

4/6/81

ISSUE 16 MARCH 1981

MICRO-80

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

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

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

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

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

Special bulk purchase rates are also available to computer shops etc. Please use the form in this issue to order your copy or subscription.

The purpose of MICRO-80 is to publish software and other information to help you get the most from your TRS-80, System 80 or Video Genie and their peripherals. MICRO-80 is in no way connected with either the Tandy or Dick Smith organisations.

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

Most of the information we publish is provided by our readers, to whom we pay royalties. An application form containing full details of how you can use your TRS-80 or System 80 to earn some extra income is included in every issue.

**** CONTENT ****

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

**** COPYRIGHT ****

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

**** LIABILITY ****

The programs and other articles in MICRO-80 are published in good faith and we do our utmost to ensure that they function as described. However, no liability can be accepted for the failure of any program or other article to function satisfactorily or for any consequential damages arising from their use for any purpose whatsoever.

***** CONTENTS *****

	<u>PAGE</u>
EDITORIAL	2
REVIEW OF SIMUTEK 1 PROGRAM	3
UPGRADING LEVEL II PROGRAMS FOR DISK BASIC	4
REVIEW OF DUNJONQUEST PROGRAM	4
MORE ABOUT SYSMON	5
READERS' REQUESTS	6
REVIEW OF PERCOM OS-80 DISK OPERATING SYSTEM	7
"WHY WASN'T I TOLD"	7
'80 USERS' GROUPS	9
MARKET PLACE	10
MICRO BUGS	10
<u>SOFTWARE SECTION</u>	
↔ BASIC MEMORY DUMPER.....L2/4K	11
COPYCAST.....L2/16K	13
↔ AGEING RELATIVELY.....L2/4K	16
FINANCE.....L1/4K	21
REMOVING LINEAR BRACKETS.....L1/4K	22
↔ TABLE TESTER.....L2/4K	24
↔ BOWLING.....L2/4K	26
↔ VICTORY AT BATHURST.....L2/16K	27
NEWS FROM MICRO-80 PRODUCTS	15
MICRO-80 PRODUCTS	17
NEXT MONTH'S ISSUE	35
CASSETTE EDITION	36
ORDER FORM	36

MICRO-80 is registered for posting in Australia as a publication - CATEGORY B
 AUSTRALIAN OFFICE AND EDITOR:-
 MICRO-80 P.O. BOX 213, GOODWOOD, SOUTH AUSTRALIA. 5034. TEL. (08) 211 7244

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

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

Published in Australia by MICRO-80, 433 Morphett Street, ADELAIDE.

***** EDITORIAL *****

Tandy held a press conference in Sydney recently, to which we (and every other publication remotely interested in computing) were cordially invited. Unfortunately, airline strikes and pressure of other work kept us in Adelaide at the appointed time, so we used the telephone the next day to find out what had happened. The purpose of the conference was to make some announcements concerning new products and new marketing services. Obviously, Tandy Australia were so impressed by our Review of the TRS-80 Model III in the last issue, that they have decided to bring it into Australia, and one of the most significant announcements at the press conference was the release and price of the Model III. Since this has been the subject of an expensive advertising campaign in the daily newspapers, it will probably come as no surprise to most of our readers to hear that the L2/16K Model III will sell in Australia for \$1399 whilst the 2 disk drive, 32K model with RS232 interface, will sell for \$3299. These prices are about 50% - 60% higher than their U.S. counterparts, only about 25% can be explained by extra freight, duty, sales tax, etc. So, once again, we are treated to high mark-ups on overseas equipment. At least the Tandy Corporation is in no danger of being accused of dumping their goods on the Australian market.

The high prices have gladdened a number of organisations around the country who had been poised ready to start importing directly from the U.S.A. themselves, unless Tandy's pricing structure made it uneconomical to do so. Look for some significant discounts on Model III's in a few months' time, particularly at the top end of the range where the cost of airfreight is a relatively smaller proportion of the total cost.

Other announcements concerned more customer support, which is good news. More computer centres will be opened and, where it is not economical to do so, some ordinary Tandy stores will be designated as "computer stores". They will continue to sell the normal range of Tandy products but will carry a full line of computers, peripherals, software, etc. and most importantly, will have sales staff specifically trained to deal with computers. We have never envied Tandy the task of supporting their products which have such a wide distribution. This move makes good sense and should benefit us all.

One item not mentioned at the conference was the release or otherwise in Australia of the TRS-80 colour computer. It is now almost 12 months since this computer was released in the U.S.A. It was described then by Tandy personnel (who had vigorously denied its very existence up to the last moment) as a "stop-gap" measure which, because it had been designed to work with the U.S.A. NTSC colour T.V. system and is therefore not compatible with the European/Australian PAL system, would never be released in Australia. The TRS-80 colour computer was originally developed to counter the TI personal computer which was a flop anyway, but has proved quite popular in the U.S.A. So, will it be released in Australia or not? Tandy is not saying but, if you want one of the first here, make sure you have saved enough pennies by October when our spies tell us that it will be released.

How will you be able to use a TRS-80 colour computer in Europe or Australia? We have not heard any whispers suggesting that a PAL version is on the way, although that could be a possibility. More likely, Tandy will also release its colour monitor over here, thus locking purchasers of TRS-80 colour computers into buying one of these too. There is an option, however. For the princely sum of \$16.00, a standard National portable T.V. can be converted to operate as a NTSC colour monitor. We know, we have a modified unit sitting in our living room right now. Closer to the time, we will publish a how to do it article.

Back to more mundane matters. Readers will have noticed a recent tendency amongst our contributors to include sound routines in many of the programs being submitted for publication. This is commendable, although so far, we have had to rely on other readers to work out modifications so that these programs will also give sound on SYSTEM 80's/Video Genie's through the external cassette port. Shortly, we will publish an article explaining how to write a program which gives sound on either system, at the user's option. The other noticeable feature of many of these programs is the way in which the machine language sound routines are integrated in the BASIC programs. In order to avoid protecting high memory, authors are using a technique pioneered by Eddy Paay in the File program (Issue 3, MICRO-80). The technique relies on the known position of the first line of BASIC code. A line 0 is entered. It starts with an unconditional jump to a higher line number and then the rest of the line is filled with spaces. A later line in the program POKEs a machine language program contained in DATA statements, into these spaces and accesses it as required via the USR function. It is indeed an ingenious way to avoid loading a SYSTEM tape as well as the BASIC tape, or having to protect high memory, etc. There is, however, one disadvantage which affects Disk users. The sound routine falls squarely in the DOS area and will not execute from Disk BASIC even if the normal cosmetic changes of converting USR to DEFUSR etc. are made. In Disk BASIC, Line 0 starts at 6A24H and the POKEd routine misses it entirely. Therefore, if you are writing such programs for publication, would you please do one of two things. Either:

- write the program such that it looks at the start of BASIC pointers in system RAM and POKEs a relocatable machine language routine into the appropriate place, OR
- at the very least, provide the source code of the machine language sub-routine for publication, so that Disk owners can change the origin themselves.

In this way, all our readers will be able to enjoy the real benefits of having sound in their programs.

***** REVIEW OF SIMUTEK ONE by R.J. Stehr *****

After trying for some time to get a copy of this SIMUTEK offering, I finally succeeded in getting one of what I suspect to be a new edition. There are shortcomings but I will go into that in greater detail later.

The price of the package is \$17.95 and for that you get a cassette and an instruction sheet in a plastic bag. The cassette is supplied loose and does not have the usual flip open storage box. I find this habit unjustifiable and most inconvenient. It means that you have to find another cast off box to protect your valuable software from dirt, etc.

The cassette itself has a computer printed label on one side only which announces that the tape is 'SIMUTEK PACKAGE ONE'; that it is copyright to Simutek and that there are copies (?) both sides.

Looking at the instruction sheet we find that there are five programs on the tape comprising: Graphic Trek 2000; Invasion Worg; Space Target; Saucers and finally, Star Wars.

Generally, liberal use is made of the CHR\$(23) command for titles etc. It could be quite effective on a TRS-80 but has the effect of showing the letters normal size, but double spaced, on a System-80. Four of the programs make good use of the graphics ability of the '80 and a lot of imagination has been put into the visual effects.

SPACE TARGET

In Space Target the player is required to launch missiles from the ground to shoot down aircraft flying overhead. If he is successful, the aircraft will crash and the pilot will parachute out. You can earn extra points for shooting down the pilot too!! I'm sure that must teach the kids something useful for later life. You are allowed 30 missiles with which to wreak as much havoc as possible.

GRAPHIC TREK 2000

The old familiar Star Trek complete with stars, star bases and the dreaded 'Klingons'. This version has more to offer the player than most, and I suspect that most of the other versions may have been based on this one originally. There are no instructions in the program, instead they occupy one and a half pages of the instruction sheet. Just as well too, because of the complexity of the game you will need them beside you while you play.

INVASION WORG

You have been elected General-in-Command of the Earth's forces and must use the assorted arsenal to repel the enemy invasion of Earth, Luna, Mars, Venus, etc. Another complex game for the 'Thinkers'. Comprehensive rules are contained in the program with a brief resume in the instruction leaflet.

SAUCERS AND STAR WARS

I have put these together because they are very similar. Both have the player looking through the windscreen of a 'fighter' with a manoevrable 'gun sight' to aim at the attacking enemy aircraft. The idea is to align the sight on the enemy and shoot them down. Star Wars goes a little further in that once you have disposed of a respectable number of ordinary aircraft, you get to have a crack at the evil 'Darth Vader' himself. He is a particularly elusive target and worth big points. The finale comes with the opportunity to drop a torpedo down the exhaust port of the Death Star.

I found the quality control of the product to be somewhat lax, really. The tape itself is excellent and loads beautifully at the volume setting I always use (about half). However, in Invasion Worg, I have found two errors in the program itself. One is an UL Error in line 4010 when the command GOT01 tells the '80 to go to a line that does not exist; and the other an SN Error in line 1420 with two consecutive commas.

My only other complaint concerns the lack of identification of the programs. They are not marked on the instruction sheet in the order they appear on the tape, nor is there any indication of where any of the programs start; so the first time you load each of them, log the counter reading for future reference.

All in all, I think that if you are after a space program package to add to your library, then this one would be very hard to beat and worthy of serious consideration.

***** FOR DISK BEGINNERS - UPGRADING LEVEL II PROGRAMS FOR DISK USE *****
by Bruce H. Bussenschutt

Some of us have decided that tape input/output is fine, but a bit slow, and we've gone the whole way and bought a disk unit. We've even survived the shock of having to buy an interface and extra RAM as well. Joy of joys, our programs load and save in seconds!

But what's this? Some of our Level II BASIC programs come up with SYNTAX ERROR IN LINE xxx when run on disk!! Eventually, we find that this happens with those BASIC programs which have a machine language subroutine POKEd into memory.

We discover that those programs have statement that perhaps look like these, and the error is in line 120 or 130;

```

100 MS=INT(30000/356)
110 LS=30000-MS*256
120 POKE 16526,LS
130 POKE 16527,MS

```

The clue, of course, is in lines 120 and 130. Whenever you see the numbers 16526 and 16527 in POKEs in a program, these are Level II pointers to the location of the machine language program, and they don't work in Disk Basic. The values after those locations give the commencing point of the machine language routine. MS is the Most Significant byte & LS is the Least Significant byte - in decimal equivalent of the hexadecimal value (it's simpler than it sounds).

- To convert this to Disk BASIC:

- (a) delete the POKE statements and the LS or MS calculations associated with them (if any).
- (b) substitute the DISK alternative method, which is to define-the-user subroutine entry point using DEFUSRn. If you are converting from Level II, use DEFUSRO to be consistent. (If you are defining one of several disk subroutines, n can be any number from 0 to 9 inclusive). The other half of the DEFUSRO statement specifies the machine language location. However, here the plot thickens a bit. Instead of using decimal numbers as the Level II POKE statements did, we now go back to using hexadecimal numbers and indicate that they are hex numbers by using the &H prefix. Our previous example location (dec.) was 30000. In hex this becomes 7530H, so the substituted statement is DEFUSRO=&H7530.
If you want to try an advanced calculation, MS from previously should have a decimal value equal to 75 hex and LS's decimal value should be equal to 30 hex. (DOS manual page 7-14).
- (c) the USR statement in Level II is of the form X=USR(0). This is just a little different in Disk Basic although often the existing form will work. Edit the line to insert another 0 so that it reads X=USRO(0). (Compare page 8-7f of Level II manual with page 7-20H of Disk manual).

That's all there is to this conversion. If you want a simple how-to-do-it formula, it probably could look like this:

- (i) Locate the POKE statements containing locations 16526 and 16527.
- (ii) Note the decimal numbers after these.
- (iii) Write down the decimal number after 16527 first, followed by the one after 16526.
- (iv) Convert each number to its hexadecimal equivalent, and write these down in the same order as (iii) above.
- (v) Delete the POKE statements completely from the program.
- (vi) Substitute them with the statement DEFUSRO=&Hxxxy where xx and yy represent the hex. numbers from (iv)
- (vii) Locate all the statements containing the command USR. They will probably be of the form X=USR(0).
- (viii) Add an extra 0 between the USR and (0), so that it reads X=USRO(0).
- (ix) Save your edited program, then RUN it. It should now work very nicely on disk.

- 000000 -

***** REVIEW OF "DUNJONQUEST" PROGRAM by Bob Sykes. *****

There are now several "Dunjonquest" programs available locally - this review concerns "Hellfire Warrior". I purchased my copy from Computerland in Grenfell Street, where others in the series were in stock.

The package contains the tape, loading instructions, two instruction sheets (one for quick scoring, another for game commands) and a very necessary 64 page booklet....and one more item, which I found quite annoying - an erratum slip containing corrections for the Dunjonquest program (which has to be loaded then corrected, then "C SAVE D" on another tape!!). This is a continuing source of irritation, because during the game one has to change the tape whenever the instruction "FLIP TAPE, REWIND, PRESS 'PLAY' AND ENTER" appears. Why pre-production corrections could not have been made is difficult to comprehend!

Now for the game.....

Those of you who are familiar with the game of "Dungeons and Dragons" should quickly feel at home - the major difference is that only one character can enter the dungeon in this computer game, compared to the many characters in D & D.

For players not familiar with D & D you MUST spend time reading the booklet (and quite a well written booklet it is)! Even D & D experts should give some time to reading here, since there are significant variations from the board game.

"Hellfire Warrior" commences at dungeon level 5 (the previous four levels being covered in earlier Dunjonquest games, e.g. "The Temple of Apshai"). Upon first entering the Innkeeper side of the cassette, one has to establish the character's attributes, i.e. Intelligence, Intuition, Ego, Strength, Constitution and amount of money. On the first game or two, the Innkeeper will do this randomly for you - after you gain experience at the game, you may enter these attributes yourself (please try to resist the temptation to cheat - it does spoil the game)!

Then the character may visit in turn, the Armory, Apothecary and Magic Shoppe (the choice being yours by answering Yea or Nay to the Innkeeper's questions). Thus, suitably attired for the fray (wait for the data files to be completed) you may enter the dungeon....(FLIP TAPE, REWIND, PRESS 'PLAY' AND ENTER). Personally I find the bargaining with the three personages in the shops almost as entertaining as the journey through the dungeon, although after a few times the limits are certainly clear.

You may enter dungeon level 5 to 8 - on your first attempts (yes, plural, because the several beasties WILL kill you) you are really learning the techniques of the game. On the screen a large-scale map of a portion of the dungeon appears, with your character's position clearly evident.

On the right hand side of the screen is a continually updated scorecard - watch it for fatigue, wounds, treasure etc. With the various weapons you have purchased, and your own skill, you aim to accumulate as much treasure as possible, and get out of the dungeon with it.

