of the game is to defend your four bases from the marauding Jungle Raiders. You kill all the Jungle Raiders and they try to hit you with their spears or drag off all four of your bases.

KILLER GORILLA $21.50

Four completely different frames. Each one offering a different challenge, makes this one of the most complex and stimulating games ever written for a TRS-80. The game keeps track of the top ten scores along with a six character name for each score.

JUNGLE BOY $21.50

The ultimate challenge! Are your reflexes fast enough to swing Jungle Boy from vine to vine? Can you swing through the jungle? Can you swim by the alligators? These are just some of the things you will find very challenging in Jungle Boy.

STELLAR WARP $20.95

Animation with superior fighter craft brings you an even greater challenge. As your computer advances your level, the aliens become more dangerous and the harder it is to stay alive!

HOPPY $21.50

The aim of the game is to get your frogs across the busy highway without being squashed and then across the river by means of floating logs and turtles.

INSECT FRENZY $21.50

The aim is to stop the centipede from getting you, all the time keeping an eye out for the giant spider.

ALIEN CRESTA $21.50

The aim is to defend your ship from numerous attacks from an assortment of aliens. If you get hit three times, it's all over.

RALLY RACER $20.95

Drive through an action packed maze and try to hit all the flags before Morgan the Mad motorist or Crazy Harry and his killer hoodlums catch you!

SPECIAL OFFER

Buy any two games on this page and pay only $35 SAVE $6

NOTE:

As the prices of imported software may vary, these prices are valid for current stock only and prices are subject to change without notice.

Double Your Disk Storage Capacity with the

LNDoubler 5/8

The LNDoubler is easily installed into your expansion interface and provides support for both 5¼" and 8" disk drives. Completely compatible with all the major Disk Operating Systems, the LNDoubler provides technically advanced, tested and reliable double-density operation with such features as:

- Analog phase lock loop data separation.
- Precision write precompensation.
- High quality PCB will all contacts gold-plated.
- Drives 1-3 may be software selected as 5" or 8" drives and a switch is provided for drive 0.
- Supports any mix of 5" or 8" drives, single or double density, single or double sided.

The LNDoubler will inrease the formatted storage capacity of **each** 40 track single-sided drive by 80% to over 180Kbytes — for just over half the cost of one disk drive. With an 8 inch double-sided double-density disk drive, you can have over one Megabyte of online storage!

The LNDoubler 5/8 doubler with documentation is available for ...

$285 plus $2.00 p.&p.

NOTE: A special cable is required for 8" drive operation and 8" double-density operation requires a 3.55 MHz CPU speed-up modification.

now available ...

DOSPLUS 3.5

is the state of the art in Disk Operating Systems for the Model 1 and Model 3 offering an order of magnitude increase in flexibility and performance over its predecessor DOSPLUS 3.4 and yet, is easier to use and more friendly with a Help facility explaining the syntax of DOS commands. The huge manual of over 350 pages describes the system in detail and is sectioned and tagged so that you can find what you want more quickly. Far greater flexibility is offered by the introduction of device drivers that are external to the system and that can be tailored to your needs.

Some of the features offered by DOSPLUS 3.5 are ...

- Single and double density support with density recognition.
- Improved file control facilities and date stamping of files.
- A keyboard driver that offers single key entry.
- An extensive Job Control Language.
- Complete and detailed technical system information.
- Two versions of BASIC, plus a BASIC label facility.
- and much more.

DOSPLUS 3.5 REPRESENTS EXCELLENT VALUE AT $160

When ordering by MAIL please specify Model 1 or Model 3 and include $2 for freight

DISK DRIVE HEADS CLEANING DISKETTES $44 incl p. & p.

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

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

5" FLOPPY DISKETTES 3M Scotch Brand

- Reinforced centre hole
- To suit soft sector format
- Rated for double density

Box of 10 single-sided for $59 incl p. & p.

Box of 10 double-sided for $69 incl p. & p.

FIT A PERCOM DATA SEPARATOR and eliminate CRC errors and locked-out tracks $45 plus $2 p. & p.

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

HARDWARE KITS

Are you the do-it-yourself type? Then these kits are for you

PCG-80 Hiresolution Graphics Board

The ultimate hi-resolution graphics board for the TRS-80 Model 1 and System 80 that provides 256 programmable characters as well as the ability to emulate a 384 x 192 dot addressable screen. The hardware capabilities of the PCG-80 enable you to plot mathematical functions, display arcs, circles, lines, waves and design and display custom characters ranging from 72 pixels to 18432 pixels. You can mix graphics and characters anywhere on the screen.

PCG-80 to suit Model 1 only **$199**
plus $5 p. & p.
PCG-80 to suit System 80 only **$219**
plus $5 p. & p.

(Note: Not suitable for later type Japanese keyboards on TRS-80 Model 1).

Typewriter Interface Kit

The MICRO-80 interface does not interfere with the normal operation of your Praxis 35 or ET-121 typewriter which becomes a reliable, high qulaity, letter perfect printer. We are convinced that you will be able to fit the interface yourself without hassles and be printing within the hour. Our interfaces provide a Centronics compatible parallel interface and are supplied with power supply, comprehensive fitting instructions and user manual. The cable to connect the interface to the computer is not included in the price. We also offer a fitting service in Adelaide for only $35.

Inteface to suit Praxis 35 only **$300**
plus $10 p. & p.
Interface to suit ET-121 only **$375**
plus $10 p. & p.

16K Memory Upgrade Kit ONLY $25

plus $2.00 p. & p.

Large volume means we can buy better and can pass the savings on to you. There are our proven, prime, branded 200 nanosecond chips, guaranteed for 12 months. A pair of DIP shunts is also required to upgrade 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 install them.

Lower Case Modification $49

plus $2.00 p. & p.

The MICRO-80 modification features true below-the-line descenders, a block cursor and symbols for the 4 playing-card suits. Each kit comes with comprehensive fitting instructions and two universal lower-case driver routines on cassette to enable you to display lower case. These routines are self-relocating, self-protecting and will co-reside with other machine language programs (the second includes keyboard-debounce and flashing cursor). Fitting requires soldering inside the computer and should only be carried out by an experienced hobbyist or technician. A fitting service is available in capital cities for only $20.00 and a list of installers is included with each kit. (Specify TRS-80 Model I or System 80 when ordering.)

All prices include Sales Tax and are correct at time of publication but are subject to changes without notice. All equipment carries MICRO-80's Australia-wide 90-day warranty covering parts and labour.

DISK OPERATING SYSTEMS & DEVELOPMENT SOFTWARE

You can increase your programming productivity, the execution speed and 'user friendliness' of your programs by using an enhanced Disk Operating System (DOS). Together with the other utility software, you can get the most from your disk drives. Note: For DOSes, include $2.00 for freight.

DOSPLUS 3.5 **$160.00**
(Specify Model I or Model III)

DOSPLUS 3.5 is a powerful, sophisticated DOS intended for the experienced user. The system can be configured to suit your requirements, provides greatly enhanced features over 3.4 and new features like single-key entry, date-stamping of files, a Help file and more. More user friendly than 3.4, DOSPLUS 3.5 comes with a very extensive stand-alone manual.

BMON by Edwin Paay **$20.50**
The ultimate High Memory Basic Monitor L2/16-48K

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?

THE FLOPPY DOCTOR/MEMORY DIAGNOSTIC
Model III Disk **$43.50**

THE MICRO CLINIC offers two programs designed to thoroughly check out the two most trouble-prone sections of the TRS-80 – the disk system (controller and drives) and the memory arrays. Both programs are written in Z80 machine code and are supplied together on diskette for a minimum 32K, one disk system.

NEWDOS 80 VERSION 2.0 **$185.00**
(Specify Model I or Model III)

Newdos 80 suits the experienced user who has already used TRSDOS, understands the manual and is prepared to learn the somewhat complicated syntax of one of the most powerful DOS's available. With the correct hardware, Newdos 80 supports any mix of single- or double-sided, single or double density, 5" or 8" disk drives with track counts up to 96. It provides powerful, flexible file handling in BASIC including variable length records up to 4096 bytes. Definitely not for the beginner.

MASTER DISK DIRECTORY **$20.95**

FIND THE PROGRAM FAST!! PAYS FOR ITSELF BY RELEASING REDUNDANT DISK SPACE!! MASTER DIRECTORY records the directories of all your individual disks onto one directory disk. Then it allows you examine them, find an individual file quickly, list files alphabetically, weed out redundant files, identify disks with free space, list files by extension, etc., etc. This program is invaluable for the serious disk user and will pay for itself many times over. Not fully compatible with NEWDOS 80.

COLOUR COMPUTER SOFTWARE

For Tandy Colour Computers with at least 16K of memory, choose from the following selection:

SPACE FIGHTER **$20.95**

You are out in space, armed with laser beams and your mission is to shoot down five enemy fighters. Skill is required to keep your enemy in your sights to destroy them. Requires 16K Extended Colour BASIC.

ESCAPE FROM MARS **$20.95**

A good first adventure that takes place on Mars where you have to explore the city and deal with the possibility of hostile aliens.

BREAKAWAY **$20.95**

Written in machine language for speed and features 15 levels of difficulty.

QUEST **$25.50**

An adventure played on a map of Alesia generated by your computer. You must gather men and supplies by combat, exploration, trading, etc. When your force is strong enough, you attack the citadel in a battle to the finish.

RACING DRIVER **$23.50**

You drive along able to see only the road in front of you, leaving little time to avoid obstacles and hazards that you meet in your path. Featuring three levels of skill, colour and sound. Requires 16K Extended Colour BASIC.

HAUNTED HOUSE **$20.95**

A real adventure for the kids, with ghosts and goblins, treasures and problems.

TREK ADVENTURE **$20.95**

You are in deep trouble aboard a starship abandoned by its crew.

PYRAMID **$20.95**

An exciting and challenging adventure where you hunt for treasure in a pyrammid full of problems.

80 COLUMN DOT MATRIX PRINTER

Features:
- 80 cps bi-directional, logic seeking
- 40, 71, 80 or 142 characters per line
- Normal and italic alphanumeric, symbol and semi-graphic characters
- Unidirectional bit image graphics (8 x 640 or 8 x 1280 dot/line)
- Tractor and friction feed

EPSON MX80 compatible control codes

FOR A LOW $599

SPECIAL ANNOUNCEMENT

We are expanding our range of disk drives to include the new

SLIMLINE

MINI-FLOPPY DISK DRIVES

These half-height, 5¼" disk drives represent the state of the art in both technological design and mechanical construction. With the characteric high quality of manufacture expected in Japanese products, these drives feature ultra-modern electronics, servo-controlled direct drive motors and exceptional physical construction that provide extremely reliable, smooth and quiet operation in both single and double density. Two varieties will be available shortly: 40 track double sided and 80 track double sided.

Please WRITE for further information.

OTHER PRINTERS AVAILABLE:

EPSON RX-80 $995

Features: 100 cps, 6 character sizes, bit image and graphic modes.

ITOH PROWRITER 8510 $1150

Features: 120 cps, bit graphics and proportional printing.

EPSON FX-80 $1399

Features: 160 cps, 6 character sizes, proportional printing, bit graphics.

All prices include Sales Tax and are correct at time of publication but are subject to change without notice.
All equipment carries MICRO-80's Australia-wide 90 day warranty covering parts and labour.
Add $10 road freight anywhere in Australia.

If in difficulty, you can ask the computer for assistance by typing 'Help' but the only response I have had so far is 'Try examining things!'.

For a price of around £12 in Britain, the game is good value, but this depends on how quickly you can solve the puzzle - it took me a month. It's not quite so difficult as some of the other Scott Adams' 'Adventures', but it is extremely well written and has plenty of nail-biting suspense together with some pleasing use of humour.

- 0000000000 -

***** MICROBUGS *****

Although we make every effort to ensure accuracy in the material we publish, inevitably errors and omissions will occur. In this section, we print corrections to those bugs that have been reported.

MORE ARGUMENTS FOR MACHINE LANGUAGE SUBROUTINES -

August, 1983 pp. 7-8.

The following typographical errors occurred in the programs accompanying the article :

On page 7 in the program "BEEP", the second and third DATA lines should read:

```
DATA 32717,-2Ø726,1972Ø,31Ø68,-14153,-18565,16Ø72,17154,
DATA -45,-496,8253,3576,-3296,-1868Ø,456,-1,12321,23Ø4,
```

On page 8 in the 'store/restore screen display' example, the DATA line should read:

```
DATA 8,8448,Ø,4352,Ø,256,Ø,-2Ø1Ø4,-4664,-139Ø4
```

- 0000000000 -

SOFTWARE

***** HAMBURGER - L2/16K by Bill Tanti *****

The idea of this game is to make as many hamburgers as you can. You make hamburgers by walking over either a bun, piece of meat, salad or an egg. When you do this the part of hamburger will fall and if there is anything beneath it, it will fall too. The hamburgers will collect in cups below. A bonus will be awarded after three hamburgers are made and, depending on the size of the hamburgers, this will decide the size of the bonus.