If you can avoid being killed, your character will gain experience, and so the various attributes will alter. By negotiating with the Innkeeper (a hard bargainer) your treasures are converted to cash, and so more potent weapons and magic become available to you.

Your ultimate aim is to rescue Brynhild, the warrior maiden trapped on level 8, "The Plains of Hell".

If you have a 16K cassette version, keep pencil and paper with you to jot down the attributes and treasure of your character (the disk version allows you to save all of this on disk).

Several people have played the game on my TRS-80, and all have enjoyed it (from my 13 year old son, to a very experienced Dungeons and Dragons Dungeon Master) - all that is required is time (which really flies - a 6 to 8 hours sitting is not unusual). Of course, it doesn't make for brilliant conversation, but I have found this game to be a whole lot of fun!

Each game is different, because you choose the site of action, and whether to fight or run! And as your character increases his/her potency, the journeys into the lower levels of the dungeon become more entertaining (and time consuming).

To sum up - a real break from the various space-war games, not played on an x by y grid. It involves imagination and skill. Real value!

- 000000 -

***** MORE ABOUT SYSMON ***** by R.J. Stehr.

This article is intended for those who are not too sure of their way around machine language. More specifically, it is intended to explain how to change from BMON to SYSCOPY and back again if the two programs have been integrated into one as described in December 80, MICRO-80. Assuming firstly that you have combined the two programs and have called the result 'SYSMON', and secondly, that the programs are all working the way they should, you may proceed as follows:

Switch on and answer the prompt-READY? with 28860 (enter)

Type and enter SYSTEM

Type and enter SYSMON

When it is loaded, and in response to *? type and enter /28944 to send the program to SYSCOPY and /31641 to send the program to BMON.

So far so good. It shouldn't be anything you haven't done a thousand times before. Here comes the good part:

To go from BMON to SYSCOPY type G for Goto Hex Location.

In response to Hex Address enter 7110

In response to the first *? type and enter /28944

In response to the second *? type and enter the name of the program in which you are interested. You will then find that it will load the program and give the Start End and Entry points as it is supposed to. At the end of that routine you will find that it will give the familiar *? prompt again ready for you to enter another program; if you enter /31641 it will return you to the BMON program.

That about covers it. Any in-between stages are merely adaptations of the above. So if you start on the opposite program to the one I did then carry on from the appropriate place above.

- 00000000 -

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

This column is a regular feature of MICRO-80. In it, we list all those articles, programs, etc. requested by our readers. We invite contributions from Readers to satisfy these requests and will, of course, pay a publication fee for all articles, programs etc. printed. As a guide, we will pay a minimum publication fee of \$10 for any article or review published. In the case of software reviews, we will aim to pay in accordance with the value of the program, up to a maximum of \$25. So, if you write a good review which we publish and the usual selling price of the program in Australia is \$19.95, then we would pay you \$20. In that way, the successful reviewer will get the program he reviews, free. (Make sure you include the selling price in your review). Unfortunately, we cannot afford that policy on hardware(!) so we will pay in accordance with the merits of the review - generally of the order of \$25. Submission of a review for publication automatically means that you are prepared to accept the figure we decide to pay you and no correspondence will be entered into. Payment will be made within 30 days of publication.

** ARTICLES **

- File handling on the '80
- Description of the functions performed by the Expansion Interface
- Reviews of '80 compatible printers
- Reviews of commercially available software (including that produced by us!)
- Reviews of commercially available hardware
- * How to SAVE onto Disk, programs such as Analogue clock and Touchtype
- * A master index to the appropriate sections in the Tandy Manuals in Level I, Level II, DOS etc.
- Comparative reviews of disk drives
- How to convert a Level I program to Level II
- A simple guide to using Level I Arrays
- * Review of Dunjonquest program
- An explanation of how to make full use of USR, PEEK and POKE statements
- Discussion of the various electric fields produced by the keyboard, tape recorder, monitor disk drives etc., how to measure them, how important they are and how to combat them

** SOFTWARE **

- * A m.l. program to enable the break key to work like RESET when using an expansion interface
- Stock market program
- * Horse racing system
- "Files" program modified for 48K system
- Morse code decoder
- Sub-routine Forum
- Program to "SET" non-graphical symbols
- A new STAR-TREK game
- Conversational programs (like Eliza)
- 3D programs (such as a maze seen from the inside)
- Program to scroll a Level II listing, one line at a time
- Modification to SCRIPSIT which will enable it to output to the SYSTEM 80 printer port
- Double Precision Trig, Log and Exponential machine language routines

** HARDWARE **

- Interfacing the '80 to external hardware
- Review on the performance of line filters
- Real Time clock
- * Radio Teletype/Morse interfacing
- RFI (Radio Frequency Interference) suppression
- Interface for a Teletype printer

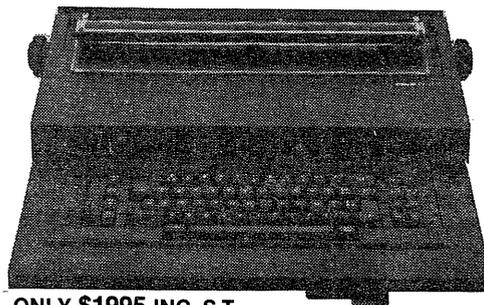
Note:- An * denotes that we already have some suitable material on hand for this topic.

- 0000000000 -

Daisy Wheel Typewriter/Printer

MICRO-80 has converted the new OLIVETTI ET-121 DAISY WHEEL typewriter to work with the TRS-80 and SYSTEM 80 or any other microcomputer with a Centronics parallel port (RS 232 serial interface available shortly). The ET-121 typewriter is renowned for its high quality, fast speed (17 c.p.s.), quietness and reliability. MICRO-80 is renowned for its knowledge of the TRS-80/SYSTEM 80 and its sensible pricing policy. Together, we have produced a dual-purpose machine:- an attractive, modern, correcting typewriter which doubles as a correspondence quality Daisy-wheel printer when used with your micro-computer.

How good is it? - This part of our advertisement was typeset using an ET-121 driven by a TRS-80. Write and ask for full details.



ONLY \$1995 INC. S.T.

PERCOM OS-80 DISK OPERATING SYSTEM - Reviewed by Michael Cooper

The Percom Disk Operating System OS-80 is a memory efficient, high speed program which extends Level 2 BASIC to permit Disk Drive control. The DOS occupies 7K of low user RAM and provides all the necessary commands to enable the user to write programs using Disk BASIC. Due to its small size OS-80 enables a 16K single disk system to be both viable and useful.

To compact the DOS as much as possible it contains no file handling routines or directory track. Instead a type of disk mapping is used. This technique enables the user to specify the drive (0 - 3) and sector (0 - 399 on a 40 track drive) as a 5 digit integer. The first digit specifies the drive, the next digit is always 0 and the last three digits specify the sector. For example, if you wished to save a BASIC program to Disk Drive 1, starting at sector 125, the command would be SAVE 10125. The OS-80 system would then save the program and finally display the number of the last sector used. To load the same program into your "80" you command LOAD 10125 (,R). This would cause the loading of the BASIC program stored on the disk in Drive 1, starting at sector 125. The (,R) is optional, but if included causes the program to be loaded then RUN. The OS-80 system also includes a MERGE command to allow a program on disk to be merged with one in memory.

The OS-80 has a number of system commands, all using the CMD verb to perform the various functions.

```
CMD"F",D is used to format the disk in drive D.
CMD"M",D dumps a core image of DOS onto drive D, sectors 0 - 19.
CMD"K",D combines the above two commands.
CMD"H",A$ allows the user to input a text string of up to 128 characters, which
appears during the booting of DOS.
CMD"K",A$ allows for a chaining mode, so that BASIC programs can be loaded and
executed from the boot.
```

OS-80 also has 3 disk I/O statements to enable the processing of data stored on disk. GET reads the specified disk sector either into a buffer or into a string variable depending on the command syntax, whilst PUT writes either a string variable or the contents of a buffer to the specified sector. For example, PUT 25,A\$ will save the contents of A\$ to sector 25 on drive 0 (please note that users with a single drive only have to specify the sector). GET #1,IX% will read the sector and drive specified by the value of variable IX%, into I/O buffer number one. Unlike TRS-DOS the number of buffers is fixed at 4. However I have never needed to use any more than that in the most complex program. Having got the data into buffer 1, the user can FIELD it up into records as in TRS-DOS Disk BASIC.

By using the DOS Vector Table in ROM, the OS-80 system links into Level 2 and implements most of the TRS-DOS Disk Basic conversion and placement functions, usually with identical syntax. Included are LSET, RSET, MKI\$, MKS\$, MKD\$, CVI, CVS, CVD, INSTR, MID\$, DEF FN, DEFUSER and LINEINPUT.

Hexadecimal constants can also be used for all arguments, but not Octal. OS-80 allows the user to easily tailor the DOS to the system requirements. To change the number of tracks OS-80 will process, just POKE the number into location 17554. To change the track to track stepping time, POKE the value into 17556. The OS-80 system disk supplied is set for Pertec drives as supplied by Dick Smith (40 tracks with 20 millisecond steps).

I would recommend OS-80 for both beginners to disks and the seasoned programmer as it offers both simplicity and versatility. The system disk is supplied with utility programs to perform simple file handling, a program of disk utilities such as FORMAT, BACKUP, COPY, FREE, VERIFY, DU:IP and ERASE and a program called "The 5 1/4 inch notebook" which is a simple data manager. Unfortunately, disks are not compatible with those made under TRS-DOS as OS-80 has no Directory track but except for minor syntax variations, most TRS-DOS Disk Basic programs will run under OS-80. The reward for users converting TRS-DOS Disk BASIC programs is faster disk I/O and no potential for disks with "zapped and unusable" directories. The major job in the conversion is the removal of all the OPEN and CLOSE verbs and modification of file record logic to encompass OS-80 sector read/writes.

If you haven't a copy of OS-80 then I strongly urge you to buy one and use it. You will find it faster than either TRS-DOS or NEWDOS-80 and for the novice, much simpler to use. The only drawback it suffers is its limitation to BASIC program and data storage, but at the price, it is one of the true software bargains.

- 0000000000 -

***** WHY WASN'T I TOLD?

by Noel Rossiter *****

An article in ETI for April 1981 entitled "Uncovering the Z80" does, in fact, uncover some surprising news regarding the Z80 micro, and gives details of a BASIC program by means of which you can test your own machine. The news is that most, if not all, machines have access to a number of further machine language instructions which make the machine even more versatile than we thought.

In particular, it is possible to regard the two index registers, IX and IY, not only as index registers but each can be used as a pair of independent 8-bit registers capable of being separately addressed. For convenience, let us call the upper 8 bits of the IX register the HX register, and the lower 8 bits the LX register; in similar manner, the IY register can be divided into the HY and LY registers.

Now, if I want to load the H register with hex 23, I use the machine language hex codes 26 23; if I want to load the same code into the HX register, I prefix the code with hex DD giving DD 26 23 (source code LD HX,23H). The prefix for the HY register is FD, so that FD 26 23 loads the hex code 23 into the HY register corresponding to the source code LD HY,23H if such a source code existed.

These are about all the rules you have to know. To operate on the HX or LX registers, find the code for the corresponding operation on the H or L register, as the case may be, and prefix it with the group DD. To operate on the HY or LY registers use the H or L operation code prefixed by FD. There are four codes where the prefixes have a slightly different effect. Thus:

64	is	LD H,H	but	DD 64	is	LD HX,HX	and	FD 64	is	LD HY,HY
65		LD H,L		DD 65		LD HX,LX		FD 65		LD HY,LY
6C		LD L,H		DD 6C		LD LX,HX		FD 6C		LD LY,HY
6D		LD L,L		DD 6D		LD LX,LX		FD 6D		LD LY,LY

You can thus load any of registers A,B,C,D or E from HX,LX,HY, and LY and vice versa. You can load HX from LX and HY from LY and vice versa; what you can't do is to load H or L registers to or from the four half index registers, or to cross load from half the IX register to half the IY register (or vice versa) directly.

Apart from loading, you can INC, DEC, ADD, SUB, ADC, SBC, XOR, AND, OR and CP each half of the index registers.

That is as far as the article in ETI goes, but as I have an excellent machine language training program which displays all the registers in both binary and hex, and accepts machine language codes to operate on them, I have been able to test all the above on my machine and verify that the operations happen in fact. In addition, I have been able to investigate the effect of using the prefixes in front of the group of codes that start with CB; this is slightly more complex.

To begin with, the prefixed codes do not address the half registers or even the index registers themselves; rather the operation is performed on the contents of the address being pointed to by the register. This is consistent with the published codes, where it will be seen that CB 06 is RLC (HL) and DD CB 00 06 is RLC (IX+). This gives the general format for the prefixed codes compared with the original - if the original code is CB yy then the format of the prefixed code is DD (or FD) CB xx yy where xx is the indexing displacement of the addressed location from the address being pointed to by the index register.

The CB codes form a number of groups. At the top are the SET group (CB C0 to CB FF). CB C0 is SET 0,B - SET bit 0 of the number in the B register. The corresponding prefixed code for the IX register is DD CB xx C0, which has the effect of SET 0, (IX+xx) followed by LD B,(IX+xx) i.e. bit 0 of the number stored at (IX+xx) is set, and the changed number loaded into register B (of course it is also still stored at the original address).

The same applies to the RES group from CB 80 to CB BF. The code CB 8A (RES 1,D) becomes, for example, FD CB xx 8A and has the effect of RES 1,(IY+xx) followed by LD D,(IY+xx).

The next group, (BIT) from CB 40 to CB 7F are different. So far as I can see, DD CB xx 40 has no different effect to DD CB xx 46 which is the published source code for BIT 0,(IX+xx). Similarly, throughout the group and I can't see any advantage here with the unpublished codes that can't be obtained just as readily with a published code.

The group from CB 00 to CB 3F, which cover the RLC, RRC, RL, RR, SLA, SRA and SRL operations behave like the SET and RES groups. Thus, CB 01 is RLC B - DD CB xx 01 is equivalent to RLC (IX+xx) followed by LD B,(IX+xx).

Incidentally, although I have used pseudo source codes like LD HY,HX above, this is purely as an aid to explanation; don't try to feed them to your assembler, as I haven't found an assembler which will support them. If you want to incorporate one of the unpublished instructions in your assembly language program, the best way I have found is to use DEFW instructions.

Well, there it is. It seems that apart from the 696 possible source codes that have been published, there are, in fact, at least another 414 usable codes available to the machine language programmer. Quite a number of these are potentially useful, and in some applications the extra four 8-bit registers made available by the use of the codes would be a boon. I can only repeat the title of this article - why wasn't I told? It may be, of course, that these codes, or some of them, are not factory tested and cannot therefore be relied on as being available in any particular chip. However, with a little ingenuity, the user can confirm whether they exist in his machine, and use them in good health.

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

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

** AUSTRALIA **

- BRISBANE: Contact: Mr. Lance Lawes,
Tel: Home (07)396 2998
Bus. (07)268 1191 Ext.15
MEETINGS 1st Sunday of the month at 2 p.m. at 21 Rodney St. Lindum, 4178.
- MELBOURNE: NORTHERN AND WESTERN SUBURBS
Contact: Mr. David Coupe (03)370 9590
MEETINGS: C.P.M. Data Systems, 284 Union Road, Moonee Ponds.
Alternate Thursdays at 7 p.m.
- EASTERN SUBURBS - 1
Contact: Mr. John Fletcher, 89 0677 bet. 9-4.
- EASTERN SUBURBS - 2
MEETINGS: 3rd Wednesday of the month, Kingswood College, 355 Station St. Box Hill.
- FRANKSTON: PENINSULAR GROUP
(Vic.) MEETINGS: 2nd Tues. of the month (except Jan.)
Contact: M.G. Thompson (03)772 2674
- GEELONG: GEELONG COMPUTER CLUB
MEETINGS: 2nd Tues. of the month at TYBAR Engineering, Hampton St. Newtown.
Contact: The Geelong Computer Club,
P.O. Box 6, Geelong, 3220.
- DARWIN: Contact: Tony Domigan, P.O. Box 39086,
Winnellie, N.T. 5789.
- ADELAIDE: Contact: Rod Stevenson, 51 5241 bet. 9-4. 36 Sturt Street, Adelaide. 5000.
- CANBERRA: Contact: Bill Cushing, 10 Urambi Village, Kambah ACT. 2902 (PH. 31 6399).
MEETINGS: 3rd Thursday of each month at 7.30 p.m. in:
Urambi Village Community Centre, Crozier Circuit, Kambah.

** UNITED KINGDOM **

- National Users Group in U.K.
Brian Pain, 40 High Street, Stoney Stratford, Milton, Keynes.
- Level 1 User Group,
Secretary: Mr. N. Rushton, 123 Roughwood Dr. Northwood, Kirkby, Merseyside,
L33 9UG.
- NEWCASTLE: Contact: John Stephen Bone 0632 770036
NPCS (Newcastle Personal Computer Society)

** NEW ZEALAND **

- AUCKLAND: Contact: Ron Feasy 799 366 (Bus.) 469 455 (Priv.)
MEETINGS: 1st Tuesday of each month, 7.30 p.m. at:
NZ Solenoid Co. Ltd.,
28 Kalmia Street,
Ellerslie, Auckland.

- 000000000 -

** FROM OUR LEVEL I EDITOR, Michael Svensdotter:

If a PROM is a programmable read only memory, what is a PRAM?

Answer next month.

***** MARKET PLACE *****

Market place is available to any reader who has hardware to dispose of. An entry costs nothing - you pay MICRO-80 \$5.00 or 5% commission, whichever is the greater - up to a maximum of \$30, after the goods are sold. The commission is calculated on your advertised price.

KTM-2 KEYBOARD UNIT in good order	\$235.00	Mr. A.R. Hall, c/o Post Office, Southbrook, Qld. 4352.

AXIOM SERIAL THERMAL PRINTER with 8 rolls of paper and TRS-232 interface	\$400.00	Mr. T. Domigan P.O. Box 390986, Winnellie, N.T. 5789.

SYSTEM-80, LEVEL II 16K MEMORY, GAMES, CASSETTES, BOOKS AND MANUALS, brand new	\$630.00	Mr. M. Gosbee, Ph. (02) 88-4468

SYSTEM-80 L2/16K with full documentation, approx. 40 blank cassettes (data), books, magazines, and programs (mostly games) including EDTASM PLUS and Adventure No.9.	\$720.00 c.n.o.	Mr. R. Glucz, i/1707 Dandenong Rd Oakleigh, VIC. 3166 Ph. (03)543 3302 AH.

TRS-80 MODEL 1 Level II 4K excellent condition. Catalogue price \$879. Sell with manuals and programs	\$750.00	James Grigg Ph.(052) 78 5056.

Radio Shack QUICK PRINTER: 150 LPM on 4.75" wide aluminised paper. Software selectable 80, 40 or 20 characters per line. Upper-case with full descenders. Automatic underlining. Seldom used and in excellent working order. Catalogue price \$699. Will sell for:	\$400.00	K. Cook, P.O. Box 931, GOVE, N.T. 5797.

SYSTEM-80 L2/16K with over \$100 in programs, including BMON, RPN Calc, Sargon 3.5 Chess, Space Invaders, 70 Maths/Finance Programs. Excellent condition	\$680.00	G. Hobbes, 42 Nareen Parade, NARABEEN, NSW.2101 Ph. (02) 913 2264
SERIAL PRINTER INTERFACE	\$ 40.00	

SYSTEM-80 Level 2 16K with input level indicator, right arrow and clear keys, I/P changeover switch plus Emerson cassette recorder for external input, plus 12" BW T.V. monitor with video I/P conversion, plus 13 months of MICRO-80 issues and cassettes, plus 10 Scotch computer cassettes	\$700.00 the lot	Mr. B. Cameron, 15 George Street, PENNANT HILLS, N.S.W. 2120.

***** MICRO-BUGS *****

In which we correct those errors which seem to creep in, no matter how careful we are.

SEA WOLF - Issue 15 Page 16. This program contains a machine language sound routine POKEd into Line 0 when the program first RUNs. Unfortunately, we omitted to tell you in the text to leave room for the m/l routine when you type in Line 0. The correct sequence is to type:-

0 GOT010:

then fill the rest of the line with spaces until the cursor will move no further. If you LIST the program after it has RUN Line 0 will contain meaningless statements. This is normal as these are the tokens etc. which correspond to the machine language code which now resides there. If you do not leave room for this code, the program will POKE its m/l routine over BASIC code, thus destroying itself.

***** BASIC MEMORY DUMPER LII/4K (C) Ken Shillito *****

BASIC Memory Dumper may be used to convert SYSTEM to CLOAD tapes, amongst other things.

When a program makes extensive use of graphics or number crunching, it is often desirable to use machine code. Now whilst machine code is very useful for such purposes, it is very laborious, inefficient at interacting with the user and difficult to customize. Moreover, SYSTEM tapes are said to be less reliable than CLOAD tapes. A common compromise solution is to create DATA statements in a BASIC program which are POKED into memory creating a machine code area that may be called with USR statements. A hybrid Basic/Z80 program is then built, which has the advantages of both. This otherwise attractive approach suffers from the following defects:

1. It eats up enormous quantities of memory.
2. It causes an awkward pause while machine code is POKEd.
3. It is even more boring translating machine code to decimal.
4. It is even more boring typing the program into DATA statements.
5. If you don't have a printer, you have to transcribe the assembler output by hand.

Problems 2-5 are wholly solved and problem 1. is 40% solved with the BASIC Memory Dumper.

The program is used as follows:

1. Load into memory, (by Monitor, System or POKEing), the machine code (or perhaps DATA) to be dumped.
2. CLOAD the BASIC MEMORY DUMPER (hereafter referred to as the "input" program), and RUN it.
3. The "input" program prints what it does and then prompts for :
 - the starting address (in decimal) of the area to be dumped.
 - the number of consecutive 64 byte blocks to be dumped (maximum 256, ie. 16K).
 - the starting address of a 186 byte area (in decimal) to be used as a workspace. (This would normally be in high order memory).
4. The "input" program then destroys itself and creates an "output" BASIC program. The computer requests you to EDIT lines 20-30 of the "output" program. Line 30 contains a REM statement giving details of what is required.
5. The "output" program consists of:
 - line 0 as per the "input", which may now be edited.
 - line 10, a 69 byte DATA statement which must not be changed.
 - line 20, a statement for POKEing the line 10 DATA into memory. You must EDIT this line to replace the asterisks by the starting and ending address of an area that will be free in your finished program, consisting of 69 bytes.
 - line 30 which POKES the entry address, i.e. the starting address in line 20, to the USR pointers and a USR call with a calling argument equal to the decimal address from which the dumped memory is to be recreated. The re-creation address need not be the same as the source address. (Of course, the program or DATA must be relocatable for different addresses to give meaningful results). The addresses appear in the statement as asterisks which must be replaced with numbers by EDITing.
 - a series of REM# statements starting from line 40 which look at first glance like a bad load, but actually contain the memory blocks dumped in hexadecimal notation, i.e. 1 line for each 64 byte block.
6. When lines 20-30 of the "output" program are RUN they re-create the memory previously dumped, starting at the address specified by line 30.

A TYPICAL APPLICATION

- # You assemble a Z80 subroutine to reside in locations 30000 to 32768 (decimal).
- # You specify READY/MEMORY SIZE? 29812 on returning to BASIC (4K system owners would smaller numbers). Then use SYSTEM to load your subroutine.
- # You CLOAD the BASIC MEMORY DUMPER and RUN.
- # You answer the prompts as follows:
 - No. 64 byte blocks : 44
 - Starting address : 30000
 - Work area : 29814

- # The program then takes less than 1 second to create the output program and destroy itself. Lines 40 to 470 contain the HEX representation of your program.
- # You EDIT lines 20-30 to read :
- ```

20 FOR I=29931 TO 29999:READ K:POKE I,K:NEXT
30 POKE 16526,235:POKE 16527, 116:USR(30000)

```
- # Optionally, you may remove the unwanted bytes from the end of line 470 if memory space is a problem.
- # Edit Line 0 to document your program and type the BASIC part after 470.
- # To use your program, specify READY/MEMORYSIZE? 29930.
- # When you RUN your program it will take less than 1 second to recreate any amount of machine code.
- # If you use the program to change a system tape written by someone else into a CLOAD tape you will need to make sure that the BASIC and Z80 program do not interfere. This may involve relocating BASIC or preferably, if possible the Z80 part. You will need to know what you are doing. The output program, once created, will merely need a USR call to the Z80 entry address. The Vendor of the program should be able to tell you if it is relocatable, or you could try it and see what happens.

#### TECHNICAL SPECIFICATIONS

The output program must not contain REM# anywhere except when followed by HEX code to be POKEd. Lines 20 and 30 will find all occurrences of REM# whatsoever, even if halfway through lines. REM # separated by a space will not be "found" however. This program is an exception to the rule that REM statements are ignored. The memory to be dumped can be anywhere, even ROM or non-existent memory! The latter will return 00H. The re-creation of memory will also take place anywhere; the computer will impotently imagine itself to be POKeing bytes into ROM or non-existent addresses. It will POKE itself to death, if you tell it to!

The memory to be dumped is limited to 256 blocks (i.e. 16K), though in practice the limit is set by the need for re-creation space plus space for BASIC storage (say, 75 blocks in a 16K system or 18 blocks in a 4K). There is no limit to the amount of memory to be re-created. The re-creation code will find the beginning and end of BASIC, regardless of any fancy patches or whatever you may have installed. Each 64 byte block dumped requires 135 bytes for storage in BASIC, as opposed to about 230 bytes in DATA statements. The REM# statements can be edited to allow for patching up, etc. A re-dump would be safer, however.

If you want to be really tricky, you can POKE machine code over the top of BASIC. If you do this, remove line 30 to a line number higher than the REM# statements and remember that the program cannot be reRUN or LISTed, etc. In fact, the only safe command after POKeing on top of a BASIC program is NEW, once it has finished execution. The approach does have the advantage of up to 50% saving in memory.

#### HOW IT WORKS.

The program consists of three parts, the BASIC part which is self-explanatory, the machine code in line 10 which is the memory re-creation code, and the code in lines 170-180 which is the memory dumping routine. The recreation begins by calling a ROM routine which dumps the end of the BASIC program address into HL (actually, that is not what the routine is for but it works). The dumping code is more straightforward. For an explanation of how the "fix BASIC pointers" section works, you should consult Eddy Paay's Level II ROM Reference Manual.

#### LISTINGS

\*\*\*\*\*CAUTION\*\*\*\*\*

The first 4 lines must be typed EXACTLY as shown, not a single character may be out of place, even in the REM, or it won't work.

```

0 REM MEMORY DUMPER
10 DATA 205,127,10,235,217,205,248,26,237,91,164,64,183,237,82,229,193,235,62,14
7,237,177,192,126,254,35,32,12,229,217,225,14,0,35,126,254,0,32,3,217,24,232,254
.65,48,4,214,48,24,2,214,55,203,65,32,8,135,135,135,135,71,12,24,225,128,18,19,2
4.218
20 REM FOR I=**** TO ***+68:READ K:POKE I,K:NEXT
30 REM POKE 16526,***:POKE 16527,**:I=USR(****):REM REPLACE *'S BY 69 BYTTEE WO
RK & POKE ADDRESSES
40 CLS:PRINT "*** MEMORY DUMPER ***":PRINT
50 PRINT "WHEN THIS PROGRAM IS RUN, IT DESTROYS ITSELF AND CREATES A NEW PROGRA
M. THE OLD PROGRAM MUST NOT BE ALTERED IN ANY WAY UNTIL THE NEW ONE IS CREATED

```

```

60 PRINT " THE OLD PROGRAM GRABS UP TO 4096 BYTES FROM MEMORY, IN 64 BYTE BL
OCKS, AND DUMPS ASCII CODE IN HEX NOTATION INTO PROGRAM LINES WITH REM & #. THE
NEW PROGRAM, WHEN RUN, FINDS *ALL* LINES WITH REM # BUT NO";
70 PRINT " INTERVENING SPACE, TRANSLATES FROM ASCII TO HEX, AND DUMPS INTO MEMO
RY."
80 PRINT:PRINT"PRESS A TO CONTINUE":A$=INKEY$:PRINT
90 IF INKEY$("<"A" THEN 90
100 INPUT "HOW MANY 64 BYTE BLOCKS ARE TO BE GRABBED (MAX 256)";Z
110 INPUT"ADDRESS (IN DECIMAL) OF FIRST BYTE TO BE GRABBED";J
115 INPUT"STARTING ADDRESS OF A 186 BYTE SPACE I CAN USE AS A WORKSPACE";W
120 Q=PEEK(16548)+256*PEEK(16549):POKE Q+268,31:POKE Q+316,31:PRINT:PRINT"EDIT L
INES 20 TO 30":PRINT
130 POKE 16612,Z-INT(Z/256)
140 FOR I=W TO W+68:READ K:POKE I,K:NEXT
150 WB=INT(W/256):POKE 16526,W-WB*256:POKE 16527,WB:B=USR(W+69)
160 WB=INT((W+69)/256):POKE 16526,W+69-WB*256:POKE 16527,WB:B=USR(J)
170 REM#CD7F0AEB3AE440019D012AA44009D9212800010A00D94FD5EB21870019EB732372D1E5D9
EBE12373237223E5EB09D9E136932336232306401A13F50F0F0FE6
180 REM#0FFE0A3802C607C6307723F1E60FFE0A3802C607C630772310DE3600230D20B736002336
0023360022F94022FB4022FD40E1C3CC06