There are two sausages running after you and they get faster as the game progresses. There is a shield which will protect you while parts of hamburgers are falling, and when you are in a lift. (If your score exceeds 10,000 then there is no shield in the lifts).

If you drop a piece of a hamburger on a sausage then a bonus ranging from 100 to 900 points is awarded. If your score exceeds 10,000, you also receive an extra man.

There is also a special way to enter your name after a game. The alphabet is displayed. By using either the left or right arrows you move a marker below the letters of the alphabet. When you reach the letter you need, you can register it by pressing the space bar.

- 0000000000 -

***** HIGH RESOLUTION SCREEN SCORE SUBROUTINE - Colour Computer by C. Bartlett *****

This subroutine can be used with any program that requires a score display on a high resolution screen. The user simply initializes the digit strings by - GOSUB32000 - at the start of his program, the variable "SC" should be used in the user's program to keep count of the score. The main subroutine at line 32130 will take the value found in the variable "SC" and turn it into the proper high resolution equivalent.

Line 32150 is used to blank out the old score. The statement COLOR 2 should be set to the background colour if a different colour set is chosen. In this line and line 32240 the scale function of the DRAW command has been set to 8. The scale can be set as low as 4 in PMODEs

3 and 4 and the display will still be clear. If you change the scale of position statements in one line, you must change them in the other line as well.

The lines following this, up to line 32170, are used to convert the variable "SC" to a string and pad it on the left with the required number of zeroes.

Lines 32190 to 32220 assign each position in the string to a variable, i.e. the leftmost character is assigned to variable "B1", the following character to variable "B2", etc.

Line 32230 builds the display string and line 32240 displays it.

It's as simple as that, so as long as you use "SC" to keep score, (or change the subroutine), all you have to do to have a high resolution screen score is put the statement - GOSUB 32130 in your program each time that the score is updated.

- 0000000000 -

***** WORLD CHAMPIONSHIP BOXING - Colour Computer by C. Bartlett *****

In this simulation of a boxing match you are viewing the boxers from above. The computer controls the boxer on the right and you control the boxer on the left. Both boxers only punch with their right hands. Use the following keys to control your boxer:-

UP ARROW	- MOVE UP THE SCREEN
DOWN ARROW	- MOVE DOWN THE SCREEN
RIGHT ARROW	- PUNCH

The scores are displayed for both players at the bottom of the screen. One point is scored for a body blow and 10 points are scored for a blow to the head.

As the Boxer you are fighting is controlled by the Computer, without some interference, the Computer would win all the time. So to give you a fighting chance, (pun intended!), the Computer controlled boxer is forced to pause now and then; the length of the pause is random and will vary from being unnoticeable to about 2 or 3 seconds. Even with this advantage the Computer's boxer will beat the hell out of you if you don't keep on your toes. Play continues until you are sick of it!!

WARNING - STANDING STILL IS A HEALTH HAZARD!

- 0000000000 -

***** OTHELLO - Hitachi Peach by Trevor Mynott *****

This is a two person game which enables two players to play the game of Othello. The game also features screen scoring and illegal move checking.

The player on the left denoted by "X" pieces starts by entering the X and Y coordinates of his move. No pressing of the return key or comma is required. The player on the right is denoted by "0" and moves subsequently.

For those people unfamiliar with the rules of Othello, the following opening moves will provide some insight into how the game is played.

	X	0
Move 1	43	33
Move 2	34	53
Move 3	64	35
Move 4	24	etc.

Notice how in these moves a player moves by moving to a square which will enable him to capture at least one or more of his opponent's pieces.

The number of pieces which were taken from your opponent is displayed on your side of the screen.

The character used in lines 440 and 450 is the graphic block character or CHR$(254) and needs to be altered when the program is typed in.

After every move the score is updated and the prompt (graphic block character) moves to the other player's side, indicating that it is his/her move.

The program requires NEW ON 7 (i.e. hi-res /80 column).

- 0000000000 -

***** BLOCK - Hitachi Peach by Trevor Mynott *****

This is a two player game written for the Hitachi Peach Computer. Each player attempts to block his opponent by positioning his graphics so as to block his opponent.

The game starts by displaying a border around the screen and two separate graphics lines which move automatically upwards. The line on the left is under the left hand player's control and the line on the right is under the right hand player's control. A player must manoeuvre his line so as to avoid hitting any graphics whilst attempting to block his opponent into an impossible position so that he hits a graphics line.

The left hand player uses the following keys to change the direction of his graphics line:

W - (Up) A - (Left) S - (Right) Z - (Down)

The right hand player simply uses the four arrow keys to change his direction.

A player may move to a random position on the screen by using the following keys:

Space bar - for left hand player,
"?" key - for right hand player.

Do not try to move in a reverse direction as this will result in a win to your opponent.

Once a player runs into a graphics dot, a winning message and an updated score is displayed. The game automatically restarts after a few seconds.

The program requires 40 column, low resolution mode so the program can be run from any current mode.

- 0000000000 -

***** SAVE - L2/16K m.l. by Michael Byrne *****

Save is a machine language program which enables you to make System tapes of any block of memory (including the screen). This only applies to non-disk users as it makes use of the Level 3 'SAVE' command. The format of the command here is

```
SAVE name addrl addr2 addr3
```

where -

name	is up to 6 characters long and is the name of the tape.
addrl	is the starting address (in Hex) of the block to be saved
addr2	is the ending address (in Hex)
addr3	is the entry point (in Hex).

The last field, addr3, may be omitted from the command in which case the first address, addrl, is used. Each of the fields should be separated by at least one space. Also, addr2 must be greater than addrl.

The command can be used either by itself or as a statement within a BASIC program. This means that you could put it in to a loop, for instance, and have multiple dumps of a block of memory.

As shown, the program is for a 16K machine. However, if you have EDTASM or EDTASM+ it is a simple matter to change the ORG statement to relocate it. When you want to 'Save' first, set memory size to 32430 to protect the routine. After it has loaded, type "/" and the machine will return to the READY prompt. The SAVE command is now available to you.

Now for a run-down on how it works. When the program is first loaded execution starts at the line labelled SETUP. This merely places, in reserved RAM, a jump to the rest of the routine.

This part is executed only once.

Label SAVE marks the beginning of the main routine. It consists essentially of a series of calls to other routines to fetch the name and addresses.

The first subroutine called is GETNAM. This extracts the name from the command line and stores it at the end of the routine. SCAN is then called to skip over any intervening spaces. Next GETADR converts the Hex characters in the command line into an address and returns it in the DE register pair. If the end of line is found then "?SN ERROR" is printed. GETADR is again called and returns the second address. If the end of the line is found then the first address is used as the entry point, otherwise the third address is fetched. At this point HL is pointing to the end of the statement and so it is saved to allow a return to the BASIC interpreter.

The first address is checked to ensure that it is less than the second address otherwise "?UE ERROR" is printed. If all is correct then the length of the block of memory is calculated and the tape is written. (The SYSTEM tape format is explained in quite a few books including the Level 2 ROM Reference Manual by Eddy Paay). After the block has been written to tape, HL is restored and a return is made. The other subroutines are fairly straightforward.

SCAN is used often and merely skips over spaces.

GETNAM reads characters from the command line until either a space is encountered or 6 characters have been read. If the name is less than 6 characters long, it is padded out with spaces.

GETADR reads characters from the command line, checks that they are valid Hex characters (i.e. Ø to 9 and A to F) and then converts it into an address and returns it to the calling routine via the DE register pair.

- 000000000 -

***** XUSR SCREEN FILL SUBROUTINE - L2/16K by C. Cranstone *****

XUSR Screen Fill is a short routine that allows you to fill the screen with any valid ASCII character (0-255).

The routine is written in Assembly Language and has been incorporated into a BASIC program so that you can use it in your own programs.

The program will work in either Level II BASIC or any version of Disk BASIC. (The BASIC program calculates what you are using and makes the adjustments for you.)

I have not tried this program on the Model 3 (mainly because I don't own one!) but it might work!

The Assembly Language part of the program makes a call to the ROM at address 0A7FH. Whether the Model 3 is the same or not, I don't know (as I said, I don't own one!).

PROGRAM DESCRIPTION

BASIC program

Lines 1 to 6 are REM statements, not really part of the program.

Line 10: S$ is set as 17 spaces. This is the length of the assembly language routine.
When you assign characters to a string (S$), it loads the characters in the string storage area in memory (at the top somewhere!). After Line 10 is executed, there are 17 spaces in memory assigned to the string S$.

Line 20: The first part of line 20 is the DATA for the m.l. routine. The next part, "V=PEEK(VARPTR (S$)+2)*256+PEEK(VARPTR(S$)+1)" finds the location in memory of the string S$ (VARPTR means variable pointer). V will equal a memory address after this instruction.

The next instruction, "FORG=VTOV+16:READF:POKEG,F:NEXT" reads the DATA and POKEs it into the memory starting at V and ending at V+16 (17 DATA numbers in all).

Line 30: This line works out what you are using, i.e. Level II, Model 3 or Disk BASIC.

The instruction, "C=PEEK(16549)*256+PEEK(16548)" works out where the program begins in the memory.

The program then executes, "IFC<>17129ANDC<>17385THEN..."

The program will start at 17129 in memory if it is Level II; it will start at 17385 if it is "Level II, Model 3".

This instruction says, if it is not Level II AND it is not Model 3 THEN...

The next part of the line is a Disk BASIC command which sets the start of the m.l. routine for the X=USR(n) command - "THENDEFUSRO=V:CMD"T"ELSE..." CMD"T" turns off the clock in Disk BASIC (also key repeat, etc. in most DOS's).

The commands for setting the start of the m.l. routine for Level II "ELSEV1=INT(V/256):V2=INT(V-V1*256):POKE16527,V1:POKE16526,V2"V1=INT(V/256)" & "V2=INT(V-V1*256)" calculate the values to be POKEd into memory addresses 16526 and 16527. These addresses contain the starting point of the m.l. routine to be called via X=USR(n).

M.L. routine

```
00100          CALL          0A7FH
00110          LD            A,L
00120          LD            HL,3C00H
00130          LD            DE,3C01H
00140          LD            BC,3FFH
00150          LD            (HL),A
00160          LDIR
00170          RET
```

Line 00100: is a ROM call which gets the value of N in X=USR(N), converts it to an integer and stores the result in register pair HL.

Line 00110: Loads the Accumulator with register L of register pair HL (Since N is between 0 and 255, HL will not exceed 255 - 00FFH. Hence, you only need to Load register L which is the half of register pair HL which contains N).

Line 00120: Loads the register pair HL with the start of the screen which is 15360 (or 3C00H).

Line 00130: Loads the register pair DE with the start of the screen plus one which is (of course!) 15361 (or 3C01H).

Line 00140: Loads the register pair BC with the length of the screen which is 1023 (or 3FFH).

Line 00150: Loads the contents of register pair HL with the value stored in the accumulator. (In BASIC that's POKEHL,A where HL=15360 and A=N from X=USR(N)).

Line 00160: Loads, Increments and Repeats. (See LDIR below).

LDIR (DE)=(HL) DE=DE+1 HL=HL+1 BC=BC-1

This is the m.l. command Load, Increment and Repeat. It loads the contents of memory address DE with the contents of memory address HL, adds 1 to memory address DE, adds 1 to memory address HL and takes 1 away from the length counter.

This instruction will repeat itself until BC is equal to zero.

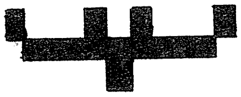
- 0000000000 -

***** GALACTIC BATTLES - L1/4K by Martin Downey *****

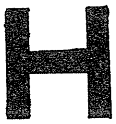
AIM: To destroy the 12 Enemy Battle Ships (E.B.S) within the time limit of 40 time UNITS.

When you RUN the program the computer will first ask: LARGE NO.?__ This is to give a RANDOM start. Enter any number.

After a short delay the computer will display the "Battle Ground". The illustrations below identify the various components. The "Battle Ground" consists of 4 quadrants one of which is shown on the screen at a time. The particular quadrant you are in is indicated by the light-block beside one of the numbers in the bottom left corner of the screen.



YOUR SHIP



ENEMY BATTLE SHIP

STAR BASE

STAR or METEOR

At the start of the game, and before each move during a game, there will be displayed 6 COMMANDS from which to choose:

```
2 1         COMP-1    PHASR-2    NAVGT-3    SHLD-4    DOCK-5    CHRG-6?_
3 4
```

You make your choice by ENTERing the corresponding number. Be careful that you ENTER the right number as any number greater than 6 will be taken as meaning 6 - you get no second chance! (Life's like that). Below are the 6 commands explained in detail.