```

\*\*\*\*\* COPYCAST LII/16K (C) D & G \*\*\*\*\*

The purpose of this program is to provide a fast and reasonably accurate method of estimating how much space in a booklet or magazine a certain number of words will fill when typeset, or conversely, how many words will be needed to fill a certain space. As such, it will be of use to journalists, graphic designers and the editors of any typeset magazine. At the outset, the user must specify the units of measurement he uses, then the page layout, before going on to specify a typeface. Most common text typefaces are included, together with their usually available variations such as italic, bold and so on. More may be added by inserting new DATA statements starting at line 1000. These statements have the following format: the first string is the type font and the next number is the number of variations included. The number after each variation is the number of lower case characters which will fit into one inch if set in 6pt type. This figure is usually included in typesetters' sample books. For most text, you should find the program accurate to within 5% but if you are dealing with text which normally includes many long words, such as scientific subjects, you may find the program continually underestimates. If this is so, change the figure in line 520, currently set to 5.8 (the average number of characters in each word plus 1 for space) to a more appropriate figure.

```

10 FT$="NONE":TS$="SO FAR"
20 CLEAR150
30 CLS:DIM ST$(16),CC(16)
40 PRINT"* COPYCAST *":PRINT"BY DAVID GRIGG"
50 PRINT:PRINT"DO YOU USE:":PRINT:PRINT"1. INCHES ONLY":PRINT"2. INCHES AND PRIN
TER'S EMS":PRINT"3. METRIC UNITS?"
60 PRINT:INPUT "ENTER APPROPRIATE NUMBER";N
70 ON N GOTO80,90,100
80 F1=1:F2=0.166666:U1$="INCHES":U2$="INCHES":GOTO110
90 F1=1:F2=1:U1$="INCHES":U2$="EMS":GOTO110
100 F1=2.54:F2=4.233333:U1$="CENTIMETRES":U2$="MILLIMETRES"
110 CLS
120 PRINT"PAGE FORMAT DETAILS":PRINT:PRINT"COLUMN WIDTH IN ";U2$::INPUTCW
130 INPUT "NUMBER OF COLUMNS PER PAGE";CP:PRINT"DEPTH OF TEXT AREA OF PAGE IN ";
U1$::INPUT DP:DW=CW/F2:DP=DP/F1
140 CLS:PRINT"OPTIONS":PRINT:PRINT"1. ENTER OR CHANGE TYPE FOUNT DETAILS":PRINT"
2. DISPLAY TYPE FOUNT NAMES":PRINT"3. CHANGE PAGE FORMAT DETAILS":PRINT"4. FROM
WORDCOUNT, DERIVE COPY SPACE NEEDED":PRINT"5. FROM COPY SPACE AVAILABLE, DERIVE
WORDCOUNT NEEDED"
150 INPUT "WHICH OPTION";N:ON N GOTO170,290,160,340,440:GOTO140
160 GOSUB560:GOTO120
170 CLS:INPUT "NAME OF TYPE FOUNT";Q$
180 RESTORE:FOR N=1TO1000:READ FT$,NO:IF FT$="END" THEN220
190 FOR K=1TONO:READ ST$(K),CC(K):NEXTK
200 IF FT$=Q$ THEN230
210 NEXT N
220 PRINT"TYPE FOUNT NOT FOUND":FORW=1TO100:NEXTW:GOTO290
230 CLS:PRINT"THE FOLLOWING TYPE STYLES ARE AVAILABLE IN ";FT$
240 FOR K=1TONO:PRINT K;TAB(5);ST$(K):FORZZ=1TO100:NEXTZZ,K
250 INPUT"ENTER NUMBER OF STYLE REQUIRED";K:CS=CC(K):TS$=ST$(K)
260 INPUT "WHAT POINT SIZE OF TYPE";PS
270 INPUT "LEADED BY HOW MANY POINTS";LD

```

```

280 GOTO140
290 CLS:PRINT"THE FOLLOWING TYPE FONTS ARE AVAILABLE IN THIS PROGRAM"
300 RESTORE:FOR N=1TO1000:READ FT$,NO:IF FT$="END" THEN330
310 PRINT FT$,:FOR K=1 TO NO:READ ST$(K),CC(K):NEXT K
320 NEXT N
330 PRINT:INPUT "HIT ENTER TO CONTINUE";Q$:GOTO140
340 GOSUB560:INPUT "HOW MANY WORDS";WC
350 GOSUB520
360 CI=WC*5.8/CI 'NUMBER OF COLUMN INCHES
370 PRINT WC;"WORDS OF";PS;"PT LEADED ";LD;"PTS WILL"
380 PRINT"FILL ";
390 PRINT USING "###.#";CI*F1;:PRINT" COLUMN ";U1$;" APPROXIMATELY"
400 PRINT"OR":L=INT(CI*LI):K=CI/DP:C=INT(K):B=C/CP:P=INT(B)
410 PRINT"APPROX";P;" FULL PAGES";C-P*CP;" FULL COLUMNS";INT(L-C*(LI*DP));" LINE
S"
420 PRINT:PRINT"HIT 'ENTER' TO CONTINUE"
430 INPUT Q$:GOTO140
440 GOSUB560:PRINT"how many COLUMN ";U1$;" DO YOU NEED TO FILL";
450 INPUT Q1:Q1=Q1/F1
460 GOSUB520
470 WC=INT(Q1*C1/5.8)
480 PRINT"you will need APPROX";WC;"WORDS TO FILL THIS SPACE USING"
490 PRINT PS;"PT ";FT$;" ";TS$;" LEADED ";LD;"PTS"
500 INPUT "HIT 'ENTER' TO CONTINUE";Q$
510 GOTO140
520 CL=(C6*6/PS)*(CW/6) 'CHARACTERS PER LINE
530 LI=72.27/(PS+LD) 'LINES PER INCH
540 C1=CL*LI 'CHARACTERS PER INCH
550 RETURN
560 CLS:PRINT STRING$(63,"-"):PRINT "PAGE FORMAT IS:"
570 PRINT:PRINT"WIDTH OF COLUMN=";CW*F2;U2$
580 PRINT"NUMBER OF COLUMNS PER PAGE=";CP
590 PRINT"DEPTH OF TEXT AREA ON PAGE=";DP*F1;U1$
600 PRINT:PRINT"TYPE FONT CHOSEN=";:IF FT$=""THENPRINT"NONE CHOSEN YET" ELSE
PRINT FT$;" ";TS$
610 PRINTSTRING$(63,"-"):RETURN
620 DATA AVANT GARDE,5,EXTRA LIGHT,23,BOOK,23,MEDIUM,20.5,DEMI BOLD,20.5,BOLD,20
.5
630 DATA BASKERVILLE,3,ROMAN,24.5,ITALIC,28,SEMI BOLD,22
640 DATA BODONI,5,BOOK,24,BOOK ITALIC,24,MEDIUM,26,MEDIUM ITALIC,28,BOLD,26
650 DATA CALEDONIA,2,ROMAN,25,ITALIC,25
660 DATA CENTURY,6,SCHOOLBOOK,23,SCHOOLBOOK ITALIC,23,SCHOOLBOOK BOLD,20.5,OLD S
TYLE,23,OLD STYLE ITALIC,24,OLD STYLE BOLD,20.5
670 DATA CLARENDON,2,LIGHT,18,BOLD,16
680 DATA GARAMOND,3,LIGHT,28,LIGHT ITALIC,32,BOLD,25
690 DATA GILL SANS,6,LIGHT,23.5,LIGHT ITALIC,25.5,BOLD,21,BOLD CONDENSED,24,BOLD
ITALIC,23,EXTRA BOLD,20
700 DATA GOUDY,3,OLD STYLE,25,OLD STYLE ITALIC,27,BOLD,24
710 DATA GROTESQUE,5,MEDIUM,27,MEDIUM ITALIC,27,BOLD,23,HEAVY,24,HEAVY ITALIC,23
720 DATA HELVETICA,16,LIGHT,24,LIGHT ITALIC,24,MEDIUM,24,MEDIUM ITALIC,24,BOLD,2
2,BOLD ITALIC,22,BLACK,19.5,BLACK ITALIC,19.5,LIGHT CONDENSED,29,LIGHT CONDENSED
ITALIC,27,MEDIUM CONDENSED,26,MEDIUM CONDENSED ITALIC,26
730 DATA BOLD CONDENSED,24.5,BOLD CONDENSED ITALIC,24.5,BLACK CONDENSED,26,BLACK
CONDENSED ITALIC,26
740 DATA KORINNA,3,ROMAN,22,BOLD,22,EXTRA BOLD,20
750 DATA MELIOR,3,ROMAN,23.5,ITALIC,23.5,SEMI BOLD,23.5
760 DATA MEMPHIS,6,LIGHT,21,LIGHT ITALIC,23,MEDIUM,21,MEDIUM ITALIC,23,BOLD,24,E
XTRA BOLD,17
770 DATA OPTIMA,3,ROMAN,24,ITALIC,24,BOLD,24
780 DATA PALATINO,3,ROMAN,23,ITALIC,26,BOLD,22
790 DATA PLANTIN,3,ROMAN,24,ITALIC,26,BOLD,24
800 DATA PERPETUA,3,LIGHT,23,LIGHT ITALIC,26,BOLD,20
810 DATA ROCKWELL,3,LIGHT,22,MEDIUM,22,BOLD,20.5
820 DATA SOUVENIR,6,LIGHT,24.5,LIGHT ITALIC,24.5,MEDIUM,24,MEDIUM ITALIC,24,DEMI
BOLD,22,DEMI BOLD ITALIC,22
830 DATA SPARTAN,3,BOOK,24,BOOK ITALIC,26,HEAVY,24
840 DATA TIFFANY,4,LIGHT,24,MEDIUM,22,DEMI BOLD,22,HEAVY,18
850 DATA TIMES,4,ROMAN,26,ITALIC,26,SEMI BOLD,24.5,SEMI BOLD ITALIC,24.5
860 DATA TRADE GOTHIC,6,17,36,18,31,19,36.5,20,28,EXTENDED,19.5,BOLD EXTENDED,19
.5
870 DATA UNIVERS,9,LIGHT,25,LIGHT ITALIC,25,BOLD,22,MEDIUM,23,MEDIUM ITALIC,23,E
XTRA BOLD,20,LIGHT CONDENSED,33,MEDIUM CONDENSED,29,BOLD CONDENSED,27
880 DATA END,0

```

## \*\*\*\*\* NEWS FROM MICRO-80 PRODUCTS \*\*\*\*\*

The demand for the LNW Research board has exceeded our expectations and, we suspect, those of LNW Research themselves who recently went into backorder for about 5 weeks. We have now caught up and have boards in stock. We have also started delivering fully built-up and tested interfaces based on this board; and complete kits. The prices are shown below:

Completely built-up and tested LNW Research interface, ..... \$415.00  
 mounted in cabinet with power supply and cable to computer.

with Data Separator fitted ..... \$440.00

Dual cassette interface .....ADD \$15.00

Kit consisting of LNW Research expansion interface board ..... \$335.00  
 and manual, ALL components including cabinet and power supply

Data separator for the above ..... \$22.00

Dual cassette interface kit .....ADD \$12.00

The Data separator is recommended if you are using high track-count disk drives or are saving and loading considerable quantities of valuable data to disk. This is a different data separator from the Percom unit recommended for the Tandy interface and requires some soldering to instal.

The prices shown above are for a complete interface including:

- Floppy disk controller
- up to 32K of memory expansion
- Parallel printer port
- Real-time clock
- Serial RS232C/20ma I/O port
- onboard power supplies.
- attractive cabinet
- power transformers mounted inside cabinet

The LNW Research Interface is over \$150 cheaper than the equivalent Tandy interface. System 80 owners may use the LNW Research Interface in conjunction with the Syspand 80 (\$119), which combination is over \$100 cheaper than the System 80 expansion interface. MICRO-80's \$30 memory expansion is, of course entirely suitable for this interface, so you could have a 32K TRS-80 interface for only \$475! This design is over 18 month's old and has been thoroughly tested in the field both in Australia and the USA.

For those who wish to build only part of the interface, we will shortly be announcing the prices of modular kits.

## \*\*\*\*\* LOWER CASE MOD KITS \*\*\*\*\*

We announced some months ago, that we were working on lower-case mod kits for both the TRS-80 and the System 80, which would give lower-case with full descenders. We finished the prototypes some time ago but have been held back from production by component shortages. These have now been overcome and deliveries will have already commenced by the time you read this.

The price for both the TRS-80 kit and the System 80 kit is \$49.00 plus \$2.00 p&p. Kits come complete with comprehensive instructions for fitting and a special machine language driver program to enable lower-case in BASIC. The modification is similar to the Tandy mod. and does not work with Electric Pencil without further modifications.

These kits require disassembly of your computer and some soldering. They should only be installed by someone who has experience in soldering integrated circuits, using a low power, properly earthed soldering iron. If you do not have the necessary experience/equipment, we will instal the modification for you for \$20 plus freight in both directions. Make sure you arrange the installation with us first, before despatching your computer, so that we can assure you of a rapid turn around. We would also like to hear from experienced reders who could undertake this work in their own cities, thus avoiding freight costs.

News from MICRO-80 Products, continued:

\*\*\*\*\* DOUBLE DENSITY FOR DISK DRIVES \*\*\*\*\*

We now have Percom Doublers in stock. The Doubler consists of a double-density disk controller board which plugs into the disk controller socket inside the Tandy expansion interface, and DBLDOS, a double density disk operating system. The Doubler can be fitted in a few minutes without soldering iron and gives you:

- Double capacity on each diskette
- Double capacity on each disk drive
- twice the speed of data transfer between computer and disk compared with the standard system.

\*\*\*\*\* PRICE ..... only \$220.00 plus \$2.00 p\$p \*\*\*\*\*

Installing a Doubler is like buying another set of disk drives, only much cheaper. The Doubler works with most modern disk drives including :- MPI, Micropolis, Pertec, TEAC (as supplied by Tandy). The Doubler is suitable for installation in the TRS-80 expansion interface, the System 80 expansion interface and the LNW Research expansion interface.

If you are using NEWDOS 80, then you also need DOUBLE-ZAP II, on diskette. This program upgrades you NEWDOS 80 to double density. It retains all the familiar features, including the ability to mix and match track counts on the same cable. In addition, it gives NEWDOS 80 the ability to mix densities on the same cable, automatically. If you place a single-density diskette in drive 0 say and a double-density diskette in drive 1, Double-Zap will recognise this and read/write to drive 0 in single-density whilst at the same time, it reads/writes to drive 1 in double-density!

\*\*\*\*\* PRICE ..... only \$53.00 plus \$1.00 p&p \*\*\*\*\*

- 000000 -

\*\*\*\*\* AGEING RELATIVELY LII/4K (C) FRANK KOSZTELNIK \*\*\*\*\*

This program deals with the familiar problem of how much a space traveller ages in comparison with an earthbound companion. The program directly uses the formula:

$$T' = \sqrt{1 - \frac{v^2 \times T}{c^2}}$$

Where T' = time relative to the traveller.  
 T = time relative to Earth companion.  
 V = velocity of traveller.  
 C = speed of light.

EXAMPLE: For a traveller's velocity of 60% the speed of light and a period of 20 years on Earth, the traveller only ages 16 years during the 20 year Earth period.

```

10 REM AGEING RELATIVELY
20 REM FRANK KOSZTELNIK 11 CARRINGTON AVE, KATOOMBA 2780
30 C=2.99793E+8: CC=C*C
40 CLS: INPUT"ENTER VELOCITY OF TRAVELLER (AS PERCENTAGE OF SPEED OF LIGHT)
;CV: CV=CV/100*C: CS=CV*CV
50 INPUT"ENTER TIME INTERVAL (IN YEARS RELATIVE TO EARTH BOUND COMPANION) ";Y
60 T=SQR(1-(CS/CC))*Y
70 PRINT:PRINT:PRINT"TRAVELLER HAS AGED";T;"YEARS"
80 PRINT"COMPARED TO";Y;"YEARS OF EARTH COMPANION"
90 PRINT:PRINT:INPUT"ENTER WHEN READY";A: GOTO40

```

**DON'T BE HELD BACK BY AN  
ANTIQUATED DISK OPERATING SYSTEM  
MOVE UP TO**

**NEWDOS 80** **\$149 incl. p&p**

NEWDOS 80 is a completely new DOS for the TRS-80 SYSTEM 80. It is well-documented, bug free and increases the power of your system many times over. It is upward compatible with TRSDOS AND NEWDOS (ie TRSDOS and NEWDOS+ programs will run on NEWDOS 80 but the reverse is not necessarily so).

These are just a few of the many new features offered by NEWDOS 80.

- \* New BASIC commands that support variable record lengths up to 4095 bytes long.
- \* Mix or match disk drives. Supports any track count from 18 to 96. Use 35, 40, 77 or 80 track 5¼ inch mini disk drives, 8 inch disk drives OR ANY COMBINATION.
- \* An optional security boot-up for BASIC or machine code application programs. User never sees "DOS-READY" or "READY" and is unable to "BREAK", clear screen or issue any direct BASIC statements, including "LIST".
- \* New editing commands that allow program lines to be deleted from one location and moved to another or to allow the duplication of a program line with the deletion of the original.
- \* Enhanced and improved RENUMBER that allows relocation of subroutines.
- \* Create powerful chain command files which will control the operation of your system.
- \* Device handling for routing to display and printer simultaneously.
- \* MINIDOS — striking the D, F and G keys simultaneously calls up a MINIDOS which allows you to perform many of the DOS commands without disturbing the resident program.
- \* Includes Superzap 3.0 which enables you to display/print/modify any byte in memory or on disk.
- \* Also includes the following utilities:
  - Disk Editor/Assembler
  - Disassembler (Z80 machine code)
  - LM offset — allows transfers of any system tape to Disk file — automatically relocated.
  - LEVEL 1 — Lets you convert your computer back to Level 1.
  - LVIDKSL — Saves and loads Level 1 programs to disk.
  - DIRCHECK — Tests disk directories for errors and lists them.
  - ASPOOL — An automatic spooler which routes a disk file to the printer whilst the computer continues to operate on other programs.
  - LCDVR — a lower case drives which display lower case on the screen if you have fitted a simple lower case modification.

**DISK DRIVE USERS  
ELIMINATE CRC ERRORS  
AND  
TRACK LOCKED OUT MESSAGES  
FIT A PERCOM DATA SEPARATOR  
\$37.00 plus \$1.20 p&p.**

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

**MPI DISK DRIVES  
HIGHER PERFORMANCE — LOWER PRICE**

MPI is the second largest manufacturer of disk drives in the world. MPI drives use the same form of head control as 8" drives and consequently, they have the fastest track-to-track access time available — 5msec! All MPI drives are capable of single or double-density operation. Double-density operation requires the installation of a PERCOM doubler board in the expansion interface.

As well as single head drives, MPI also makes dual-head drives. A dual-head drive is almost as versatile as two single-head drives but is much cheaper.

Our MPI drives are supplied bare or in a metal cabinet — set up to operate with your TRS-80 or SYSTEM 80. All drives are sold with a 90 day warranty and service is available through MICRO-80 PRODUCTS.

**MPI B51 40 Track Single Head Drive. . . . . only \$339**  
**MPI B52 40 Track Double Head Drive. . . . . only \$449**

Prices are for bare drives and include p&p. Add \$10.00 per drive for a cabinet and \$60.00 for a power supply to suit two drives. 40 track drives are entirely compatible with 35 track drives. A 40 track DOS such as NEWDOS 80 is necessary to utilise the extra 5 tracks.

**OVER 800 KILOBYTES ON ONE DISKETTE!  
WITH MPI 80 TRACK DRIVES**

MPI 80 track drives are now available. The B91 80 track single-head drive stores 204 Kilobytes of formatted data on one side of a 5¼ inch diskette in single-density mode. In double-density mode it stores 408 Kilobytes and loads/saves data twice as quickly.

The B92 80 track dual-head drive stores 204 Kilobytes of formatted data on EACH side of a 5¼ inch diskette in single-density mode. That's 408 Kilobytes per diskette. In double-density mode, the B92 stores a mammoth 408 Kilobytes per side or 816 Kilobytes of formatted data per diskette. With two B92's and a PERCOM double, you could have over 1.6 Megabytes of on line storage for your TRS-80 for less than \$1500!!

**MPI B91 80 Track Single Head Drive. . . . . only \$499**  
**MPI B92 80 Track Dual Head Drive . . . . . only \$599**

Prices are for bare drives and include p&p. Add \$10.00 per drive for a cabinet and \$60.00 for a power supply to suit two drives. Note: 80 track drives will not read diskettes written on a 35 or 40 track drive. If drives with different track counts are to be operated on the same system, NEWDOS 80 must be used.

**CARE FOR YOUR DISK DRIVES?  
THEN USE  
3M's DISK DRIVE HEAD CLEANING DISKETTES  
\$30.20 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's have come to our rescue with SCOTCH BRAND, non-abrasive, head cleaning diskettes which thoroughly clean the head in seconds. The cleaning action is less abrasive than an ordinary diskette and no residue is left behind. Each kit contains:

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

**MICROPOLIS 77 TRACK DISK DRIVES**

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

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

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

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

Same as above but no cable or NEWDOS 80.

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

4 drive connector cable – suitable for any disk drives.

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

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

**PROGRAMS BY MICROSOFT**

**EDITOR ASSEMBLER PLUS (L2/16K)**

**\$37.50 + \$1.20 p&p**

A much improved editor-assembler and debug/monitor for L2/16K TRS-80 or SYSTEM 80. Assembles directly into memory, supports macros and conditional assembly, includes new commands-substitute, move, copy and extend.

**LEVEL III BASIC \$59.95 plus \$1.20 p&p**

Loads on top of Level II BASIC and gives advanced graphics, automatic renumbering, single stroke instructions (shift-key entries) keyboard debounce, suitable for L2/16K and up (Not Disk BASIC)

**ADVENTURE ON DISK \$35.95 plus \$1.20 p&p**

This is the original ADVENTURE game adapted for the TRS-80. The game fills an entire diskette. Endless variety and challenge as you seek to rise to the level of Grand Master. Until you gain skill, there are whole areas of the cave that you cannot enter. (Requires 32K One Disk)

**BASIC COMPILER \$208 plus \$2.00 p&p**

New improved version, the Basic Compiler converts Disk BASIC programs to machine code, automatically. A compiled program runs, on average, 3-10 times faster than the original BASIC program and is much more difficult to pirate.

**GREEN SCREEN SIMULATOR**

**\$19.95 incl. p&p**

The GREEN SCREEN SIMULATOR is made from a deep green perspex, cut to fit your monitor. It improves contrast and is much more restful to the eyes than the normal grey and white image.

All editorial staff of MICRO-80 are now using GREEN SCREEN SIMULATORS on their own monitors.

Please make sure to specify whether you have an old (squarish) or new (rounded) style monitor when ordering. Not available for Dick Smith monitors.

**UPGRADE TO 16K FOR ONLY \$30.00!!**

**MICRO-80's 16K MEMORY EXPANSION KIT HAS BEEN REDUCED IN PRICE EVEN MORE**

Larger volume means we buy better and we pass the savings on to you. These are our proven, prime, branded 200 ns (yes, 200 nanosecond) chips. You will pay much more elsewhere for slow, 350 ns. chips. Ours are guaranteed for 12 months. A pair of DIP shunts is also required to upgrade the CPU memory in the TRS-80 – these cost an additional \$4.00. All kits come complete with full, step-by-step instructions which include labelled photographs. No soldering is required. You do not have to be an experienced electronic technician to instal them.

**USE TANDY PERIPHERALS ON YOUR SYSTEM-80 VIA**

**SYSPAND-80 – \$119 incl. p&p**

The SYSTEM-80 hardware is not compatible with the TRS-80 in two important areas. The printer port is addressed differently and the expansion bus is entirely different. This means that SYSTEM-80 owners are denied the wealth of economical, high performance peripherals which have been developed for the TRS-80. Until now, that is. MICRO-80 has developed the SYSPAND-80 adaptor to overcome this problem. A completely self-contained unit in a small cabinet which matches the colour scheme of your computer, it connects to the 50-way expansion part on the rear of your SYSTEM 80 and generates the FULL Tandy 40 way bus as well as providing a Centronics parallel printer port. SYSPAND-80 enables you to run an Exatron Stringy Floppy from your SYSTEM 80, or an LNW Research expansion interface or a MICROTEK memory expansion module or any other desirable peripherals designed to interface to the TRS-80 expansion port. Make your SYSTEM 80 hardware compatible with the TRS-80 via SYSPAND-80.

**UPGRADE TO A 48K SYSTEM FOR ONLY \$245!! VIA THE MICROTEK MEMORY EXPANSION/PRINTER MODULE**

Need more memory but don't want to pay over \$600 for an expansion interface? Then the MICROTEK MT-32 memory expansion/printer module is for you. Ready to plug in and go, this module provides you with sockets for an extra 32K of ram in 16K blocks plus a printer port. It is housed in an attractive, grey and black metal cabinet of similar size to the Tandy expansion interface so that you can sit your monitor atop it. The MICROTEK unit runs from the same external power pack as the Tandy CPU. The Centronics parallel printer port enables you to run most printers including the Olivetti ET-121 with MICRO-80 interface. Full instructions for connecting to your system and adding memory chips are included.

SYSTEM 80 owners can use the MICROTEK module via the SYSPAND-80 adaptor. A separate external power supply providing 7.5–0–7.5 volt ac at 0.5 amp plus 20v unfiltered dc at 250 ma is required. (not available at present from MICRO-80 PRODUCTS)

|                        |            |              |              |
|------------------------|------------|--------------|--------------|
| <b>MICROTEK MT32-A</b> | <b>0K</b>  | <b>.....</b> | <b>\$149</b> |
| <b>MICROTEK MT32-B</b> | <b>16K</b> | <b>.....</b> | <b>\$179</b> |
| <b>MICROTEK MT32-C</b> | <b>32K</b> | <b>.....</b> | <b>\$209</b> |

## SOFTWARE BY AUSTRALIAN AUTHORS

All our software is suitable for either the SYSTEM 80 or the TRS-80

### NEW SOFTWARE FROM MICRO-80 PRODUCTS BUSINESS PROGRAMS

#### MICROMANAGEMENT STOCK RECORDING SYSTEM (L2/16K)

Cassette version. . . . . \$29.95 + \$1.00 p&p  
Stringy Floppy version. . . . . \$33.95 + \$1.00 p&p

This system has been in use for 9 months in a number of small retail businesses in Adelaide. It is therefore thoroughly debugged and has been tailor made to suit the requirements of a small business. MICROMANAGEMENT SRC enables you to monitor the current stock level and reorder levels of 500 different stock items per tape or wafer. It includes the following features:—

- Add new items to inventory
- Delete discontinued items from inventory
- List complete file
- Search for any stock number
- Save data to cassette or wafer
- Load data from cassette or wafer
- Adjusts stock levels from sales results and receipt of goods
- List all items requiring reordering

We can thoroughly recommend this program for the small business with a L2/16K computer.

#### SCOTCH BRAND COMPUTING CASSETTES

Super-quality personal computing cassettes.

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

#### UTILITIES

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

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

##### S-KEY features:

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

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

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

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

#### EDUCATIONAL

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

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

#### GAMES

**MICROPOLY (L2/16K)** \$7.50 \$ 50c p&p

Now you can play Monopoly on your micro. The old favourite board game has moved into the electronic era. This computer version displays the board on the screen, obeys all the rules and, best of all, the banker does not make mistakes with your change!

**CONCENTRATION (L2/16K)** \$7.50 + 50c p&p

Another application of supergraphics. There are 28 "cards" displayed on the screen, face down. Players take it in turn to turn them over with the object of finding matching pairs. There are 40 different patterns which are chosen at random, so the game is full of endless variety. This is of particular value in helping young children to learn the art of concentrating and, at the same time, to introduce them to the computer.

**METEOR AND TORPEDO ALLEY (L2/16K)**  
\$9.95 + 50c p&p

Those who frequent games arcades will recognize these two electronic games. In METEOR you must destroy the enemy space ships before they see you. In its most difficult mode, the odds are a thumping 238 to 1 against you being successful. In torpedo alley you must sink the enemy ships without hitting your own supply ship. Both games include sound effects and are remarkably accurate reproductions of the arcade games.

**AUSTRALIAN SOFTWARE (Cont.)****TOUCHTYPE (L2/4K) \$19.95 + 50c. p&p**

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

**GAMES****U BOAT \$7.50 plus 50c p&p**

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

**SPACE INVADERS \$7.50 plus 50c p&p**

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

**GOLF (L2/16K) \$7.50 + 50c p&p**

Pit your skills at mini-golf against the computer. Choose the level of difficulty, the number of holes and whether you want to play straight mini golf or crazy golf. Complete with hazards, water traps, bunkers and trees. Great fun for kids of all ages.

**DOMINOES(L2/16K) \$7.50 + 50c p&p**

Pit your skill at dominoes against the computer, which provides a tireless opponent. Another application of supergraphics from the stable of Charlie Bartlett. Dominoes are shown approximately life size in full detail (except for colour!). The monitor screen is a window which you can move from one end of the string of dominoes to the other. Best of all, you don't lose any pieces between games!

**KID'S STUFF (formerly MMM-1) \$7.50 plus 50c. p&p**

Three games on one cassette from that master of TRS-80 graphics, Charlie Bartlett. Includes INDY 500, an exciting road race that gets faster and faster the longer you play, SUBHUNT in which your warship blows up unfortunate little submarines all over the place, and KNEIVEL (as in motorcycle, ramp and buses).

**OTHER PROGRAMS****INFINITE BASIC BY RACET (32K/1 DISK)****\$49.95 + 50c. p&p**

Full matrix functions — 30 BASIC commands; 50 more STRING functions as BASIC commands.

**GSF/L2/48K \$24.95 + 50c. p&p**

18 machine language routines including RACET sorts.

**BUSINESS ADDRESS AND INFORMATION SYSTEM (48K/DISK) \$24.95 + 50c. p&p**

Allows you to store addresses and information about businesses, edit them and print them out.

**HISPED (L216, 32 or 48K) \$29.95**

This machine language program allows you to SAVE and LOAD programs and data to tape at speeds up to 2000 baud (4 times normal) using a standard cassette recorder. A switch must be installed to remove the XRX III loading board, if fitted.

**PROGRAMS FROM CREATIVE COMPUTING****ADVENTURE PROGRAMS****ADVENTURELAND (L2/16K) \$14.95 + 50c. p&p**

Try to find and take treasures as you explore a fantasy world. The computer acts as your puppet and carries out your two word commands.

Sometimes you will need special objects to do certain things, often a little magic is necessary. Absorbing and challenging.

**THE COUNT ADVENTURE (L2/16K)****\$14.95 + 50c. p&p**

In this adventure, you awaken in a bed in a castle in Transylvania. You don't know why you are there but you'd better solve the puzzle before it's too late. Just as enthralling as ADVENTURELAND but blood thirstier!

**ADVENTURELAND AND PIRATE ADVENTURE ON DISK (32K ONE DISK) \$24.95 + 50c. p&p**

This is the Adventureland program on disk plus Pirate Adventure, complete with buried treasure, keel hauling, planks for walking and skulls and crossbones.

You can save the game to disk at any point and return to it later when your nerves are steady.

**GAMES****AIR TRAFFIC CONTROLLER (L2/16K)****\$9.95 + 50c. p&p**

One of the hottest selling games in the USA, you are the Air Traffic Controller and the monitor is your radar screen. Bring down the aircraft safely and avoid mid-air collisions.

**Z CHESS (L2/16K) \$19.95 + 50c. p&p****(DISK/32K)****\$24.95 + 50c. p&p**

Seven levels of ability, contains all standard moves including castling and En Passant captures. It can play either black or white and its versatile board set-up mode allows specific positions to be played as desired.

**SPACE GAMES (L2/16K) \$13.50 + 50c. p&p**

3 Space Games including ULTRA-TREK, ROMULAN, and STARWARS. Fast, real-time graphics.

**STRATEGY GAMES (L2/16K) \$9.50 + 50c. p&p**

5 Strategy games including TUNNEL VISION (find your way out of a 3-D maze), EVASION — avoid the deadly snake), JIGSAW (put the puzzle together), THE MASTERS (Golf on the '80 for up to 4 players), MOTOR RACING (Compete against the computer at Indy or the Grand Prix).

**GRAPHING PACKAGE (L2/16K) \$9.95 + 50c. p&p**

A set of 6 utility programs which allow you to draw BAR GRAPHS, GRAPH CARTESIAN COORDINATES, carry out POLAR GRAPHING, PARAMETRIC GRAPHING, LINEAR REGRESSION and PARABOLIC REGRESSION.

**BOOKS****LEVEL II ROM REFERENCE MANUAL****\$24.95 + \$1.20 p&p**

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

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

The hottest selling TRS-80 book in the U.S.A. Disk file structures revealed, DOS's compared and explained, how to recover lost files, how to rebuild crashed directories — this is a must for the serious Disk user and is a perfect companion to any of the NEWDOS's.