COMP-1: Gives access to the on-board computer. If INOPERATIVE is displayed then the computer has been damaged by an Enemy phasor. DOCKing with a STARBASE may get it fixed, but don't count on it. If the computer is still operational it will display the following:

```
ENERGY= 9ØØØ  SHIELDS= Ø  TIME= 4Ø        (ENTER)?_
```

By pressing the ENTER key the rest of the information is displayed:

```
STARBASE QUADS: 3 & 4        E.B.S= 12 ?_      (ENTER)?
```

Pressing ENTER again will return you to the 6 COMMAND choice. At the start of a game you will have 9ØØØ ENERGY UNITS, your SHIELDS will be off (= Ø), you have a total of 4Ø TIME UNITS, and there are 12 E.B.S to destroy. There will be 2 STARBASES at all times although they may both be in the same quadrant. Using COMP does not use up any TIME UNITS.

PHASR-2: Used to destroy the E.B.S. The computer will first ask at what ANGLE you wish to fire. You ENTER a number from Ø to 36Ø degrees (SEE ANGLE CHART) by estimating the E.B.Ship's position.

After you have ENTERed the angle the computer will ask for the POWER. A POWER of 1ØØØ will usually destroy the E.B.S if you make a direct hit. However, like you, the Enemy has shields and may absorb your PHASOR. Small POWERs may be used as guides to help estimate the Enemy's position. POWERs of less than 1ØØ may fade before reaching the Enemy. PHASR uses 1 TIME UNIT unless PHASR is inoperative then no TIME UNITS are used. PHASR may become inoperative when ship is hit by the Enemy depending on SHIELDS. DOCKing with a STARBASE will always repair the PHASORS.

The STARBASE cannot be destroyed by your or Enemy PHASORS. PHASORS cannot pass through STARS or METEORS, they are absorbed. The POWER of your PHASOR is subtracted from your ENERGY.

NAVGT-3: Used to NAVIGATE from one quadrant to another and within any quadrant. The computer will first ask: QUADRANT?_

If you want to move to another quadrant you ENTER the number of that quadrant. You are then transported through hyperspace to the new quadrant using 1ØØØ ENERGY UNITS but no time units.

If you wish to NAVIGATE around the quadrant you are already in then ENTER your present quadrant number. The computer will then ask: GIRO-ANGLE (Ø--36Ø)?_

You then ENTER the angle you wish to travel at to get to your desired position (For DOCKing etc.). The angles are the same as for PHASR. (SEE ANGLE CHART).

After you have ENTERed the angle the computer will ask for the MAGNITUDE of your move. A MAGNITUDE of about 5Ø will get you from the top of the screen to the bottom and a MAGNITUDE of about 12Ø will move you from one edge of the screen to the other. You will have to learn from practise the correct MAGNITUDES for other angles.

You can safely move through METEORS or STARS.

Specifying a quadrant other than 1,2,3 or 4 will place you in unknown parts of the Galaxy requiring NAVIGATION back to where you are supposed to be completing your MISSION.

The amount of ENERGY used is 1Ø times the MAGNITUDE or 1ØØØ UNITS when changing quadrants. Changing quadrants uses no time units, otherwise 1 TIME UNIT is used. If after entering the MAGNITUDE you get an INOPERATIVE signal then either you entered an unacceptable number (e.g. negative) or you have insufficient ENERGY to make the journey.

SHLD-4: When you choose this command the computer will show you the present SHIELD rating and ask you to ENTER the new rating. You can set the shields to any value between Ø and 1ØØØ inclusive. The higher the rating then the less chance of damage to the ship if you are hit by the Enemy. If the ship is damaged then you may lose the use of the on-board computer or the phasors or both. However, when the ship is hit the ENERGY BANKS are drained by ½ the amount the SHIELDS are set at. The I.S.A. (International Space Administration) suggests full power on shields at all times. However, if you are short on ENERGY UNITS you may have to risk less.

Resetting the SHIELD rating uses 1 TIME UNIT. The amount allocated to the shields is subtracted from the ENERGY STORE but is not actually used up unless hit. After a hit ENERGY is automatically diverted from STORE to keep the SHIELDS at their set rating.

DOCK-5: To DOCK you must manoeuvre your space ship to within about 1 BLOCK UNIT of a STARBASE. (A BLOCK UNIT is the size of a STAR or METEOR). If you NAVIGATE over a STARBASE you will effectively blank it from the screen making your next attempt to get close enough for DOCKing more difficult. When you think you are close enough try the DOCK command. If you get an INOPERATIVE sign then you are not close enough. If you get INOPERATIVE no TIME UNITS are used up.

If you do not get an INOPERATIVE then DOCKing is complete. You must now wait until your ENERGY is replenished to 9ØØØ UNITS, and any repairs are carried out. Your damaged PHASORS will be repaired - however, your COMPUTER may not.

Docking uses 5 TIME UNITS and during this time the E.B.S will be free to have a number of shots at you. However, you will not be able to fire back while in DOCK. For this reasons the I.S.A. advises all STAR-FLEET commanders only to DOCK in quadrants free of Enemy Cruisers.

When DOCKing is completed your ship will be jettisoned from the STARBASE and you once again have command.

CHRG-6: This command enables you to CHARGE your ENERGY BANKS from the available solar radiation. CHRG uses 1 TIME UNIT and charges your ENERGY BANKS By 1ØØ UNITS. Since all usage of 1 TIME UNIT with the other commands (PHASR, NAVGT, SHLD) also charge your ENERGY by 1ØØ UNITS, this command should only be used as a last resort. The I.S.A. does not approve of its STAR-FLEET cruisers sitting idly recharging, so all commanders are advised to DOCK and re-energise before ENERGY stocks are too low.

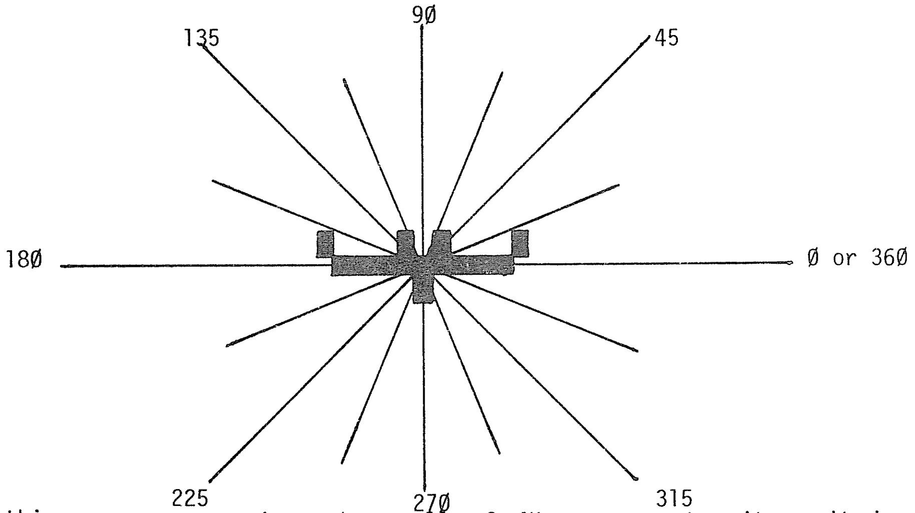
The above is only a brief outline of the operation of the STAR-FLEET cruiser you will command. How you go about completing your mission (Destroy ALL E.B.S. in the 4 quadrants you patrol before the I.S.A. convention in 4Ø TIME UNITS) is up to you. You'll need more than good luck anyway.

The Enemy Battle Ships are controlled by FIGUTRONS (Illusionary Robots) so their battle strategy and firing accuracy is unpredictable. They may even blow up their allies or even themselves. If they do, however, score a hit on your ship the signal: -SHIP HIT- will be displayed. The amount of damage resulting will depend on your SHIELD strength and the angle at which your ship is struck.

If through an INPUT error or double ENTER the screen is messed up, simply press BREAK and re-run the program from line 13Ø (i.e. ENTER the following: RUN 13Ø). This will redraw the scene with your ship randomly placed in the same quadrant. All other parameters (TIME, ENERGY...) will be unchanged.

May the FORCE always be at your back and not flying in your face!

ANGLE CHART



NOTES: Because this program uses just about all of 4K memory enter it as it is without the addition of extra spaces or expansion of the statement abbreviations (unless you have 16K RAM or larger memory). HOWEVER do not insert a space between the line number and the start of the line as this is done by the computer when LISTing the program.

The fact that this progam makes use of just about every available BYTE of memory means that multiple statement lines, abbreviations and line-wrap-around are used extensively. Although this makes the listing confusing, it does mean that you get the most out of a 4K TRS-80.

- 0000000000 -

***** WHAT YOU HAVE MISSED *****

Set out below is a list of some of the programs published in early issues of MICRO-80 magazine. Back issues are available for $2.50 each or at the annual subscription rate for 12 or more copies. Cassette editions are available for all issues for $4.00 each whilst disks are available for all issues from September 1981 onwards. For 12 or more magazines with cassettes/disks ordered at the same time, the relevant annual subscription rate applies. Programs for the Hitachi Peach/ TRS-80 Colour Computer were first published in the April 1982 issue.

ISSUE 4 - MARCH 1980	
LEARNING NIM	(L1)
LUNAR LANDER	(L1)
RICOCHET	(L2)
INVADERS	(L2)
RANDOM NUMBER GENERATOR TEST 1	(L2)
RANDOM NUMBER GENERATOR TEST 2	(L2)
ANALOGUE CLOCK	(L2)
BMON - Part 2	(L2)

ISSUE 22 - SEPTEMBER 1981	
ROVING TARGETS	(L1)
SOLITAIRE	(L2)
SEMAPHORE	(L2)
THREE BILLY GOATS GRUFF	(L2)
MOVIE	(L2)
BASIC ARRAY SAVER/LOADER	(L2)
LOWER CASE DRIVER FOR THE ESF	(L2)
SOUND EFFECTS REVISITED	(L2)

VOL.3 ISSUE 5 - APRIL 1982	
ATOMIC TABLES	(CC)
ATOMIC TABLES	(PEACH)
MULTIPLE REGRESSION ANALYSIS	(CC/PEACH)
SPACE COMMANDER	(L1)
SPACE GALAXY	(L1)
READ A LINE	(L2)
FLIP	(L2)
KEWORD	(L2)

L1 - Level 1 program CC - Colour Computer

L2 - Level 2 program HP - Hitachi Peach

The following back issues of MICRO-80 magazine are still available:

	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sept	Oct	Nov	Dec
'79	-	-	-	-	-	-	-	-	-	-	-	✓
'80	✓	✓	✓	x	✓	x	✓	x	✓	x	x	✓
'81	✓	x	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
'82	✓	✓	✓	✓	✓	✓	✓	✓	✓	x	-	-
'83	-	-	-	-	-	-	✓	✓	✓			

(- means never published, ✓ means issue available, x means issue out of print)

- 0000000000 -

***** NOTICE TO SUBSCRIBERS *****

In the interests of providing the best possible service to our subscribers, MICRO-80 is implementing a facility to deal with subscription enquiries and difficulties. Errors and omissions have occurred. We apologize for the inconvenience caused and hope that our new system will reduce the occurrence of these to the absolute minimum. We request that, in order to attend to your problem as quickly as possible, you address your queries in writing only to:

MICRO-80,
SUBSCRIPTION DEPARTMENT,
P.O. BOX 213,
GOODWOOD, S.A. 5034.

In your correspondence, please include:

1. Your name and address
2. Type of subscription and expiry issue.
3. Your most recent mailing label.
4. The nature of the query or difficulty
5. Details of payment, e.g. B/C - cheque - money order.

Changes of address should also be forwarded to the Subscription Department in writing with at least 14 days notice. Please do not ring with queries as they cannot normally be satisfactorily resolved over the telephone in a reasonable length of time.

Cassette and disk editions of the magazine are normally posted at the same time as the magazine itself but under separate cover. The magazine and cassette/disk will usually arrive within a few days of each other.

- 0000000000 -

When entering listings from the magazine please note that some characters displayed on the screen cannot be printed. For example, the up-arrow character displayed by Tandy machines is printed as "[".