\*\*\*\*\* FINANCE LI/4K by M. Svensdotter \*\*\*\*\*

The best way to demonstrate this program is to give you some data and show you what happens once you enter it.

Let's say that your income for this week was \$185.25, you already had \$30.32 in your pocket and you had various expenses during the week of \$37.25 and \$142.30 leaving you with \$36.02.

One thing to remember is, if you are asked to enter 1 for a particular thing to happen, entering any other number will stop it happening. First of all, you are asked to enter 1 if this is a new year. For this demonstration enter 1 then you are asked to enter the financial year (1981), when the year is repeated on the screen, press ENTER. Now comes the time for entering your income and expenses. Money paid to you should be entered as a negative number and your expenses and money in hand as a positive number. The above amounts must be entered in this form :-

```
-18525 ENTER
 3725 ENTER
14230 ENTER
 3602 ENTER (3032+570)
 0 ENTER (this tells your 80 that you have finished).
```

You are then asked to enter 1 if you want a data check. This is so you can have a look at all values in memory.

After this you will see

#### INCOME & EXPENSES BALANCE

at the top of the screen and the last entry displayed at the centre of the screen (\$36.02). You won't want to place this amount on your data tape so when ENTER 1 TO DELETE DATA appears, all you do is enter 1 and that amount will be erased from memory.

The next thing to do is enter a number other than 1, you will then be told to press ENTER when the tape is ready (place a fresh tape in the recorder and press play and record). As each amount is placed on the tape it will be displayed on the screen. You will be asked to enter 1 if you require a graph. ENTER 1 and you will be asked to specify the low limit. If the low limit is \$20, then any column with an amount less than this will be empty except for an L at the bottom. An I at the bottom of a column means that this amount was entered as income. Once the graph has been drawn you will be asked if you want to magnify the graph. If you don't, then you have finished. If you do, enter .5 for half size, whilst whole numbers will increase magnification.

```
10 C.:GOS.500:P.:P."FOR ";A(0)
15 B$=PRESS ENTER
20 P.B$;:I.A$:CLS
30 P."ENTER WHOLE NUMBERS ONLY, FOR EXAMPLE $ 5.40 IS 540
40 I."-FOR INCOME + FOR EXPENSES";A(B):C.
50 IFA(B)=0G.B0
60 B=B+1
70 G.40
80 F.Z=B+1TOB+10:A(Z)=0:N.Z
100 C=0:B=1:C.:I."ENTER 1 FOR DATA CHECK";Q:C.:IFQ<>1G.115
110 P.AT46B,"$";A(B)/100;AT44B,B$;:I.A$:G.125
115 C.
120 P.AT0,"$";A(B)/100,
125 C=C+A(B)
130 IFA(B+1)=0C.:P.AT2B.INT(C)/100:G.160
140 B=B+1:IFQ<>1G.120
150 C.:G.110
160 IFC<0P.AT0,"INCOME EXCEEDS EXPENSES BY $";:G.200
170 IFC>0P.AT1,"EXPENSES EXCEED INCOME BY $";:G.200
180 C.:P."INCOME & EXPENSES BALANCE
200 P.AT44B,"$";A(B)/100,:I."1 WILL DELETE THE DATA";Q
205 P.AT44B:P.:P.:P.
210 IFQ=1A(B)=0:B=B-1:G.200
215 IFC<>0B=B+1:G.15
220 B=1
230 P.B$;:I." WHEN TAPE IS READY TO RECIEVE DATA";A$
240 C.:R=0
250 D=A(B):E=A(B+1):F=A(B+2):G=A(B+3):H=A(B+4):I=A(B+5)
255 J=A(B+6):K=A(B+7):L=A(B+8)
260 P.#D;",";E;",";F;",";G;",";H;",";I;",";J;",";K;",";L
270 F.Q=0TOB:R=R+10:IFR>55R=0:P.
```

```

280 IFA(B+Q)=0P.:G.310
290 P.T.(R);A(B+Q)/100;:N.Q
300 B=B+9:G.250
310 P.#A(0)
320 G.700
500 B=1:I."ENTER 1 IF THIS IS A NEW YEAR";Q:IFQ=1G.610
510 C.:R=0:P.AT1,"DATA TAPE IN PLAYER & PRESS PLAY";AT0;
520 I.#D,E,F,G,H,I,J,K,L
530 A(B)=D:A(B+1)=E:A(B+2)=F:A(B+3)=G:A(B+4)=H:A(B+5)=I
535 A(B+6)=J:A(B+7)=K:A(B+8)=L
540 F.Q=0T08:R=R+10:IFR>55R=0:P.
550 IFA(B+Q)=0G.580
560 P.T.(R);A(B+Q)/100;:N.Q
570 B=B+9:G.520
580 Z=B-5:IFZ<0Z=0
590 Z=Z+1:IFA(Z)<>0G.590
600 B=Z:I.#A(0):RET.
610 I."ENTER FINANCIAL YEAR";A(0):RET.
700 I."ENTER 1 IF YOU REQUIRE A GRAPH";R
710 V=1:IFR<>1ST.
720 I."WHAT IS THE LOW LIMIT (IN DOLLARS)";S:S=INT(S*100)
730 C.:T=0:C=0
740 F.B=1T0Z:IFA(B)=0G.830
750 C=C+A(B):U=47-(ABS(C*U/2173))
765 IFU<0U=0
770 T=T+2:IFC<-SF.Y=46TOUS.-1:S.(T,Y):S.(T+1,Y):N.Y:G.780
775 P.AT960+T/2,"L";AT0,
780 IFABS(A(B))<>A(B)G0S.900
790 IFT<124G.820
800 P.AT0,B$;:I.A$
810 C.:T=0
820 N.B
830 I."ENTER 1 TO CHANGE MAGNIFICATION";U
840 IFU<>1ST.
850 I."WHAT MAGNIFICATION DO YOU REQUIRE";U
860 G.720
900 P.AT960+T/2,"I";AT0,
910 F.W=44TOU-2S.-2
920 IFW-1>0R.(T,W):R.(T+1,W-1)
930 N.W:RET.
960 P.:RET.
999 REM-INITIAL EXPLANATION
1000 U=0
1010 CLS:G0S.950:U=U+1
1020 P."ARITHMETIC REVISION : 4 MULTIPLIED BY 3 = 12"
1030 P.T.(22)," - 4 MULTIPLIED BY 3 = - 12"
1040 P.T.(22)," 4 MULTIPLIED BY - 3 = - 12"
1050 P.T.(22)," - 4 MULTIPLIED BY - 3 = 12"
1055 F.W=1T064:P."-";:N.W
1060 A=4:B=3:C=5:D=2:E=4:F=6
1070 IFU<>1T.1090
1080 G.1200
1090 IFU<>2T.1110
1100 F=-6:G.1200
1110 IFU<>3T.1130
1120 E=-4:G.1200
1130 IFU<>4T.1150
1140 D=-2:F=-6:G.1200
1150 IFU<>5T.1200
1160 D=-2:E=-4:F=-6
1200 G0S.605:G0S.900
1210 IFU=5CLS:P."TRY SOME PROBLEMS -":P.:RET.
1220 G.1010

```

\*\*\*\*\* REMOVING LINEAR BRACKETS LI/4K by - A. W. Holland \*\*\*\*\*

This program is intended to assist students of algebra in learning how to remove linear brackets. Initially, a step-by-step explanation is given for a number of examples, then the student is posed one problem at a time and asked for an answer. If the problem is answered correctly, the next problem is presented. If an incorrect answer is given, the student is requested to try again. On giving another incorrect answer, the correct solution is displayed together with its step-by-step derivation and the student is advised to seek human help if he does not understand the process.

```

5 CLS:IN."PLEASE ENTER A NUMBER BETWEEN 1 & 100";A
5 F.J=1T0A:Z=RND(30000):N.J
10 CLS:Y=1:N=0:G0S.950
20 P."WOULD YOU LIKE AN EXPLANATION"
30 IN."BEFORE TRYING SOME PROBLEMS";G

```

```

40 IFG=1GOS.1000:G.100
50 IFG<>0P."INPUT ERROR":G.20
100 CLS:L=0:J=0
110 A=RND(19)-10:B=RND(19)-10:C=RND(19)-10
120 IF(A=0)+(B=0)+(C=0)T.110
121 IF(ABS(A)=1)+(ABS(B)=1)+(ABS(C)=1)T.110
130 D=RND(19)-10:E=RND(19)-10:F=RND(19)-10
140 IF(D=0)+(E=0)+(F=0)T.130
141 IF(ABS(D)=1)+(ABS(E)=1)+(ABS(F)=1)T.130
150 L=L+1:GOS.700
180 X=A*B+D*E:V=A*C+D*F
190 P.:P."THE ANSWER IS IN THE FORM A X + B"
200 J=J+1:IN."WHAT IS THE VALUE OF A";T
210 IN."WHAT IS THE VALUE OF B";Z
220 P."YOUR ANSWER IS ";
230 H=T:I=Z:GOS.800
260 P.:IF(T=X)*(Z=V)P."CORRECT - WELL DONE!":F.W=1T03000:N.W:G.380
290 IFJ<>2P."HAVE ANOTHER GO":G.200
300 P."SORRY.THE CORRECT ANSWER IS ";
310 H=X:I=V:GOS.800
340 P."WOULD YOU LIKE ME TO SHOW YOU HOW TO DO IT";
350 IN.W:IFW=1GOS.600:GOS.900:G.380
360 IFW<>0P."INPUT ERROR":G.340
380 CLS:IFL<>5J=0:G.110
390 IN."DO YOU WANT ANOTHER 5 PROBLEMS";W:IFW=1T.100
400 IFW<>0P."INPUT ERROR":G.390
410 CLS:P.A.470,"OKAY SEE YOU LATER"
420 F.W=1T02000:N.W:CLS
430 G.430
539 REM-PROBLEM EXPLANATION
600 CLS
605 GOS.700
608 P."=" ";
609 IF(A*B)<0P."-";
610 P.ABS(A*B);"X ";;IFA*C>0P."+":G.620
615 P."-";
620 P.ABS(A*C);:IFD*E>0P."+":G.630
625 P."-";
630 P.ABS(D*E);"X ";;IFD*F>0P."+":G.640
635 P."-";
640 P.ABS(D*F)
645 P."=" ";
647 IF(A*B+D*E)=0T.650
648 IF(A*B+D*E)<0P."-";
649 P.ABS(A*B+D*E);"X ";
650 IF((A*B+D*E)<>0)*((A*C+D*F)>0)P."+":G.660
655 IF(A*C+D*F)<0P."-";
660 P.ABS(A*C+D*F):P.
665 F.W=1T064:P."-";:N.W
670 P."IF YOU CAN'T UNDERSTAND THE ABOVE"
680 P."PLEASE ASK A HUMAN TO EXPLAIN."
690 P.:RET.
699 REM-WRITE PROBLEM
700 IFA<0P."-";:
702 P.ABS(A);"(";;IFB<0P."-";
703 P.ABS(B);"X ";;IFC>0P."+":G.710
705 P."-";
710 P.ABS(C);") ";;IFD>0P."+":G.720
715 P."-";
720 P.ABS(D);"(";;IFE<0P."-";
722 P.ABS(E);"X ";;IFF>0P."+":G.730
725 P."-";
730 P.ABS(F);")":RET.
739 REM.WRITE ANSWER
800 IFH=0T.820
805 IFH<0P."-";:H=-H
810 P.H;"X ";;IFI=0P." ":RET.
820 IF(H<>0)*(I>0)P."+":G.830
825 IFI<0P."-";
830 P.ABS(I):RET.
899 REM-READY?
900 IN."PRESS 'ENTER' WHEN READY TO GO ON";A$
910 RET.
949 REM.TITLE
950 CLS:P.T.(11), "*** LINEAR ALGEBRA - REMOVING BRACKETS ***"

```

\*\*\*\*\* TABLES TESTER LII/4K

by - Peter &amp; Shirley Miller \*\*\*\*\*

Having just purchased a TRS 80, we were eager to justify the expense by producing a useful program for our children. This period coincided with them learning their multiplication tables and instead of testing them manually, it seemed an ideal task for the 80. The main problem is retaining their attention and interest and the bulk of the program is devoted to this end. An excellent program by Spencer George (MICRO 80 issue 8) provided the interest with his graphics, not to mention his program layout.

The 'Tables Tester' is very simple with its base being two random numbers between 1 and 12, (lines 400 & 410) multiplied together and printed. The 80 tests the reply for correct, too small, or too large, and reacts accordingly. The main loop (W) controls the number of questions asked (12) whilst 'T' accumulates the number of correct answers. It should be noted that a correct answer must be given on the first attempt to increment the score. If by the second attempt the correct answer is not given, the 80 will provide it.

The only opportunity to exit this program is when the student has completed a set correctly, otherwise it loops back for another 12 questions. It may look a lot of typing but AUTO will help with the input and to reassure you - it works and the kids enjoy it.

```

10 *****
 * TABLES TESTER TRS 80 LEV2*

20
30 BY P & S MILLER
 GRAPHICS FROM SPENCER GEORGE
 SYMBOL/BAS-MICRO 80 ISSUE 8
40 PLEASE DELETE ALL REM STATEMENTS
50 ** DATA FOR ANDROID
60 DATA 3, 168, 5, 148, 66, 160, 67, 158, 68, 188, 69, 173, 130, 178, 131, 187,
 132, 188, 133, 183
70 DATA 193, 174, 194, 145, 195, 191, 196, 198, 197, 191, 198, 162, 199, 157
80 DATA 258, 131, 259, 171, 261, 151, 262, 131, 322, 176, 323, 186, 325, 181, 32
 6, 176, 70, 144, 134, 177
90 DIM A1(27), B1(27)
100 FOR J=1 TO 27
110 READ A1(J), B1(J)
120 NEXT J
130 ** OPENING MESSAGE
140 CLS
150 INPUT"HELLO - WHAT IS YOUR NAME THEN ?";N$
160 CLS
170 PRINT CHR$(23)
180 PRINT N$;" IS A LOVELY NAME "
190 PRINT
200 GOSUB 910
210 PRINT" WELL ";N$;" I'M THE TRS 80"
220 PRINT"**** TABLES TESTER !!! ****"
230 GOSUB 910 ' ** SCROLL DELAY SUBROUTINE
240 PRINT"---- AND I WILL THINK OF SOME TO ASK YOU"
250 GOSUB 910
260 PRINT
270 PRINT"--I KNOW ALL MY TABLES--"
280 GOSUB 910
290 PRINT
300 PRINT" NOW LETS SEE IF YOU KNOW YOURS!"
310 GOSUB 910
320 PRINT
330 PRINT"RIGHT THEN ";N$
340 PRINT" THIS IS HOW IT GOES - "
350 GOSUB 940

```

```
360 GOSUB 910
370 RANDOM '** THESE NEXT 4 LINES ARE WHAT ITS ALL ABOUT
380 CLS
390 PRINT CHR$(23)
400 A=RND(12)
410 B=RND(12)
420 W=W+1 '** NUMBER OF QUESTIONS ASKED (12 TIMES THROUGH)
430 IF W=13 THEN 670
440 PRINT @ 510, A;" TIMES ";B;" = ";
450 INPUT X
460 Y=A*B' ** THE TRS 80 FINDS THE CORRECT ANSWER
470 IF X=Y THEN 500 '** CORRECT ANSWER
480 IF X<Y THEN 530 '** ANSWER TOO SMALL
490 IF X>Y THEN 580 '** ANSWER TOO LARGE
500 GOSUB 1070
510 T=T+1
520 GOTO 380

530 PRINT"TOO SMALL ! TRY ONCE MORE";
540 INPUT X
550 IF X<>Y THEN 630
560 GOSUB 970
570 GOTO 380
580 PRINT"TOO BIG ! TRY ONCE MORE";
590 INPUT X
600 IF X<>Y THEN 630
610 GOSUB 970
620 GOTO 380
630 PRINT
640 PRINT"SORRY-BUT ";A;" TIMES ";B;" = ";Y
650 GOSUB 940
660 GOTO 380
670 CLS
680 PRINT CHR$(23)
690 FORQ=1 TO 300
700 U=RND(1000):PRINT@U,"*";
710 NEXTQ
720 GOSUB910
730 CLS
740 PRINTCHR$(23)
750 PRINT @ 454, "YOUR SCORE IS ";T
760 PRINT TAB(8)"OUT OF 12"
770 IF T<12 THEN 830
780 PRINT"WONDERFUL - ALL CORRECT. "
790 PRINT" WOULD YOU LIKE TO PLAY AGAIN",
800 INPUT H$
810 IF LEFT$(H$,1)="Y" THEN 880
820 END
830 PRINT"NOT YOUR BEST "
840 PRINT"---I THINK WE WILL PLAY AGAIN"
850 GOSUB 940
860 CLS
870 PRINT CHR$(28)
880 T=0
890 W=0
900 GOTO 380
910 FOR K=1 TO 980
920 NEXT K
930 RETURN
940 FOR E=1 TO 2000
950 NEXT E
960 RETURN
970 V=15360' ** RANDOM STARS ROUTINE
980 FOR SS=1 TO 25
990 D=RND(500)
1000 POKE V+2*D, 42
1010 FOR JH=1 TO 25
1020 NEXT JH
1030 NEXT SS
1040 FOR L=1 TO 400
1050 NEXT L
1060 RETURN
1070 CLS' ** ANDROID PRINTING ROUTINE
1080 FOR SS=1 TO 27
1090 POKE 15880 +A1(SS), B1(SS)
1100 NEXT SS
```

```

1110 FOR TY =1 TO 3' ** HEAD POSITION 1
1120 POKE 16012, 179
1130 POKE 15883, 128
1140 POKE 15885, 128
1150 POKE 15947, 174
1160 POKE 15949, 157
1170 FOR S=1 TO 100
1180 NEXT
1190 POKE 15883, 168' ** HEAD POSITION 2
1200 POKE 15885, 148
1210 POKE 16012, 188
1220 POKE 15947, 142
1230 POKE 15949, 141
1240 FOR S=1 TO 100
1250 NEXT S
1260 NEXT
1270 FOR EE=1 TO 100
1280 NEXT EE
1290 RETURN
1295 '
1300 ' ** LIST OF VARIABLES
1305 '
1310 ' A & B : TABLES VALUE
1320 ' E : EXTRA DELAY
1330 ' K : SCROLL DELAY
1340 ' N$: PLAYERS NAME
1344 ' Q : LOOP COUNTER FOR STARS
1345 ' U : RANDOM SCREEN POSITION
1350 ' T : NUMBER CORRECT
1360 ' W : NUMBER OF QUESTIONS ASKED
1370 ' X : PLAYERS ANSWER
1380 ' Y : TRS 80'S ANSWER TO ABOVE
1390 ' J,U,JH,S,SS AND EE - REFER SYMBOL/BAS
 BY SPENCER GEORGE

```

---

\*\*\*\*\* BOWLING                    LII/4K                    by - Carl Cranstone \*\*\*\*\*

This program has sound for TRS-80's. System 80 users should consult Issue 14, page 7 for guidance on converting the program to give sound from their computers.

When typing in Line 0, type "0 GOT020:" then fill the remainder of the line with blank spaces. This allows room for the machine language sound routine to be POKE'd into memory on RUNNING.

Before you RUN the program, set up the tape recorder for sound effects. Put the large grey plug in MIC and press the PLAY button. Remove the other plugs and turn up the volume. .

HOW TO PLAY:-

You are in a bowling alley and you have ten shots. ie. 5 frames -- 2 balls per frame. Move the graphic man up and down using the keys:- "W" for up and "S" for down. You will hear a beep when you move the man. Press both keys at once to roll the ball when you have the man in position. If you manage to knock all the pins down on the first ball you will see the man "jump with joy" on the screen. A strike is accompanied by sound effects. Some of the pins have a bonus on them so, if you knock the right ones over, you will get extra points. If you gutter the ball you will get zero for the rest of the frames so watch where you put the man.

0 GOT020:

```

10 DATA221,33,36,67,221,78,0,121,183,200,221,70,1,62,1,211,255,16,254,221,70,1,6
2.2.211,255,16,254,13,194,253,66,221,35,221,35,1,255,255,33,48,0,9,219,29,67,195
,247.66
20 POKE16526,243:POKE16527,66:FORN=17139T017187:READY:POKEN,Y:NEXT:POKE17188,100
:POKE17189,100:POKE17190,0:X=USR(0)
30 ' BOWLING:C.CRANSTONE:JAN./81:(C):S.A.LII.4K
40 CLS:PRINT@20,"BOWLING":FORT=1T0500:NEXTT
50 CLS:CLEAR50:FORT=15574T015615:POKET,176:NEXTT:FORT=15895T015935:POKET,131:NEX
TT:FORT=15615T015935STEP64:POKET,191:NEXTT:POKE15615,176:POKE15935,131:MP=15564
60 GOSUB200

```

```

70 POKEMP,130:POKEMP+1,186:POKEMP+2,170:POKEMP+3,146:POKEMP-64,168:POKEMP-63,174
:POKEMP-62,191:POKEMP-61,172:POKEMP+4,131:POKEMP-60,176
80 IFPEEK(15340)=128THENGOSUB140:MP=MP-64:IFMP-64<=15488THENMP=MP+64
90 IFPEEK(15340)=8THENGOSUB140:MP=MP+64:IFMP+64>=16063THENMP=MP-64
100 IFBA=>=10THEN180
110 IFPEEK(15340)=136THENBA=BA+1:GOSUB170:IFZ=2THEN70ELSEGOSUB150
120 GOSUB460
130 POKEMP-6,32:POKEMP-5,32:POKEMP-4,32:POKEMP-3,32:GOTO70
140 FORT=MPTOMP+4:POKET,32:NEXT:FORT=MP-64TOMP-60:POKET,32:NEXTT:POKE17188,25:PO
KE17189,255:POKE17190,0:X=USR(0):RETURN
150 POKEMP+4,32:POKEMP-60,32:S=MP+5:FORK=STOS+(RND(4)+39):POKEK,32:POKEK-1,32:PO
KEK+1,32:POKEK-65,32:POKEK-64,32:POKEK-63,32:FORT=1T020:NEXTT:POKEK,131:POKEK-64
.176:NEXTK:POKEK,32:POKEK-64,32
160 POKEK-63,32:POKEK-1,32:POKEK-65,32:GOTO360
170 IFMP<>15564ANDMP<>15628ANDMP<>15948ANDMP<>15884THENRETURNELSEPRINT@0,"GUTTER
!!!":V=V+1:FORT=1T090:NEXTT:PRINT@0," ":Z=2:GOTO210
180 CLS:PRINT"END OF GAME.":PRINT"YOU SCORED-";A;"-";B;"-";C;"-";D;"-";E;"-";F;"
-";G;"-";H;"-";I;"-";J:PRINT"TOTAL =";A+B+C+D+E+F+G+H+I+J
190 PRINT"ANOTHER GAME?":INPUT$;IFLEFT$(T$,1)="Y"THENRUNELSEEND
200 P1=15794:P2=15733:P3=15797:P4=15672:P5=15800:P6=15864:P7=15675:P8=15739:P9=1
5603:P0=15867:POKEP1,131:POKEP2,140:POKEP3,140:POKEP4,176:POKEP5,131:POKEP6,131:
POKEP7,140:POKEP8,140:POKEP9,140:POKEP0,140:RETURN
210 IFBA=1THENA=0:RETURN
220 IFBA=2THENB=0:RETURN
230 IFBA=3THENC=0:RETURN
240 IFBA=4THEND=0:RETURN
250 IFBA=5THENE=0:RETURN
260 IFBA=6THENF=0:RETURN
270 IFBA=7THENG=0:RETURN
280 IFBA=8THENH=0:RETURN
290 IFBA=9THENI=0:RETURN
300 IFBA=10THENJ=0:RETURN
310 IFPEEK(P1)<>131SC=SC+1:IFPEEK(P2)<>140SC=SC+1:IFPEEK(P3)<>140SC=SC+1:IFPEEK(
P4)<>176SC=SC+1:IFPEEK(P5)<>131SC=SC+1:IFPEEK(P6)<>131SC=SC+1:IFPEEK(P7)<>140SC=
SC+1:IFPEEK(P8)<>140SC=SC+1:IFPEEK(P9)<>140SC=SC+1
320 IFPEEK(P0)<>140SC=SC+1
330 IFBA=1THENA=SCELSEIFBA=2THENB=SCELSEIFBA=3THENC=SCELSEIFBA=4THEND=SC
340 IFBA=5THENE=SCELSEIFBA=6THENF=SCELSEIFBA=7THENG=SCELSEIFBA=8THENH=SCELSEIFBA
=9THENI=SCELSEIFBA=10THENJ=SC
350 SC=0:RETURNELSEIFQ=2THENQ=0GOTO200
360 O=RND(7):ONOGOTO370,380,390,400,410,420,430
370 POKEP1,32:GOTO310
380 POKEP1,32:POKEP2,32:GOTO310
390 POKEP4,32:POKEP7,32:POKEP8,32:GOTO310
400 POKEP3,32:POKEP4,32:POKEP5,32:POKEP0,32:GOTO310
410 POKEP0,32:POKEP2,32:POKEP4,32:POKEP6,32:POKEP8,32:GOTO310
420 POKEP6,32:POKEP5,32:POKEP8,32:POKEP1,32:POKEP9,32:POKEP7,32:GOTO310
430 POKEP0,32:POKEP1,32:POKEP3,32:POKEP4,32:POKEP5,32:POKEP6,32:POKEP7,32:POKEP8
.32:POKEP9,32:IFBA=10RBA=30RBA=50RBA=70RBA=90THEN440ELSEGOTO310
440 POKE17189,84:POKE17189,130:POKE17190,84:POKE17191,195:POKE17192,0:FORT=1T05:
X=USR(0):NEXTT:BA=BA+1
450 PRINT@0,"STRIKE!!!":FORT=1T040:NEXTT:POKEMP,176:POKEMP-6,89:POKEMP-5,65:POKE
MP-4,89:POKEMP-3,33:POKEMP+1,134:POKEMP+2,130:POKEMP-64,138:POKEMP+3,184:POKEMP-
61,142:POKEMP+4,144:FORT=1T0130:NEXTT:PRINT@0," ":Q=2:GOTO310
460 IFBA=0ORBA=20RBA=40RBA=60RBA=80RBA=100GOSUB200:RETURNELSERETURN

```

\*\*\*\*\* VICTORY AT BATHURST L2/16K

\*\*\*\*\*

Overseas readers should note that this program is modelled on a real saloon car race held in October each year at a spectacular and long road circuit at Bathurst, Australia. You will compete against other well-known saloon car drivers on the Australian racing circuit. U.K. readers may wish to substitute Brands Hatch or whatever for Bathurst and may want to use the names of well-known British drivers.

Up to four players may compete in the race and up to four people may bet on the outcome, both before and during the course of the race up to lap 120. Players are required to enter their average speed per lap and decide if they are going to stop at the pits for fuel or tyres. Bathurst is a counter-clockwise circuit so the right front tyre wears the quickest. The game can take a couple of hours to play (just like a real race) so settle back and see if you can outwit Brock or any of the other professionals out there in the next hour or so. Full instructions are provided throughout the game.

```

10 REM:VICTORY AT BATHURST
20 REM:BY JOHN S. RICHARDSON
30 REM:14 ABOYNE RD.,GOOSEBERRY HILL,W.A. 6076.
40 CLS:GOTO70
50 INPUT"PRESS 'NEW LINE' TO CONTINUE";C$
60 RETURN
70 PRINTTAB(10)"VICTORY AT BATHURST 1981"
80 PRINT:PRINTTAB(5)"THE HARDIE-FERODO 1000 SIMULATION GAME"
90 FORX=2TO10
100 READ N$(X)
110 NEXTX
120 DATA CARTER,HARVEY,JANSON,BROCK,JOHNSON,MOFFAT,MORRIS,GRICE,JONES
130 FORX=1TO1500:NEXTX
140 CLS
150 INPUT"HOW MANY PLAYERS (MAXIMUM 4)";NU
160 IFNU<1ORNU>4THENPRINT"INVALID NUMBER - MAX. 4"
170 IFNU<1ORNU>4GOTO150
180 CLS:PRINT"WELCOME TO THE MT. PANORAMA CIRCUIT AT BATHURST"
190 PRINT:PRINT"YOU ARE PRIVELEGED TO HAVE BEEN ACCEPTED TO DRIVE
IN THIS YEAR'S HARDIE-FERODO 1000 RACE"
200 PRINT:PRINT"YOU ARE FORTUNATE IN THAT ONLY"10-NU"OTHER DRIVERS
HAVE ENTERED, ALL DRIVING SIMILAR CARS"
210 PRINT:PRINT"HOWEVER....YOU WILL BE COMPETING AGAINST AUSTRALIA'S
TOP DRIVERS..."
220 FORX=NU+1TO10
230 PRINTN$(X),
240 NEXT X
250 PRINT:GOSUB50
260 CLS:PRINT"THE DRIVERS' BRIEFING IS ABOUT TO BEGIN"
270 PRINT
280 FORX=1TONU
290 PRINT"PLEASE ENTER NAME OF PLAYER NUMBER*X;
300 INPUTN$(X)
310 NEXTX
320 PRINT:PRINT"GOOD LUCK....";
330 FORX=1TONU
340 PRINTN$(X)". ";
350 NEXTX
360 PRINT
370 PRINT".....YOU'RE GOING TO NEED IT"
380 FORX=1TO1500:NEXTX
390 CLS:PRINT@400,"DRIVERS' BRIEFING"
400 FORX=1TO1000:NEXTX
410 CLS:PRINT"THE MT. PANORAMA CIRCUIT IS 6.172 KILOMETRES LONG."
420 PRINT"163 LAPS ARE TO BE COMPLETED."
430 PRINT"EVERY 5 MINUTES, YOU WILL BE REQUIRED TO INDICATE
YOUR PLANNED AVERAGE SPEED FOR THE NEXT 2 LAPS."
440 PRINT
450 PRINT"THE ACCEPTABLE RANGE OF SPEEDS IS:
 158 KM/HR (LAP TIME 2 MINS. 20 SECS.) - FASTEST
 124 KM/HR (LAP TIME 3 MINS. 0 SECS.) - SLOWEST"
460 PRINT
470 PRINT"SOMETIMES YOUR LAP TIMES WILL BE FASTER OR (USUALLY)
SLOWER THAN YOU PLANNED"
480 PRINT:PRINT"E.G....AS YOUR FUEL LOAD DECREASES, YOUR SPEED MAY INCREASE"
490 PRINT" AS YOUR TYRES WEAR, YOUR SPEED MAY DECREASE"
500 GOSUB50
510 CLS:PRINT"YOU MAY ALSO SUFFER MECHANICAL OR ACCIDENT DAMAGE,
ESPECIALLY IF YOU DRIVE FAST."
520 PRINT:PRINT"FUEL: YOU MAY STOP FOR FUEL AT ANY TIME. IF YOUR
FUEL LOAD IS BELOW 5% OF TOTAL, A STOP IS MANDATORY."
530 PRINT"FUEL STOPS TAKE 1 MINUTE."
540 PRINT:PRINT"TYRES: YOU MAY STOP FOR TYRES AT ANY TIME. TYRES WILL BE
CHANGED DURING FUEL STOPS IF MORE THAN 50% WORN."
550 PRINT"TYRES 90% WORN REQUIRE A MANDATORY STOP."
560 PRINT"TYRE STOPS TAKE 1 MINUTE."
570 PRINT"FUEL IS REPLENISHED DURING TYRE STOPS."
580 PRINT"PUNCTURES ARE MORE LIKELY WITH WORN TYRES."