**** HIGH RES SCREEN SCORE ****

COLOUR COMPUTER

```
10 ' HIGH RESOLUTION SCREEN
     SCORE SUBROUTINE
     (C) 1983  C.BARTLETT

20 CLEAR500:PMODE3,1:PCLS(2):SCR
EEN1,0:GOSUB32000
40 SC=9990
50 SC=SC+1:GOSUB32130
60 FOR T=1TO500:NEXT T:GOTO50
70 '
80 '
90 '
100 '
32000 ' SUBROUTINE TO LOAD IN
32010 ' NUMBER STRINGS

32020 N$(1)="BR4BD1E1D6L1R2BU6"
32030 N$(2)="BR4BD1E1R2F1D1G3L1D
1R4BU6"
32040 N$(3)="BR4BD1E1R2F1D1G1F1D
1G1L2H1BR5BU5"
32050 N$(4)="BR6D6U3R2L5E3BR2"
32060 N$(5)="BR5R4L4D2R3F1D2G1L2
H1BR4BU5"
32070 N$(6)="BR6L1G1D4F1R2E1U1H1
L3BR4BU3"
32080 N$(7)="BR4R4D1G2D1G2BR4BU6
"
32090 N$(8)="BR5R2F1D1G1L2G1D1F1
R2E1U1H1L2H1U1E1BR3"
32100 N$(9)="BR4BD1E1R2F1D4G1L2H
1BU4D1F1R3BU3"
32110 N$(0)="BR5R2F1D4G1L2H1U4E1
BR3"
32120 RETURN
32130 '
      SUBROUTINE TO DRAW
32140 ' GRAPHIC NUMBERS
32150 COLOR2:DRAW"S8BM100,100;XK
B$;":IF SC <10THENSS$="000"+RIGH
T$(STR$(SC),1):GOTO32190
32160 IF SC <100THENSS$="00"+RIG
HT$(STR$(SC),2):GOTO32190
32170 IF SC <1000THENSS$="0"+RIG
HT$(STR$(SC),3):GOTO32190
32180 SS$=RIGHT$(STR$(SC),4)
32190 B1=VAL(LEFT$(SS$,1))
32200 B2=VAL(MID$(SS$,2,1))
32210 B3=VAL(MID$(SS$,3,1))
32220 B4=VAL(RIGHT$(SS$,1))
32230 KB$=N$(B1)+N$(B2)+N$(B3)+N
$(B4)
32240 COLOR3:DRAW"S8BM100,100;XK
B$;"
32250 RETURN
```

**** BOXING ****

COLOUR COMPUTER

```
10 ' WORLD CHAMPIONSHIP BOXING

     (C) 7/6/83    C. BARTLETT
                   --------

20 CS=0:PT=0:A$="V31L255T25501N1
N2N4N6N4N2N1":B$="V31L255T25502N
1N2N4N6N4N2N1"
30 CLS(0):X=1034:Y=1036
40 POKEX,192:POKEX+1,195:POKEX+2
,226:POKEX+32,192:POKEX+33,202:P
OKEX+34,192:POKEX+64,197:POKEX+6
5,207:POKEX+66,242:POKEX+96,196:
POKEX+97,206:POKEX+98,192:POKEX+
128,192:POKEX+129,203:POKEX+130,
226
50 GOSUB300
60 IFPEEK(341)=247THENPOKEX+128,
128:POKEX+129,128:POKEX+130,128:
X=X-32:IFX<1024THENX=1034
70 IFPEEK(342)=247THENPOKEX+1,12
8:POKEX+2,128:POKEX+3,128:X=X+32
:IFX>1376THENX=1354
80 IFPEEK(344)=247THENPOKEX+130,
195:POKEX+131,226:PLAYA$:POKEX+1
31,128:GOSUB270
90 R=RND(5):IFR<>2THEN40
100 GOSUB110:GOTO130
110 POKEY,128:POKEY+1,161:POKEY+
2,195:POKEY+35,128:POKEY+32,128:
POKEY+33,128:POKEY+34,197:POKEY+
64,128:POKEY+65,177:POKEY+66,207
:POKEY+67,202:POKEY+96,128:POKEY
+97,128:POKEY+98,205:POKEY+99,20
0:POKEY+128,128:POKEY+129,161:PO
KEY+130,199
120 POKEY+131,128:RETURN
130 IF Y-66=X THENGOSUB190:CS=CS
+10:GOSUB200
140 IF Y-34=X THENGOSUB190:CS=CS
+1
150 IF Y-98=X THENGOSUB190:CS=CS
+1
160 IF Y<X+66  THEN POKEY,128:PO
KEY+1,128:POKEY+2,128:Y=Y+32:IFY
>1356 THEN Y=1356:GOSUB110:GOTO4
0ELSEGOSUB110:GOTO40
170 IF Y>X+66 THEN POKEY+128,128
:POKEY+129,128:POKEY+130,128:Y=Y
-32:IF Y<1036 THEN Y=1036:GOSUB1
10:GOTO40ELSEGOSUB110:GOTO40
180 GOTO40
190 POKEY,161:POKEY+1,195:PLAYB$
:FORT=1TO500:NEXT:POKEY,128:RETU
RN
200 POKEX,195:POKEX+1,195:POKEX+
2,195:POKEX+3,226
210 POKEX+32,202:POKEX+33,192:PO
KEX+34,192:POKEX+39,225
220 POKEX+62,206:POKEX+63,203:PO
KEX+64,207:POKEX+65,207:POKEX+66
,207:POKEX+67,239:POKEX+68,236:P
OKEX+69,204:POKEX+70,204:POKEX+7
1,236
230 POKEX+94,204:POKEX+95,200:PO
KEX+96,206:POKEX+97,204:POKEX+98
,204:POKEX+99,236:POKEX+100,236:
POKEX+101,204:POKEX+102,204:POKE
X+103,237
240 POKEX+128,203:POKEX+129,195:
POKEX+130,195:POKEX+131,227
250 PRINT@20,"coco";CHR$(128);"g
otcha";
260 FORT=1TO2000:NEXT:CLS(0):GOT
O30
```

```
270 IF X+34=Y THENPT=PT+1:PRINT@
0,"body";CHR$(128);"blow";
280 IF X+66=Y THENPT=PT+10:PRINT
@0,"head";CHR$(128);"blow";
290 IF X+98=Y THENPT=PT+1:PRINT@
0,"body";CHR$(128);"blow";
300 PRINT@480,"your";CHR$(128);"
score";PT;:PRINT@494,"cocos";CHR
$(128);"score";CS;
310 RETURN
```

**** OTHELLO ****

HITACHI PEACH

```
10 COLOR 7 : WIDTH80 : SCREEN 1
20 '------- Initialize ---------
30 DIM B$(8,8)
40 B$(4,5)="X" : B$(5,4)="X"
50 B$(4,4)="O" : B$(5,5)="O"
60 DATA 0,1, 1,1, 1,0, 1,-1, 0,-1, -1,-1
, -1,0, -1,1
70 FOR I=1 TO 8
80 READ X(I),Y(I)
90 NEXT I
100 X=2 : O=2
110 '------- Set up Board -------
120 LINE(114,2)-(510,134),PSET,B
130 FOR I=4 TO 132 STEP 16
140 LINE(120,I)-(504,I),PSET
150 NEXT I
160 FOR I=120 TO 504 STEP 48
170 LINE(I,4)-(I,132),PSET
180 NEXT I
190 PAINT(116,4),7
200 '--------------------------
210 FOR I=1 TO 8
220 LOCATE I*6+11,18 :PRINT USING "#";I;
230 LOCATE 11,I*2-1  :PRINT USING "#";9-
I
240 NEXT I
250 '------ PLACE PIECES ---------
260 FOR J=4 TO 5
270 FOR I=4 TO 5
280 LOCATE I*6+12,16-(J*2-1)
290 PRINT B$(I,J);
300 NEXT I,J
310 LOCATE 70,3 : PRINT " X      O";
320 LINE(540,20)-(640,50),PSET,7,B
330 LINE(540,33)-(640,33),PSET
340 LINE(590,20)-(590,50),PSET
350 '------- COMMAND ------------
360 LOCATE 70,5 :PRINT USING "##";X; :LO
CATE 76,5:PRINT USING "##";O;
370 LOCATE 75,20 : PRINT USING "##";GAIN
380 Y$="X" : T$="O" :  GOSUB 440
390 LOCATE 70,5 :PRINT USING "##";X; :LO
CATE 76,5:PRINT USING "##";O;
400 LOCATE 3,20 : PRINT USING "##";GAIN
410 Y$="O" : T$="X" :  GOSUB 450
420 GOTO 360
430 '------------------------
440 LOCATE 4,22 : PRINT CHR$(26);"!
8
5
; : GOTO 460
450 LOCATE 1,22 : PRINT CHR$(26); : LOCA
TE 76,22 : PRINT "!
8
5
;
460 IN$=""
470 BUFF$=INKEY$ : IF BUFF$<>"" THEN 470
480 BIT$ =INKEY$ : IF BIT$ ="" THEN 480
490 IN$=IN$+BIT$
500 IF LEN(IN$)<2 THEN 480
510 X1=VAL(LEFT$(IN$,1))
520 Y1=VAL(RIGHT$(IN$,1))
530 IF X1<1 OR X1>8 OR Y1<1 OR Y1>8 THEN
 650
540 IF B$(X1,Y1)<>"" THEN 650
550 '-----------------------------------
-
560 PIVOTS=0
570 FOR I=1 TO 8
580 IF X1+X(I)>8 OR X1+X(I)<1 OR Y1+Y(I)
>8 OR Y1+Y(I)<1 THEN 600
590 IF B$(X1+X(I),Y1+Y(I))=T$ THEN K=2 :
 GOSUB 670
600 NEXT I
610 IF PIVOTS=0 THEN 650
620 GOSUB 740
630 RETURN
640 '---------ERROR PROCESSING----------
-----
650 BEEP : IF Y$="X" THEN 440 ELSE 450
660 '------ CHECK FOR PIVOT ------------
---------
670 A=X1+K*X(I) : B=Y1+K*Y(I)
680 IF A<1 OR A>8 OR B<1 OR B>8 THEN RET
URN
690 IF B$(A,B)=T$ THEN K=K+1 : GOTO 670
700 IF B$(A,B)="" THEN RETURN
710 PIVOTS = PIVOTS +1  : PX(PIVOTS)=X(I
) : PY(PIVOTS)=Y(I): PG(PIVOTS)=K-1
720 RETURN
730 '------- PRINT OUT ----------------
740 B$(X1,Y1)=Y$
750 LOCATE X1*6+12, 16-(Y1*2-1)
760 PRINT B$(X1,Y1);
770 IF Y$="X" THEN X=X+1 ELSE O=O+1
780 '---------------------------------
790 GAIN=0
800 FOR I=1 TO PIVOTS
810 FOR J=1 TO PG(I)
820 A=X1+J*PX(I) : B=Y1+J*PY(I)
830 B$(A,B)=Y$
840 LOCATE A*6+12,16-(B*2-1)
850 PRINT B$(A,B);
860 NEXT J
870 GAIN=GAIN+PG(I)
880 NEXT I
890 IF Y$="X" THEN X=X+GAIN : O=O-GAIN E
LSE O=O+GAIN : X=X-GAIN
900 RETURN
```

**** BLOCK GAME ****

HITACHI PEACH

```
10 SCREEN 0 :WIDTH40
20 X1=200 : X2=440 : Y1=100 : Y2=100
30 DIR1=1 : DIR2=1
40 LINE(0,0)-(640,200),PSET,7,B
50 FOR I=1 TO 400 : NEXT I
60 IN$=INKEY$
70 IF IN$<>"" THEN GOSUB 260
80 GOSUB 110
90 GOTO 60
100 REM ------- Move Lines -----------
110 ON DIR1 GOTO 120,130,140,150
120  Y1=Y1-2 : GOTO 160
130  X1=X1+8 : GOTO 160
140 Y1=Y1+2 : GOTO 160
150 X1=X1-8
160 IF POINT(X1,Y1) THEN R=R+1:LOCATE 15
,10:PRINT "Right Wins"; :GOTO 320
170 PSET(X1,Y1,7)
180 ON DIR2 GOTO 190,200,210,220
190 Y2=Y2-2 : GOTO 230
200 X2=X2+8 : GOTO 230
210 Y2=Y2+2 : GOTO 230
220 X2=X2-8
230 IF POINT(X2,Y2) THEN L=L+1:LOCATE 16
,10:PRINT "Left Wins"; :GOTO 320
240 PSET(X2,Y2,7)
```

```
250 RETURN
260 REM ----- Change Direction ------
270 IF ASC(IN$)=28 THEN DIR2=2 ELSE IF A
SC(IN$)=29 THEN DIR2=4 ELSE IF ASC(IN$)=
30 THEN DIR2=1 ELSE IF ASC(IN$)=31 THEN
DIR2=3
280 IF IN$="A" THEN DIR1=4 ELSE IF IN$="
S" THEN DIR1=2 ELSE IF IN$="Z" THEN DIR1
=3 ELSE IF IN$="W" THEN DIR1=1
290 IF IN$=" " THEN X1=INT(RND*640):Y1=I
NT(RND*200) ELSE IF IN$="?" THEN X2=INT(
RND*640): Y2=INT(RND*200)
300 RETURN
310 REM ------ Start New Game Here ----
320 LOCATE 16,12:PRINT USING "##    ##";
L;R;
330 FOR I=1 TO 800 : NEXT I
340 IF INKEY$<>"" THEN 340
350 GOTO 10
```

```
                 **** L2/16K    HAMBURGER ****

                      TRS-8Ø/SYSTEM-8Ø


1Ø REM                ****************
                      **  HAMBURGER **
                      **      BY    **
                      ** BILL TANTI **
                      ****************

2Ø REM         ***********************************
               ** 49 RONALD ST. COBURG MELB 3Ø58 **
               ***********************************

3Ø REM         ***       (C) COPYRIGHT  1983      ***

4Ø CLEAR 1ØØØ:DEFINTA-Z:DIMCH(2Ø):LEVEL=1:DIMHA$(5ØØ):FAST=1:SLO
W=2:DIMOS(14Ø):DIMSO(14Ø):DIMN$(2Ø):DIMNN$(2Ø):DIMAA$(2Ø)
5Ø A=1:S=1:N$(1)="BILL TANTI":SO(1)=1654Ø:'

6Ø GB=GB+1:IF GB=1 THEN GOSUB 94Ø
7Ø GOTO 114Ø
8Ø REM ***               MAIN LOOP                  ***

9Ø GOSUB15Ø5:MAN$=UN$:NU=Ø:DE=DE+1:IF DE>SL THEN DE=Ø:GOSUB 88Ø

1ØØ IF RO=1 THEN PRINT@RP,"      ";:RO=Ø
11Ø IF K=64 THEN PO=PO+4:PRINT@PO-4,"  ";
12Ø IF K=32 THEN PO=PO-4:PRINT@PO+4,"  ";
13Ø BP=PEEK(1536Ø+64+PO):MP=PEEK(1536Ø+PO):SP=PEEK(1536Ø+PO-1)
14Ø PRINT@PO,MAN$;
15Ø IF SC=>1ØØØØ THEN RT=RT+1:IF RT=1 THEN NH=NH-1:GOSUB 44Ø
16Ø IF MP=19Ø THEN 35Ø              '** BLOW UP
17Ø IF BP=32 THEN 35Ø    ELSE IF BP<>131 THEN IF BP<>19Ø THEN GOS
UB 64Ø    '* MAN,HAM FALL
18Ø IF SP=169 OR SP=17Ø THEN IF K=16 OR K=8 THEN MAN$=OK$:GOTO 2
1Ø
19Ø GOTO 9Ø
2ØØ '***                    UP LADDERS                ***

21Ø GOSUB15Ø5:IF SC=>1ØØØØ THEN MAN$=UN$ ELSE MAN$=OK$
22Ø DE=DE+1:IF DE>SL THEN DE=Ø:GOSUB 88Ø
23Ø IF SC=>1ØØØØ THEN IF PO=S1 OR PO=S2 THEN PO=PO+64:GOTO 39Ø
24Ø TP=PEEK(1536Ø+PO-65):IF TP=17Ø OR TP=169 THEN 25Ø   ELSE IF
K=8 THEN 21Ø
25Ø IF K=16 THEN IF PEEK(1536Ø+PO+64)<>131 THEN PO=PO+64:PRINT@P
O-64,"  ";
26Ø IF K=8 THEN PO=PO-64:PRINT@PO+64,"  ";
27Ø IF K=32 THEN PO=PO-4:PRINT@PO+4,"  ";:GOTO 32Ø
28Ø IF K=64 THEN PO=PO+4:PRINT@PO-4,"  ";:GOTO 32Ø
29Ø IF SC=>1ØØØØ THEN IF PO=S1 OR PO=S2 THEN PO=PO+64:GOTO 39Ø
3ØØ PRINT@PO,MAN$;:IF SC=>1ØØØØ THEN RT=RT+1:IF RT=1 THEN NH=NH-
1:GOSUB 44Ø
31Ø GOTO 21Ø
32Ø IF PO=S1 OR PO=S2 THEN 35Ø   ELSE MP=PEEK(1536Ø+PO):IF PEEK(
1536Ø+PO+64)=32 THEN MAN$=UN$:GOTO 35Ø
33Ø MAN$=UN$:PRINT@PO,MAN$;:GOTO 9Ø
34Ø '***                    BLOW UP MAN               ***

35Ø SP=PEEK(1536Ø+PO-1):IF SP=169 OR SP=17Ø THEN 21Ø
36Ø PO=PO+64:PRINT@PO-64,"  ";
37Ø IF PEEK(1536Ø+PO)<>32 THEN 39Ø
38Ø CO=CO+4:PRINT@PO,MAN$;:FOR LOOP=1 TO 2Ø-CO:NEXT LOOP:GOTO 36
Ø
39Ø PRINT@PO-64,CHR$(153)+CHR$(166);:CO=Ø
4ØØ FOR LOOP=1 TO 5ØØ:NEXT LOOP
41Ø NH=NH+1:IF NH=3 THEN NH=Ø:NS=SC:SC=Ø:GOTO 42Ø   ELSE GOTO 11
4Ø
42Ø SLOW=2:FAST=1:IF NS=Ø OR A=11 AND NS<SO(1Ø) THEN GOTO 94Ø
ELSE GOSUB 49Ø   :NN$=AA$:GOTO 94Ø
43Ø REM ***                HOW MANY MEN              ***

44Ø IF NH=-1 THEN PRINT@45,MAN$+" "+MAN$+" "+MAN$;
45Ø IF NH=Ø THEN PRINT@45,MAN$+" "+MAN$;
46Ø IF NH=1 THEN PRINT@45,MAN$;
47Ø RETURN
48Ø REM ***                ENTER NAME                ***

49Ø AA$="":TT=64:CLS:PRINT@128+12,"Your score has reached the to
p ten.                                       Please enter your name.
";:FOR PP=385 TO 435 STEP 2:TT=TT+1:PRINT@PP,CHR$(TT);:NEXT PP
5ØØ LL=Ø:AA=65:PRINT@441," END";:PRINT@437,CHR$(93);:PRINT@439,"
SP";:PRINT@729,CHR$(95);
51Ø GOSUB15Ø5:V$=INKEY$
52Ø IF K=64 THEN AA=AA+1:LL=LL+2:ELSEIF K=32 THEN AA=AA-1:LL=LL-
2
53Ø IF AA=64 THEN AA=93 ELSE IF AA=94 THEN AA=65
```

```
540 IF LL=-2 THEN LL=56:PRINT@449," "; ELSE IF LL=58 THEN LL=Ø:
PRINT@5Ø5," ";:
55Ø PRINT@LL+447,STRING$(2,32)+CHR$(131)+STRING$(2,32);
56Ø IF V$=" " THEN IF LEN(AA$)>12 THEN IF AA=93 OR AA=91 THEN 59
Ø   ELSE GOTO 51Ø
57Ø IF V$=" " THEN IF AA=91 OR AA=92 OR AA=93 THEN 59Ø   ELSE AA
$=AA$+CHR$(AA):PRINT@729,AA$+CHR$(95);
58Ø GOTO 51Ø
59Ø IF LEN(AA$)=Ø THEN 51Ø   ELSE IF AA=91 THEN AA$=LEFT$(AA$,LE
N(AA$)-1):PRINT@729,AA$+CHR$(95)+" ";
6ØØ IF AA=92 THEN AA$=AA$+CHR$(32):PRINT@729,AA$+CHR$(95);
61Ø IF AA=93 THEN RETURN
62Ø GOTO 51Ø
63Ø REM ***                HAMBURGER FALL           ***

64Ø PW=PO
65Ø BS=BS+1:SP=PEEK(1536Ø+PW+64-BS):IF SP=32 OR SP=149 THEN BS=B
S-1:HP=PW+64-BS ELSE GOTO 65Ø
66Ø IF OP=Ø THEN HA=PEEK(1536Ø+HP)
67Ø HP=HP+64:PRINT@PO,OK$;:GOSUB 88Ø
68Ø SEC=SEC+1:IF SEC=1 THEN PRINT@HP,HA$(HA);:PRINT@HP-64,BLANK$
;:GOTO 67Ø
69Ø DP=PEEK(1536Ø+HP):MP=PEEK(1536Ø+HP):IF MP=131 THEN PRINT@HP,
HA$(HA);:PRINT@HP-64,STRING$(8,32);:BS=-1:SEC=Ø:OP=Ø:RETURN
7ØØ IF HP>768 THEN EP=PEEK(1536Ø+HP):IF EP=32 THEN PRINT@HP,HA$(
HA);:PRINT@HP-64,STRING$(8,32);:GOTO 67Ø   ELSE BS=-1:SEC=Ø:OP=Ø
:ON LEVEL GOSUB 74Ø   ,78Ø   ,82Ø   :RETURN
71Ø IF MP=32 THEN PRINT@HP,HA$(HA);:IF HP>7Ø4+64 THEN PRINT@HP-6
4,STRING$(8,32);:GOTO 67Ø   ELSE GOTO 67Ø
72Ø IF MP<>131 OR MP<>32 OR MP<>176 THEN BS=-1:PW=HP-64+7:OP=HA:
HA=PEEK(1536Ø+HP):PRINT@HP,HA$(OP);:PRINT@HP-64,STRING$(8,32);:G
OTO 65Ø
73Ø '***                 WHEN CUPS FULL           ***

74Ø FOR LOOP=1 TO 3
75Ø     IF PEEK(1536Ø+CH(LO))<>32 THEN NU=NU+1
76Ø NEXT LOOP
77Ø IF NU=3 THEN LE=2:SC=SC+1ØØØ:GOSUB 86Ø   :GOTO 114Ø   ELSE NU
=Ø:RETURN
78Ø FOR LOOP=4 TO 6
79Ø     IF PEEK(1536Ø+CH(LO))<>32 THEN NU=NU+1
8ØØ NEXT LOOP
81Ø IF NU=3 THEN LE=3:SC=SC+2ØØØ:GOSUB 86Ø   :GOTO 114Ø   ELSE NU
=Ø:RETURN
82Ø FOR LOOP=7 TO 9
83Ø     IF PEEK(1536Ø+CH(LO))<>32 THEN NU=NU+1
84Ø NEXT LOOP
85Ø IF NU=3 THEN LE=1:SC=SC+4ØØØ:GOSUB 86Ø   :GOTO 114Ø   ELSE NU
=Ø:RETURN
86Ø SLOW=SLOW-1:IF SL<=-1 THEN FAST=FAST+1:IF FA>=1 THEN FA=1:RE
TURN
87Ø '***                 MOVE SAUSAGE             ***

88Ø FOR LO=1 TO FA:PRINT@S1," ";:PRINT@S2," ";:READ S1,S2:IF S
1=1 THEN RESTORE:GOTO 88Ø
89Ø IF PEEK(1536Ø+S1)=143 THEN WS=1:GOTO 91Ø   ELSE IF PEEK(1536
Ø+S2)=143 THEN WS=2:GOTO 91Ø
9ØØ PRINT@S1,SA$;:PRINT@S2,SA$;:NEXT LO:RETURN
91Ø RN=RND(9):RN=RN*1ØØ:RO=1:SC=SC+RN:PRINT@17,SC;:IF WS=1 THENI
F PEEK(1536Ø+S1-13Ø)<>32 THEN PRINT@S1-129,RN;:RP=S1-129:RETURN
ELSE PRINT@S1-13Ø,RN;:RP=S1-13Ø:RETURN
92Ø IF WS=2 THEN IF PEEK(1536Ø+S2-13Ø)<>32 THEN PRINT@S2-129,RN;
:RP=S2-129:RETURN ELSE PRINT@S2-13Ø,RN;:RP=S2-13Ø:RETURN
93Ø '***                 HIGH SCORE               ***

94Ø IF LEN(NN$)>12 THEN NN$=LEFT$(NN$,12)
95Ø IF NS=Ø THEN A=Ø:GOSUB 1Ø3Ø  :GOTO 1Ø9Ø
96Ø FOR I=1 TO 1Ø:OS(I)=SO(I):NA$(I)=N$(I):NEXT I
97Ø S=S+1:IF S=11 THEN S=1Ø
98Ø FOR T=1 TO S:IF NS>SO(T) THEN P=T:GOTO 1ØØØ  ELSE NEXT T
99Ø A=Ø:GOSUB 1Ø3Ø  :GOTO 1Ø9Ø
1ØØØ FOR U=1 TO 1Ø:IF P=U THEN SO(U)=NS:N$(U)=NN$:GOTO 1Ø1Ø  ELS
E NEXT U
1Ø1Ø FOR E=U+1 TO 1Ø:SO(E)=OS(E-1):N$(E)=NA$(E-1):NEXT E
1Ø2Ø A=Ø:GOSUB 1Ø3Ø  :GOTO 1Ø9Ø
1Ø3Ø CLS:GOSUB 1Ø4Ø  :PRINT@1Ø8," Hamburger ";:PRINT@172,"High s
cores";:FOR W=246 TO 886 STEP 64:A=A+1:IF SO(A)=Ø OR A=11 THEN R
ETURN ELSE PRINT@W-18,A;:PRINT@W-14,N$(A)+" "+STRING$(14-LEN(N$(
A)),95);:PRINT@W,USING"###,###";SO(A);:NEXT W:RETURN
1Ø4Ø PRINT@66,"  It is your  job to make as many
 hamburgers  as  you  can. A  bonus
 is   awarded   every   time  three
 hamburgers  are  made. The  bigger
 the  hamburgers   the  bigger  the
 bonus.";
1Ø5Ø PRINT@45Ø,"  You make a hamburger by running
 over either a bun,a piece of salad
 an  egg  or a  piece of meat. What
 ever you run over will drop.";
1Ø6Ø PRINT@7Ø6,"  There are two sausages  running
 after  you  and if they catch you,
 you die. A shield will protect you
 in the lifts & when buns etc fall.";
1Ø7Ø PRINT@Ø,STRING$(62,14Ø);:PRINT@96Ø,STRING$(63,14Ø);:PRINT@Ø
,CHR$(188);:PRINT@96Ø,CHR$(143);:PRINT@35,CHR$(172);:PRINT@62,CH
R$(188);:PRINT@995,CHR$(142);:FOR Q=64 TO 896 STEP 64:PRINT@Q,CH
R$(191);:PRINT@Q+35,CHR$(17Ø);:PRINT@Q+62,CHR$(191);:NEXT Q
1Ø8Ø POKE 16382,143:RETURN
1Ø9Ø PRINT@934,"PRESS <CLEAR> TO START";
11ØØ SX=SX+1:IF SX<3 THEN PRINT@94Ø,"       "; ELSE IF SX>3 THEN
 PRINT@94Ø,"<CLEAR>";
111Ø IF PEEK(14444)=2 THEN LEVEL=1:GOTO 114Ø  ELSE IF SX>12 THEN
 SX=Ø
112Ø GOTO 11ØØ
113Ø REM ***              SET VARIABLES            ***

114Ø BS=-1:PO=816:HATBUN$(142)=CHR$(142)+STRING$(6,143)+CHR$(141
):HAEGG$(14Ø)=CHR$(14Ø)+STRING$(4,143)+STRING$(3,14Ø):HAMEAT$(14
3)=STRING$(8,143):HASALAD$(134)=CHR$(134)+CHR$(139)+STRING$(2,14
2)+CHR$(134)+CHR$(141)+CHR$(139)
```

```
1150 CH(9)=744:SA$=CHR$(190)+CHR$(180):OK$=STRING$(2,191):UNSAFE
$=CHR$(158)+CHR$(173)
1160 CH(1)=847:CH(2)=859:CH(3)=872:CH(4)=783:CH(5)=795:CH(6)=808
:CH(7)=719:CH(8)=731:HABBUN$(139)=CHR$(139)+STRING$(6,143)+CHR$(
135):LADER$=CHR$(170)+STRING$(2,32)+CHR$(149):MAN$=CHR$(158)+CHR
$(173):BLANK$=STRING$(8,131):L$=CHR$(169)+STRING$(2,32)+CHR$(150
)
1170 '***
                SET THE SCREENS          ***

1180 CLS
1190 FOR LOOP=67 TO 899 STEP 64
1200     PRINT@LOOP,L$;:PRINT@LOOP+44,L$;:PRINT@LOOP+52,L$;
1210 NEXT LOOP
1220 PRINT@964,STRING$(2,131);:PRINT@1008,STRING$(2,131);:PRINT@
1016,STRING$(2,131);
1230 FOR LOOP=83 TO 339 STEP 64
1240     PRINT@LOOP,LADR$;
1250 NEXT LOOP
1260 FOR LOOP=351 TO 607 STEP 64
1270     PRINT@LOOP,LA$;
1280 NEXT LOOP
1290 FOR LOOP=136 TO 712 STEP 128
1300     PRINT@LOOP,STRING$(2,131);:PRINT@LOOP+36,STRING$(2,131);
1310 NEXT LOOP
1320 FOR LOOP=160 TO 608 STEP 128
1330     COUNT=COUNT+1:IF CO<=2 THEN PRINT@LOOP,STRING$(2,131); E
LSE PRINT@LOOP-12,STRING$(2,131);
1340 NEXT LOOP:PRINT@672,STRING$(2,131);:PRINT@660,STRING$(2,131
);
1350 PRINT@969,CHR$(137)+CHR$(144)+"         "+CHR$(152)+CHR$(13
7)+CHR$(144)+"        "+CHR$(152)+CHR$(137)+CHR$(144)+"
"+CHR$(152)+CHR$(129);
1360 PRINT@10,"SCORE";:PRINT@17,SC;:PRINT@27,USING"####,###";SO(
1);:GOSUB 440
1370 ON LEVEL GOSUB 1390  ,1410  ,1430
1380 GOTO 210
1390 PRINT@139,HATBUN$(142);:PRINT@151,HAMTBUN$(142);:PRINT@163,
BLANK$;:PRINT@267,BLANK$;:PRINT@279,BLANK$;:PRINT@291,HATBUN$(14
2);:PRINT@395,HAMEAT$(143);:PRINT@407,HAMEAT$(143);:PRINT@419,HA
MEAT$(143);:PRINT@523,HABBUN$(139);:PRINT@535,HABBUN$(139);
1400 PRINT@148,"  ";:PRINT@276,"  ";:PRINT@160,STRING$(2,131);:P
RINT@288,STRING$(2,131);:PRINT@547,HABBUN$(139);:PRINT@651,BLANK
$;:PRINT@663,BLANK$;:PRINT@675,BLANK$;:RETURN
1410 PRINT@139,HATBUN$(142);:PRINT@151,BLANK$;:PRINT@163,HATBUN$
(142);:PRINT@267,BLANK$;:PRINT@279,HATBUN$(142);:PRINT@291,HAEGG
$(140);:PRINT@395,HAEGG$(140);:PRINT@407,HASALAD$(134);:PRINT@41
9,HAMEAT$(143);:PRINT@523,HAMEAT$(143);:PRINT@535,HAMEAT$(143);
1420 PRINT@148,"  ";:PRINT@276,"  ";:PRINT@160,STRING$(2,131);:P
RINT@288,STRING$(2,131);:PRINT@547,HAMEAT$(139);:PRINT@651,HABBU
N$(139);:PRINT@663,HABBUN$(139);:PRINT@675,BLANK$;:RETURN
1430 PRINT@139,HATBUN$(142);:PRINT@151,HATBUN$(142);:PRINT@163,H
ATBUN$(142);:PRINT@267,HASALAD$(134);:PRINT@279,HAEGG$(140);:PRI
NT@291,HASALAD$(134);:PRINT@395,HAMEAT$(143);:PRINT@407,HAMEAT$(
143);:PRINT@419,HAMEAT$(143);:PRINT@523,HAEGG$(140);
1440 PRINT@148,"  ";:PRINT@276,"  ";:PRINT@160,STRING$(2,131);:P
RINT@288,STRING$(2,131);:PRINT@535,HASALAD$(134);:PRINT@547,HAEG
G$(140);:PRINT@651,HABBUN$(139);:PRINT@663,HABBUN$(139);:PRINT@6
75,HABBUN$(139);:RETURN
1450 DATA 632,580,568,516,504,452,440,388,376,324,312,260,248,19
6,244,132,240,68,236,72,232,76,228,80,224,84,220,88,216,92,212,9
6,276,100,340,104,344,108,348,112,352,176,356,240,360,236,364,23
2,368,228,432,224,496,220,492
1460 DATA 216,488,212,484,276,480,340,416,344,352,348,348,352,34
4,416,340,480,336,544,332,608,328,612,324,616,388,620,452,624,51
6,560,580,496,584,492,588,488,592,484,596,480,600,544,604,608,60
8,604,612,600,616,596,620,592,624,588,628,584,1,1
1470 REM  FOR MACHINES WITHOUT LEFT AND RIGHT ARROW KEYS
          CHANGE THE FOLLOWING LINES.
1480 REM  IN LINES 110,280,520 CHANGE THE VALUE OF 32 TO 4
1490 REM  IN LINES 120,270,520 CHANGE THE VALUE OF 64 TO 32
1500 REM  THEN USE ":" KEY FOR LEFT AND THE "-" KEY FOR RIGHT
1505 K=PEEK(14444)
1510 IFK=0THENK=8*(255-INP(0))
1520 RETURN
```

**** L2/16K   SYSTEM TAPE MAKER ****

TRS-80/SYSTEM-80

```
00010 ;SYSTEM TAPE PROGRAM   -  V1.0
00020 ; BY MICHAEL BYRNE - NOV '81
00030
00040 ; - MAKES USE OF THE L3 SAVE COMMAND
00050 ; THE FORMAT OF THE COMMAND IS :-
00060 ;       SAVE <NAME> <ADDR1> <ADDR2> <ADDR3>
00070
00080 ; WHERE <NAME> IS UP TO 6 CHARS
00090 ;       <ADDR1> IS THE ADDRESS TO START SAVING FROM
00100 ;       <ADDR2> IS THE ADDRESS TO SAVE TO
00110 ;       <ADDR3> IS THE ENTRY POINT
00120 ;ADDR3 MAY BE OMITTED IN WHICH CASE ADDR1 IS USED
00130         ORG     07EB0H
00140
00150 BASIC   EQU     06CCH    ;EP INTO BASIC
00160 OUTCH   EQU     0264H    ;WRITES A BYTE TO TAPE FROM A
00170 SNERR   EQU     1997H    ;PRINTS ?SN ERROR
00180 UEERR   EQU     2003H    ;PRINTS ?UE ERROR
00190
00200
00210
00220 SETUP   LD      HL,SAVE ;ADDR OF INTERCEPT
00230         LD      A,0C3H  ;JP OPCODE
00240         LD      (41A0H),A;
00250         LD      (41A1H),HL
00260         JP      BASIC   ;RETURN TO BASIC
00270
```

```
00280 ; THE MAIN ROUTINE STARTS HERE
00290 ; ON ENTRY HL IS POINTING TO THE FIRST NON-BLANK
00300 ; CHAR FOLLOWING THE WORD "SAVE".  THIS SHOULD
00310 ; BE THE FIRST CHAR OF THE NAME.
00320
00330 SAVE    CALL    GETNAM  ;
00340         CALL    SCAN    ;
00350         CALL    GETADR  ;GET START ADDR
00360         LD      IX,START;
00370         LD      (IX+00),E;STORE START ADDR
00380         LD      (IX+01),D
00390         LD      A,(HL)  ; GET NEXT CHAR
00400         CP      00      ;END OF LINE (EOLN)?
00410         JP      Z,SNERR ;IF SO ITS AN ERROR
00420         CALL    SCAN
00430         CALL    GETADR  ;GET END ADDR
00440         LD      IX,EOB  ;
00450         LD      (IX+00),E       ;AND STORE IT
00460         LD      (IX+01),D
00470         LD      A,(HL)  ;GET NEXT CHAR
00480         CP      00      ;EOLN?
00490         JR      NZ,LASTAD       ;
00500         PUSH    HL      ;
00510         LD      HL,(START)      ;ELSE GET START ADDR
00520         LD      (EP),HL ;
00530         JP      CHKADR  ;CHECK START > END
00540 LASTAD  CALL    SCAN
00550         CALL    GETADR
00560         PUSH    HL      ;
00570         LD      (EP),DE ;
00580 CHKADR  XOR     A       ;ZERO A
00590         LD      HL,(EOB)        ;
00600         LD      BC,(START)
00610         SBC     HL,BC   ;HL=HL-BC
00620         JP      Z,UEERR ;ERROR IF START=END
00630         JP      C,UEERR ;       OR START<END
00640         LD      DE,0001 ;
00650         ADD     HL,DE   ;ADD 1 TO GET BLOCK LENGTH
00660         LD      (LEN),HL
00670
00680 ;WE NOW HAVE THE NAME, START, LENGTH AND ENTRY PT
00690 ;OF THE BLOCK SO START MAKING THE TAPE
00700
00710         CALL    0284H   ;WRITE LEADER
00720         LD      A,55H   ;SYSTEM TAPE INDIC
00730         CALL    OUTCH   ;WRITE IT
00740         LD      A,'*'   ;
00750         LD      (3C3EH),A
00760         LD      (3C3FH),A       ;ASTERISKS TO BLINK
00770         LD      B,06    ;NAME LENGTH
00780         LD      HL,NAME ;
00790 NLOOP   LD      A,(HL)
00800         INC     HL      ;
00810         CALL    OUTCH
00820         DJNZ    NLOOP
```

```
00830         LD      HL,(START)
00840         LD      DE,(LEN)
00850 BLOCK   LD      A,D     ;
00860         AND     A       ;
00870         LD      B,00    ;
00880         JR      NZ,MORE
00890         LD      B,E
00900         OR      B
00910         JR      Z,FINISH
00920 MORE    LD      A,3CH   ;
00930         CALL    OUTCH
00940         LD      A,B
00950         CALL    OUTCH
00960         LD      A,L
00970         CALL    OUTCH
00980         ADD     A,H
00990         LD      C,A
01000         LD      A,H
01010         CALL    OUTCH
01020 BYTE    LD      A,(HL)
01030         CALL    OUTCH
01040         ADD     A,C
01050         LD      C,A
01060         INC     HL
01070         DEC     DE
01080         DJNZ    BYTE
01090         LD      A,C
01100         CALL    OUTCH
01110         CALL    022CH
01120         JR      BLOCK
01130 FINISH  LD      A,78H
01140         CALL    OUTCH
01150         LD      HL,(EP)
01160         LD      A,L
01170         CALL    OUTCH
01180         LD      A,H
01190         CALL    OUTCH
01200         CALL    01F8H
01210         POP     HL
01220         RET
01230
01240 ;ROUTINES USED
01250
01260 ;ON ENTRY TO GETNAM HL PTS TO FIRST CHAR OF NAME
01270
01280 GETNAM  LD      B,06H
01290         LD      IX,NAME ;WHERE TO PUT IT
01300 STORE   LD      A,(HL)  ;
01310         INC     HL      ;NEXT
01320         CP      ' '     ;END OF FIELD
01330         JR      Z,PAD   ;IF SO PAD IT
01340         CP      00      ;EOLN?
01350         JP      Z,SNERR ;IF SO ERROR
01360         LD      (IX+00),A
01370         INC     IX
```

```
01380          DJNZ     STORE
01390          RET
01400 PAD      LD       (IX+00),A         ;STORE A SPACE
01410          INC      IX
01420          DJNZ     PAD
01430          RET               ;END OF GETNAM ROUTINE
01440
01450
01460 ;ROUTINE GETS ADDR AND RETURNS VALUE IN DE PAIIR
01470
01480 GETADR   LD       DE,0000 ;INIT DE
01490 ADRLP    LD       A,(HL)  ;GETCHR
01500          CP       0       ;EOLN?
01510          RET      Z       ;YES, RETURN
01520          CP       ' '     ;END OF FIELD?
01530          RET      Z       ;YES, RETURN
01540          CP       '0'     ;< '0'?
01550          JP       C,SNERR ;IF SO THEN ERROR
01560          CP       'G'     ;> 'F'?
01570          JP       NC,SNERR;IF SO THEN ERROR
01580          SUB      '0'     ;CONVERT
01590          CP       0AH     ;> 9?
01600          JP       C,ADDIT ;NO, SO ADD IT IN
01610          SUB      07      ;YES, SUB 7
01620          CP       0AH     ;< 10?
01630          JP       C,SNERR ;YES, AN ERROR
01640 ADDIT    LD       B,04    ;FOR SHIFTING
01650          LD       C,A     ;SAVE VALUE
01660 SHIFT    SLA      D       ;
01670          SLA      E       ;
01680          JR       C,ADDONE;SHIFT 1'S FROM E TO D
01690          DJNZ     SHIFT   ;
01700          JP       NEXT
01710 ADDONE   LD       A,D     ;
01720          OR       01      ;
01730          LD       D,A     ;
01740          DJNZ     SHIFT   ;
01750 NEXT     LD       A,C     ;RESTORE VALUE
01760          ADD      A,E     ;
01770          LD       E,A     ;
01780          INC      HL
01790          JP       ADRLP
01800
01810 SCAN     LD       A,(HL)  ;
01820          CP       00
01830          RET      Z
01840          CP       21H     ;<SPACE?
01850          RET      NC
01860          INC      HL
01870          JP       SCAN
01880 ;NOW FOR VARIABLES
01890 START    DEFW     0000
01900 EOB      DEFW     0000
01910 EP       DEFW     0000
01920 LEN      DEFW     0000
01930 NAME     DEFS     06
01940
01950          END      SETUP
```

```
SYSTEM TAPE MAKER DUMP
START   END      ENTRY
7EB0    7FF0     7EB0


7EB0:  21 BE 7E 3E C3 32 A0 41 22 A1 41 C3 CC 06 CD 80
7EC0:  7F CD DE 7F CD A1 7F DD 21 E9 7F DD 73 00 DD 72
7ED0:  01 7E FE 00 CA 97 19 CD DE 7F CD A1 7F DD 21 EB
7EE0:  7F DD 73 00 DD 72 01 7E FE 00 20 0A E5 2A E9 7F
7EF0:  22 ED 7F C3 01 7F CD DE 7F CD A1 7F E5 ED 53 ED
7F00:  7F AF 2A EB 7F ED 4B E9 7F ED 42 CA 03 20 DA 03
7F10:  20 11 01 00 19 22 EF 7F CD 84 02 3E 55 CD 64 02
7F20:  3E 2A 32 3E 3C 32 3F 3C 06 06 21 F1 7F 7E 23 CD
7F30:  64 02 10 F9 2A E9 7F ED 5B EF 7F 7A A7 06 00 20
7F40:  04 43 B0 28 26 3E 3C CD 64 02 78 CD 64 02 7D CD
7F50:  64 02 84 4F 7C CD 64 02 7E CD 64 02 81 4F 23 1B
7F60:  10 F6 79 CD 64 02 CD 2C 02 18 D0 3E 78 CD 64 02
7F70:  2A ED 7F 7D CD 64 02 7C CD 64 02 CD F8 01 E1 C9
7F80:  06 06 DD 21 F1 7F 7E 23 FE 20 28 0D FE 00 CA 97
7F90:  19 DD 77 00 DD 23 10 EE C9 DD 77 00 DD 23 10 F9
7FA0:  C9 11 00 00 7E FE 00 C8 FE 20 C8 FE 30 DA 97 19
7FB0:  FE 47 D2 97 19 D6 30 FE 0A DA C3 7F D6 07 FE 0A
7FC0:  DA 97 19 06 04 4F CB 22 CB 23 38 05 10 F8 C3 D7
7FD0:  7F 7A F6 01 57 10 EF 79 83 5F 23 C3 A4 7F 7E FE
7FE0:  00 C8 FE 21 D0 23 C3 DE 7F 00 00 00 00 00 00 00
7FF0:  00
```

**** L2/16K   XUSR SCREEN FILL SUBROUTINE ****

TRS-80/SYSTEM-80

```
1 REM This is a subroutine that allows you to fill the screen
2 REM with any character you wish.
3 REM X=USR(n)     where n is an ASCII character from 0 to 255
4 REM (c) 1983 Carl Cranstone
5 REM Lines 10 to 30 are main program
6 REM To be incorporated into your own programs
10 S$="THIS IS A DUMMY STRI"
20 DATA205,127,10,125,33,0,60,17,1,60,1,255,3,119,237,176,201:V=
PEEK(VARPTR(S$)+2)*256+PEEK(VARPTR(S$)+1):FORG=VTOV+16:READF:POK
EG,F:NEXT
30 C=PEEK(16549)*256+PEEK(16548):IFC<>17129ANDC<>17385THENDEFUSR
0=V:CMD"T"ELSEV1=INT(V/256):V2=INT(V-V1*256):POKE16527,V1:POKE16
526,V2
```

**** L1/4K   GALACTIC BATTLES ****

TRS-80/SYSTEM-80

```
20 IN."LARGE NO.";X:P=1:B$="FAILED"
25 IFX>1000X=X/300:G.25
30 F.I=1TOX:J=R.(100):N.I
40 S=0:K=12:C=1:F=9000:N=0:F.I=1TO6S.3:A(40+I)=R.(4):A(41+I)=R.(
115)+5
60 A(42+I)=R.(38)+2:N.I:F.I=1TO36S.3:A(I)=R.(4)
100 A(I+1)=R.(121)+3:A(I+2)=R.(36)+2:N.I:Q=R.(4):T=40
130 CLS:F.I=1TO6S.3:IFA(40+I)<>QT.150
140 F.J=A(41+I)TO(A(41+I)+5):S.(J,A(42+I)):N.J
150 N.I:F.I=1TO36S.3:IFA(I)<>QT.180
160 X=A(I+1):Y=A(I+2):S.(X+3,Y):S.(X,Y):S.(X,Y+1):S.(X+3,Y+1)
170 S.(X,Y+2):S.(X+3,Y+2):S.(X+1,Y+1):S.(X+2,Y+1)
180 N.I:P.A.897,"2 1";:P.A.961,"3 4  ";
200 IFQ=1S.(8,43)
210 IFQ=2S.(4,43)
220 IFQ=3S.(4,46)
230 IFQ=4S.(8,46)
235 U=R.(117)+5:V=R.(37)+2
240 F.I=(U-6)TO(U+6):F.J=(V-2)TO(V+2):IFP.(I,J)=1T.235
260 N.J:N.I:I=U:J=V:GOS.2002
375 N=R.(4)+4:F.I=1TON:X=R.(127):Y=R.(40):IFP.(X,Y)=0S.(X,Y)
385 N.I
400 P.A.906,"COMP-1  PHASR-2  NAVGT-3  SHLD-4  DOCK-5  CHRG-6";:
IN.N
410 P.A.906:ONNGOS.4000,5000,6000,8000,9000
422 IFN=100V=V+4:G.130
425 IFN<>1T=T-1:F=F+100:GOS.7000
427 IF(T=0)+(K=0)P.A.906,"---MISSION ";B$;"---":END
430 G.400
1000 P.A.906,"--INOPERATIVE--":N=1:F.J=1TO999:N.J:RET.
2002 F.L=(I-4)TO(I+4):S.(L,J):N.L:S.(I+1,J-1):S.(I,J+1):S.(I-1,J
-1)
2010 S.(I+5,J-1):S.(I-5,J-1):U=I:V=J:RET.
2502 F.L=(U-4)TO(U+4):R.(L,V):N.L
2505 R.(U+1,V-1):R.(U,V+1):R.(U-1,V-1):R.(U+5,V-1):R.(U-5,V-1):R
ET.
4000 IFC=0G.1000
4020 P.A.906,"ENERGY=";F;" SHIELDS=";S;" TIME=";T:P.A.950,"(ENTE
R)";
4025 IN.A$:P.A.906,"STARBASE QUADS:";A(41);"&";A(44);"       E.B
.S=";K;
4030 IN.A$:P.A.906:RET.
5000 IFP=0G.1000
5010 P.A.906,"ANGLE (0-^360) ";:IN.E
5020 P.A.906,"POWER (1-^1000)    ";:IN.D:IF(D>F)+(D<1)+(D>1000)T
.1000
5030 F=F-D:X=E:GOS.30360:G=Y:X=E:GOS.30376:B=-Y:X=U:Y=V
5035 IF(U>122)+(U<5)+(V>40)+(V<2)RET.
5037 IFABS(G)<=.04G=0:G.5060
5040 F.I=(X+G*6+.5)TO(X+G*6+D/G+.5)S.(G/ABS(G))
5042 J=Y+(I-X)*B/G/2+B*2:IF(I>122)+(I<5)+(J>40)+(J<2)RET.
5043 GOS.5045:IFN=0RET.
5044 N.I:RET.
5045 IFP.(I,J)=1N=0:G.5500
5047 IFN=3GOS.2002:GOS.2502:RET.
5050 S.(I,J):R.(I,J):RET.
5060 F.J=(Y+B*2)TO(Y+B*(D/2+2))S.B
5062 I=X:IF(I>122)+(I<5)+(J>39)+(J<3)RET.
5065 GOS.5045:IFN=0RET.
5070 N.J:RET.
5500 IF(P.(I-1,J)=0)*(P.(I+1,J)=0)*(P.(I,J-1)=0)*(P.(I,J+1)=0)RE
T.
5510 F.X=1TO36S.3:IFA(X)<>QT.5600
5520 IF(I<A(X+1))+(I>A(X+1)+3)T.5600
5530 IF(J<A(X+2))+(J>A(X+2)+3)T.5600
5540 IFR.(1100)>DRET.
5550 A(X)=0:K=K-1:IFK=0B$="COMPLETED"
5560 F.I=A(X+1)TO(A(X+1)+3):F.J=A(X+2)TO(A(X+2)+2):R.(I,J):N.J:N
.I:RET.
5600 N.X:IF(I<U-5)+(I>U+5)+(J<V-1)+(J>V+1)RET.
5610 F=F-S/2:IF((D>S/2)*(C=0))+(D>S)P=0
5615 IFD>S/1.6C=0
5620 P.A.906,"-SHIP HIT-":F.X=1TO500:N.X:P.A.906:RET.
6000 P.A.906,"QUADRANT";:IN.B:IF(B<>Q)*(F<1000)T.1000
6005 IFB<>QF=F-1000:Q=B:G.130
6010 P.A.906,"GIRO-ANGLE (0-^360) ";:IN.E
6020 P.A.906,"MAGNITUDE   (1-^500)     ";:IN.D:IF(D<1)+(D*10>F)T.
1000
6030 F=F-10*D:I=U:J=V:GOS.2502:GOS.5030:I=U:J=V:GOS.2002:RET.
7000 F.H=1TO36S.3:IFA(H)<>QT.7500
7010 X=A(H+1):Y=A(H+2):D=R.(500)+300:G=-R.(0)*(X-U)/X
7015 B=-R.(0)*(Y-V)/Y:N=-1:GOS.5035
7500 N.H:RET.
8000 P.A.906,"SHIELDS=";S;" NEW RATING";:IN.L:IF(L>F+S)+(L>1000)
T.8000
8020 F=F+S:S=L:F=F-S:RET.
9000 F.L=1TO6S.3:IFA(40+L)<>QT.9020
9010 IF(U>A(41+L)-6)*(U<A(41+L)+6)*(V>A(42+L)-5)*(V<A(42+L)+5)T.
9500
9020 N.L:GOS.1000:RET.
9500 M=Q:F.L=1TO9:IFR.(0)>.8C=1
9510 F=F+1000:GOS.7000:N.L:N=100:T=T-5:P=1:RET.
30360 W=ABS(X)/X:X=X+90:GOS.30376:IF(Z=-1)*(W=1)T.Y=-Y
30365 RET.
30376 Z=ABS(X)/X:X=Z*X
30380 IFX>360T.X=X/360:X=(X-INT(X))*360
30390 IFX>90T.X=X/90:Y=INT(X):X=(X-Y)*90:ONYG.30410,30420,30430
30400 X=X/57.29578:IFABS(X)<2.48616E-4Y=0:RET.
30405 G.30440
30410 X=90-X:G.30400
30420 X=-X:G.30400
30430 X=X-90:G.30400
30440 Y=X-X*X*X/6+X*X*X*X*X/120-X*X*X*X*X*X*X/5040
30450 Y=Y+X*X*X*X*X*X*X*X*X/362880:IFZ=-1T.Y=-Y
30455 RET.
```

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

Next month's issue will contain at least the following programs plus the usual features and articles. An (80) after a program title indicates that the program will be for TRS-80 Model 1/3 or System 80/Video Genie. A (CC) indicates that the program will be for the TRS-80 Colour Computer and (Peach) that the program is for the Hitachi Peach.

** SOURCE Model 3 (80) **

This program is a utility that will allow Edtasm source files to be transferred between tape and disk for Model 3 users of NEWDOS80 V2.0. Tape I/O can be high or low speed, as desired.

** HI RES WRITER 1.1 (CC) **

At last it's been done!!! A relocatable machine language program that lets you write on a high resolution screen with the normal syntax of the PRINT@ statement. That's not all! This program gives you TRUE lower case characters as well. As if that was not enough....this machine language program will append itself to your BASIC program so that when you save it to tape or disk, the next time you load the program, the machine language section will also load into memory along with your program.

** TRACK RACER (Peach) **

You are in control of a racing car, tearing down the track dodging the other cars. Only quick reactions will keep you in this race. This all new program will keep you on the edge of your seat.

** GRAFX L2/32K Disk (80) **

GRAFX is a tool for use in generating bold titles or graphics for incorporation in your programs. Once you have designed your titles or graphics, GRAFX will write a BASIC Program to reproduce them. The BASIC program thus created is a completely stand-alone normal program that, when run, will reproduce your graphics or titles.

** LVAR L2/16K (80) **

This is an extremely useful debugging function for BASIC programs. The program resides at the top of 16K memory (though it can easily be relocated elsewhere). When the program is first run, it initialises the DOS exit NAME to point to this program. After that, when NAME is typed it will give a list of all of the single precision, double precision, integer and string variables used in your program and their values.

APPLICATION FOR PUBLICATION
OF A PROGRAM
IN MICRO-80

Date ....

To **MICRO-80**
SOFTWARE DEPT.,
P.O. BOX 145,
MORPHETT VALE, S.A. 5162.

Please consider the enclosed program for publication in MICRO-80.

Name ....

Address ....

....

.... Postcode ....

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

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

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

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

The changes or improvements that you think may improve it.

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

***** CASSETTE/DISK EDITION INDEX *****

The cassette edition of MICRO-80 contains all the applicable software listed each month, on cassette. For machine language programs copies of both the source and object file are provided. All programs are recorded twice. Level 1 programs can only be loaded into a Level 2 machine if the 'Level 1 in Level 2' program from the MICRO-80 Software Library - Vol.1 is loaded first.

Note: System 80/Video Genie computers have had different tape-counters fitted at different times. The approximate start positions shown are correct for the very early System 80 without the volume control or level meter. They are probably incorrect for later machines. The rates for a cassette subscription are printed on the inside front cover of each issue of the magazine.

The disk edition contains all applicable programs which can be executed from disk. Level 1 disk programs are saved in NEWDOS format. Users require the Level I/CMD utility supplied with NEWDOS+ or NEWDOS 80 version 1.0 to run them.

				APPROX. START POSITION		
SIDE 1	TYPE	I.D.	DISK FILESPEC	CTR-41	CTR-80	SYSTEM 80
HAMBURGER	L2/16K	H	HAMBURG/BAS	18	10	6
"	"	"	"	126	70	47
XUSR UTILITY	L2/4K	X	XUSR/BAS	220	123	92
"	"	"	"	229	128	96
SAVE COMMAND	SYSTEM	SAVE	SAVE/CMD	240	134	104
"	"	"	"	247	138	108
SAVE COMMAND	EDTASM	SAVE	SAVE/EDT	255	142	112
"	"	"	"	295	165	135
GALACTIC BATTLES	L1/4K	-	GALACTIC/LV1	335	187	161
"	"	-	"	384	215	197
SIDE 2						
HI-RES SCREEN SCORE	COCO	S	-	18	10	-
"	"	"	-	29	16	-
BOXING	COCO	B	-	45	25	-
"	"	"	-	58	32	-
OTHELLO	HP	OTHELLO	-	90	50	-
"	"	"	-	126	70	-
BLOCK GAME	HP	BLOCK	-	161	90	-
"	"	"	-	179	100	-

TO:
MICRO-80, P.O. BOX 213, GOODWOOD,
SOUTH AUSTRALIA. 5034.

Please RUSH to me the items shown below:

$ .... enclosed Date ....

.... 12 month subscription to MICRO-80

.... 12 mth. subs. to MICRO-80, plus the cassette edition

.... 12 mth. subs. to MICRO-80, plus the disc edition

.... The latest issue of MICRO-80 (see inside front cover for prices)

The MICRO-80 PRODUCTS listed below:

DESCRIPTION	PRICE
TOTAL ENCLOSED WITH ORDER	

□ Cheque □ Bankcard □ Money Order

Bankcard Account Number

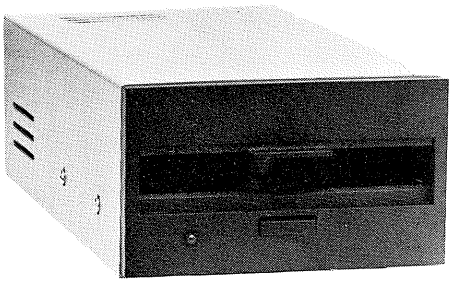
Signature .... Exp. End ....

NAME ....

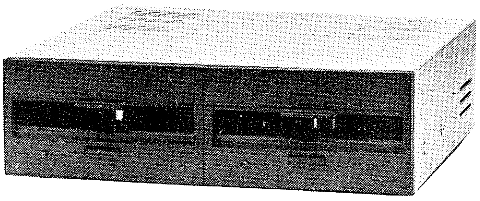
ADDRESS ....

.... Postcode ....

SAVE A PACKET ON MICRO-80's DISK DRIVE PACKAGES FOR TRS-80 MODEL 1 AND SYSTEM 80 MICROCOMPUTERS



SINGLE DRIVE PACKAGE *from* ... $499



DUAL DRIVE PACKAGE *from* ... $874

Bigger volume means lower cost price, which we are passing on to you. Avoid the annoying bundle of cables, wires and separate boxes. MICRO-80 is now offering our well-proven MPI disk drives in attractive, self-contained single or dual-drive cabinets complete with internal power supply. Our drive Ø and dual-drive packages also include the appropriate version of DOSPLUS and dual-drive cable.

The best news of all is the specially reduced package prices ...
SAVE $23 — $107 over our already low prices!

Choose the appropriate system from the table below:

DRIVE TYPE	No. of Tracks	No. of Heads	Capacity	Dosplus Version	Price	* Saving
DRIVE Ø						
1 x MPI B51	40	1	100K	3.4	$499	$137.95
1 x MPI B52	40	2	200K	3.4	$639	$97.95
1 x MPI B92	80	2	400K	3.4	$799	$107.95
DRIVE 1						
1 x MPI B51	40	1	100K	—	$415	$33.00
1 x MPI B52	40	2	200K	—	$525	$23.00
1 x MPI B92	80	2	400K	—	$695	$23.00

*Represents the saving compared with buying all the items included in the package separately

- Drive Ø package includes one bare disk drive, self-contained single-drive cabinet/power supply as illustrated, two drive cable and the version of DOSPLUS indicated.
- Drive 1 package includes one bare disk drive and self-contained single-drive cabinet/power supply as illustrated.

If it's a dual-drive system you need, then take advantage of our dual-drive package and
SAVE a further $40 on the price of two single-drive packages ...

DRIVE TYPE	No. of Tracks	No. of Heads	Capacity	Dosplus Version	Price
2 x MPI B51	40 ea	1 ea	2 x 100K	3.4	$874
2 x MPI B52	40 ea	2 ea	2 x 200K	3.4	$1125
2 x MPI B92	80 ea	2 ea	2 x 400K	3.4	$1454

Dual-drive package includes two bare disk drives, self-contained dual-drive cabinet/power supply as illustrated, two drive cables and the version of Dosplus indicated.

NOTE: All 40 track drives are completely compatible with 35 track operating systems such as TRSDOS. DOSPLUS allows you to realise an additional 14% capacity compared with TRSDOS. Under DOSPLUS 3.4, 80 track drives can read 35/40 track diskettes.

All disk drive components are still available separately:

BARE DRIVES — MPI drives offer the fastest track-to-track access time (5 milliseconds) available. All drives are capable of operating in double density for 80% greater storage capacity.

	Price	Freight
MPI B51 40 track, single-head, 100K	$349	$5.00
MPI B52 40 track, dual-head, 200K	$449	$5.00
MPI B92 80 track, dual-head, 400K	$619	$5.00
Separate, dual-drive power supply	$85	$8.00

	Price	
Self-contained, single drive cabinet/power supply	$99	$5.00
Self-contained, dual-drive cabinet/power supply	$135	$5.00
Two drive cable	$39	$2.00
Four drive cable	$49	$2.00
DOSPLUS 3.4	$149.95	$2.00

Prices are FOB Adelaide. Add $5.00 freight for single drive package, $10.00 for dual-drive package. Prices are in Australian dollars. Freight is road freight anywhere in Australia.

All items carry a 90-day parts and labour warranty. Repairs to be carried out in our Adelaide workshops.

MICRO-80

LEVEL 2 ROM

ASSEMBLY LANGUAGE TOOLKIT

by Edwin Paay

FOR TRS-80 MODEL 1, MODEL 3 AND SYSTEM 80/VIDEO GENIE

This is a new package consisting of two invaluable components:

- **A ROM REFERENCE** Manual which catalogues, describes and cross-references the useful and usable ROM routines which you can incorporate into your own machine language or BASIC programs.
- **DBUG**, a machine language disassembling debugging program to speed up the development of your own machine language programs. DBUG is distributed on a cassette and may used from disk or cassette.

Part 1 of the ROM REFERENCE manual 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/Assembly tapes. There is a special section devoted to those additional routines in the TRS-80 Model 3 ROM. This is the first time this information has been made available, anywhere. Differences between the System 80/Video Genie are also described. Part 1 is organised into subject specific tables so that you can quickly locate all the routines to carry out a given function and then choose the one which meets your requirements.

Part 2 gives detailed information about each of the routines in the order in which they appear in the ROM. It describes their functions, explains how to use them in your own machine language programs and notes the effect of each on the various Z80 registers.

Part 2 also details the contents of system RAM and shows you how to intercept BASIC routines. 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!

The Appendices contain sample programmes 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.

DBUG: Eddy Paay was not satisfied with any of the commercially available debugging programs, so he developed his own. DBUG: allows you to single-step through your program; has a disassembler which disassembles the next instruction before executing it or allows you to bypass execution and pass on through the program, disassembling as you go; displays/edits memory in Hex or ASCII; allows Register editing; has the ability to read and write System tapes and all this on the bottom 3 lines of your screen, thus freeing the rest of the screen for program displays. Four versions of DBUG are included in the package to cope with different memory sizes.

The best news of all is the price. The complete Level 2 ROM ASSEMBLY LANGUAGE TOOLKIT is only:

— Aus. $29.95 + $2.00 p&p
— UK £18.00 + £1.00 p&p

SPECIAL OFFER TO OWNERS OF THE LEVEL II ROM REFERENCE MANUAL ...

UPGRADE TO THIS ASSEMBLY LANGUAGE TOOKIT FOR ONLY $19.95!

Send back your original Level II ROM Reference Manual plus a cheque, money order or Bankcard authorisation for $19.95 plus $2.00 p&p and we will send you the new ASSEMBLY LANGUAGE TOOLKIT

MICRO-80