```

```

590 PRINT
600 INPUT "DO SPECTATORS WISH TO BET ON THE OUTCOME OF THE RACE (Y/N)";C$
610 IFC$="Y" THEN GR=1
620 IF GR=1 THEN GOSUB 2830
630 IF GR=1 GOTO 650
640 CLS:PRINT "CAR" TAB(20) "DRIVER" TAB(40) "BEST PRACTICE LAP":GOTO 660
650 CLS:PRINT "CAR" TAB(10) "DRIVER" TAB(20) "BEST PRACTICE LAP" TAB(44) "BOOKIE'S ODDS
"
660 PRINT "-----":PRINT
670 FOR X=1 TO 10
680 L(X)=RND(25)+20
690 IFL(X)<30 THEN MOD(X)=INT(L(X)/9)
700 IFL(X)<40 AND L(X)=>30 THEN MOD(X)=INT(L(X)/6)
710 IFL(X)=>40 THEN MOD(X)=INT(L(X)/5)
720 P(X)=L(X)
730 NEXT X
740 GOSUB 4530
750 FOR PP=1 TO 10
760 IF GR=1 GOTO 780
770 PRINT EE(PP) TAB(20) N$(EE(PP)) TAB(40) "2 MINS. "L(EE(PP))" SECS.":GOTO 790
780 PRINT EE(PP) TAB(10) N$(EE(PP)) TAB(20) "2 MINS. "L(EE(PP))" SECS." TAB(46) OD(EE(P
P))":1"
790 NEXT PP
800 EE=0:AA=0:BB=0:PP=0
810 GOSUB 50
820 IF GR=1 THEN GOSUB 2900
830 CLS:PRINT "THE RACE IS READY TO START.":PRINT
840 PRINT "PREPARE TO ENTER YOUR AVERAGE SPEED FOR THE FIRST FIVE
MINUTES OF THE RACE. RESULTS WILL THEN BE CALCULATED AND YOU CAN ESTIMATE YOUR SP
EED FOR THE NEXT PERIOD...ETC.":PRINT
850 PRINT "PLEASE TYPE IN YOUR PLANNED AVERAGE SPEED (124 TO 158 KM/HR)"
860 FOR X=1 TO NU
870 PRINT "CAR" X, "N$(X)";
880 INPUT SP(X)
890 L(X)=(6.172/SP(X))*3600
900 IFL(X)<140 OR L(X)>180 GOTO 920
910 GOTO 930
920 PRINT "INVALID LAP TIME":GOTO 870
930 L(X)=L(X)-120
940 P(X)=L(X)
950 NEXT X
960 FOR X=1 TO 10
970 F(X)=100
980 FOR W=1 TO 4
990 T(X,W)=0
1000 NEXT W
1010 NEXT X
1020 Q=5
1030 CLS:PRINT "INCIDENT REPORT. RACE DURATION "Q" MINUTES.":PRINT
1040 FOR X=1 TO 10
1050 PRINT "CAR" X, "N$(X)" ";
1060 HA=0:H=0:S=0:T=0:C=0
1070 IFR(X)<0 GOTO 1380
1080 Z=120
1090 IFL(X)<30 GOTO 1130
1100 IFL(X)<40 GOTO 1140
1110 IFL(X)>50 GOTO 1150
1120 GOTO 1160
1130 Z=Z/3:GOTO 1160
1140 Z=Z/2:GOTO 1160
1150 Z=Z*2:GOTO 1160
1160 C=RND(Z)
1170 IFC=10 GOSUB 2530
1180 IFC=9 GOSUB 2620
1190 IFH=1 GOTO 1380
1200 Z=40
1210 IF(T(X,1)+T(X,2)+T(X,3)+T(X,4))>250 GOTO 1250
1220 IF(T(X,1)+T(X,2)+T(X,3)+T(X,4))<150 GOTO 1260
1230 IF(T(X,1)+T(X,2)+T(X,3)+T(X,4))>350 GOTO 1270
1240 GOTO 1280
1250 Z=Z/2:GOTO 1280
1260 Z=Z*2:GOTO 1280
1270 Z=Z*4:GOTO 1280
1280 B=RND(Z)

```

```

1290 IFB=560SUB2760
1300 IFF(X)<560SUB3780
1310 FORY=1T04
1320 IFT(X,Y)>90GOSUB3860
1330 NEXTY
1340 IFJ(X)=1GOTO1380
1350 C=RND(50)
1360 IFC=1260SUB2390
1370 IFC=2560SUB2500
1380 M(X)=5-S-T+R(X)-O(X)
1390 R(X)=0
1400 IFM(X)<=0THENR(X)=M(X)
1410 IFM(X)<=0THENM(X)=0
1420 IFH=1THENM(X)=RND(4)
1430 IFH=1THENR(X)=R(X)-M(X)
1440 IFAZ(X)=1THENPRINT"***** WITHDRAWN FROM RACE *****":PRINT
1450 IFAZ(X)=1GOTO1580
1460 IFR(X)<0GOTO1480
1470 GOTO1490
1480 IFR(X)>-500THENPRINT"IN PITS FOR A FURTHER: "TAB(35)ABS(R(X))" MINS."
1490 GOSUB2060
1500 D(X)=INT(D(X))+INT(S(X)/60*M(X)*10)/10
1510 A(X)=INT(D(X)/6.172*10)/10
1520 IFAZ(X)=1GOTO1570
1530 IFR(X)<0GOTO1570
1540 FORZ=1T0350:NEXT
1550 IFHA=1ORH=1GOTO1570
1560 PRINTTAB(25)"NO INCIDENT":PRINT:GOTO1580
1570 PRINT
1580 NEXTX
1590 PRINT"PLEASE WAIT... ";
1600 GOSUB4440
1610 GOSUB50
1620 CLS:PRINT"POSITION SUMMARY AFTER "Q" MINS. OF RACE"
1630 PRINT:PRINT"NO."TAB(5)"DRIVER"TAB(20)"LAPS"TAB(26)"DISTANCE"TAB(38)"FUEL(%)"
 "TAB(46)"TYRE WEAR(%)"
1640 PRINT"-----"
1650 IFZZ=1GOTO1690
1660 FORX=1T010
1670 IFA(X)>=163THEN3890
1680 NEXTX
1690 FORX=1T010
1700 GOSUB2420
1710 PRINTTAB(5)N$(EE(X))TAB(19)A(EE(X))TAB(26)D(EE(X))TAB(38)INT(F(EE(X))*10)/1
 0TAB(46)INT(T(EE(X),1));INT(T(EE(X),2));INT(T(EE(X),3));INT(T(EE(X),4))
1720 NEXTX
1730 PRINT"*** = CAR WITHDRAWN. * = CAR IN PITS."
1740 GOSUB50
1750 IFZZ=1GOTO4400
1760 Q=Q+5
1770 IFGR=1THENGOSUB3380
1780 CLS:FORX=1TONU
1790 IFR(X)<=-560T01960
1800 PRINT:PRINTTAB(16)"CAR NO. "X". "N$(X)
1810 PRINTTAB(16)"-----"
1820 PRINT"TYPE IN YOUR SPEED FOR NEXT PERIOD (124 -158 KM/HR)"
1830 INPUTSP(X)
1840 L(X)=(6.172/SP(X))*3600
1850 IFL(X)<140ORL(X)>180THENPRINT"INVALID LAP TIME"
1860 IFL(X)<140ORL(X)>180GOTO1820
1870 PRINT
1880 Y$=" "
1890 L(X)=L(X)-120
1900 PRINT"DO YOU WISH TO STOP FOR FUEL OR TYRES (Y/N)"
1910 PRINT"(TYRES WILL ONLY BE REPLACED IF 50% WORN)"
1920 INPUTY$
1930 O(X)=0
1940 IFY$="Y"GOSUB1980
1950 PRINT
1960 NEXTX
1970 GOTO1030
1980 F(X)=100:PRINT"FUEL TOPPED UP.",
1990 FORY=1T04
2000 IFT(X,Y)>50THENPRINT"TYRE "Y" CHANGED.",

```

```

2010 IFT(X,Y)>50THEN(X,Y)=0
2020 NEXT Y
2030 FOR Z=1 TO 1000:NEXT Z
2040 PRINT:O(X)=1
2050 RETURN
2060 FOR Y=NU+1 TO 10
2070 L(Y)=P(Y)+(RND(5)-3)
2080 NEXT Y
2090 IFM(X)=0THENRETURN
2100 L(X)=L(X)*(1+(T(X,1)+T(X,2)+T(X,3)+T(X,4))/2000)+K(X)
2110 IFH=1GOTO2250
2120 IFM(X)=5GOTO2160
2130 T(X,1)=T(X,1)+2/(5-M(X)):T(X,2)=T(X,2)+1.5/(5-M(X))
2140 T(X,3)=T(X,3)+1.5/(5-M(X)):T(X,4)=T(X,4)+1/(5-M(X))
2150 GOTO2250
2160 IFL(X)<30GOTO2240
2170 IFL(X)<40GOTO2200
2180 T(X,1)=T(X,1)+2:T(X,2)=T(X,2)+1.5:T(X,3)=T(X,3)+1.5:T(X,4)=T(X,4)+1
2190 GOTO2250
2200 T(X,1)=T(X,1)+2.3:T(X,2)=T(X,2)+1.8:T(X,3)=T(X,3)+1.8:T(X,4)=T(X,4)+1.3
2210 GOTO2250
2220 T(X,1)=T(X,1)+1.7:T(X,2)=T(X,2)+1.2:T(X,3)=T(X,3)+1.2:T(X,4)=T(X,4)+.8
2230 GOTO2250
2240 T(X,1)=T(X,1)+2.6:T(X,2)=T(X,2)+2.1:T(X,3)=T(X,3)+2.1:T(X,4)=T(X,4)+1.6
2250 L(X)=L(X)*(1+F(X)/600)
2260 IFM(X)<5THENF(X)=F(X)-(3.5*M(X)/5)
2270 IFH=1THENF(X)=100
2280 IFH=1GOTO2370
2290 IFM(X)<5GOTO2370
2300 IFL(X)<30GOTO2350
2310 IFL(X)<40GOTO2340
2320 IFL(X)>50GOTO2360
2330 F(X)=F(X)-3.5:GOTO2370
2340 F(X)=F(X)-3.8:GOTO2370
2350 F(X)=F(X)-4.1:GOTO2370
2360 F(X)=F(X)-3.2:GOTO2370
2370 S(X)=L(X)+120:S(X)=INT(22219/S(X))
2380 RETURN
2390 PRINT:PRINT"GEARBOX PROBLEMS. ONLY 3RD & 4TH GEARS STILL OPERATING.
SLOWED 10 SECS/LAP.":HA=1
2400 K(X)=10:J(X)=1
2410 RETURN
2420 IFAZ(EE(X))=1GOTO2460
2430 IFR(EE(X))<0GOTO2480
2440 PRINTEE(X);
2450 GOTO2490
2460 PRINT EE(X)"***";
2470 GOTO2490
2480 PRINT EE(X)"*";
2490 RETURN
2500 PRINT:PRINT"CAR NOT HANDLING WELL. SLOWED 5 SECS PER LAP.":HA=1
2510 K(X)=5:J(X)=1
2520 RETURN
2530 CA=RND(6)
2540 ONCAGOTO2550,2570,2570,2600,2600,2600
2550 PRINT:PRINT"MAJOR CRASH. CAR HAS ROLLED IN THE ESSES AND IS A
WRITE OFF. FORTUNATELY "N$(X)" UNHURT."
2560 H=1:S=1000:AZ(X)=1:RETURN
2570 PRINT:PRINT"CAR CRASHED. SPUN INTO FENCE. SIGNIFICANT BODY DAMAGE.
30 MINUTES DELAY TO CARRY OUT REPAIRS."
2580 GOSUB3950
2590 H=1:S=30:RETURN
2600 PRINT:PRINT "SPINOUT. LUCKILY NO DAMAGE. 1 MINUTE DELAY TO REGAIN COMPOSURE
AND RESTART."
2610 HA=1:S=1:RETURN
2620 CA=RND(6)
2630 ONCA GOTO2640,2670,2700,2730,2730,2670
2640 PRINT:PRINT "ENGINE BLOWN. CONROD THROUGH SUMP. ENGINE CHANGE WILL TAKE AT
LEAST 90 MINUTES."
2650 GOSUB3950
2660 H=1:T=90:RETURN
2670 PRINT:PRINT"ELECTRICAL PROBLEMS. ALTERNATOR REPLACED. DELAY 10 MINUTES."

```

```
2680 GOSUB3950
2690 H=1:T=10:RETURN
2700 PRINT:PRINT "DIFFERENTIAL BLOWN. REPLACEMENT WILL TAKE 20 MINUTES."
2710 GOSUB3950
2720 H=1:T=20:RETURN
2730 PRINT:PRINT"SPARK PLUGS OILED UP. 2 MINUTES DELAY.":HA=1
2740 GOSUB3950
2750 T=2:RETURN
2760 PRINT:PRINT"FLAT TYRE. 3 MINUTE DELAY TO CRAWL TO THE PITS TO CHANGE IT.":H
A=1
2770 F(X)=100:PRINT"FUEL TOPPED UP."
2780 Y=RND(4)
2790 PRINT"TYRE "Y" CHANGED."
2800 T(X,Y)=0
2810 GOSUB3960
2820 T=3:RETURN
2830 CLS:PRINT@80,"*** BATHURST BOOKIE ***"
2840 PRINT:PRINT"THE BATHURST BOOKIE WILL ACCEPT BETS FROM UP TO 4 SPECTATORS.
EACH MAY BET ON ANY OR ALL CARS. EACH HAS $100."
2850 PRINT:PRINT"BETS MAY BE PLACED BEFORE THE RACE STARTS, AND AT FIVE
MINUTE INTERVALS DURING THE RACE. ODDS ARE RECALCULATED EVERY
5 MINUTES."
2860 PRINT:PRINT"NO BETS WILL BE ACCEPTED AFTER LAP 120 HAS BEEN COMPLETED.
BETS TO BE IN WHOLE DOLLARS."
2870 PRINT:PRINT"BEWARE - THE BATHURST BOOKIE DRIVES A ROLLS ROYCE.":PRINT
2880 GOSUB50
2890 RETURN
2900 CLS:INPUT"HOW MANY SPECTATORS WISH TO BET ON THE RACE";NR
2910 IFNR>4THEN2930
2920 GOT02940
2930 PRINT"SORRY, MAX. 4 BETTORS.":GOT08100
2940 FORX=1TONR
2950 PRINT"TYPE IN NAME OF BETTOR #"X". MAX. 5 LETTERS";
2960 INPUTPL$(X)
2970 IFLEN(PL$(X))>5THENGOT02990
2980 GOT03000
2990 PRINT"NAME TOO LONG":GOT02950
3000 CR(X)=100
3010 NEXTX
3020 GOSUB3040
3030 RETURN
3040 FORX=1TONR
3050 CS=CS+CR(X)
3060 NEXTX
3070 IFCS=0THENRETURN
3080 CS=0
3090 FORZ1=1TONR
3100 IFCR(Z1)<=0GOT03360
3110 CLS:PRINT"CAR";
3120 Y=0
3130 FORX=1T010
3140 PRINTTAB(5+Y)EE(X);
3150 Y=Y+6
3160 NEXTX
3170 PRINT:PRINT"LAPS";
3180 Y=0
3190 FOR X=1T010
3200 PRINTTAB(5+Y)INT(A(EE(X)));
3210 Y=Y+6
3220 NEXT X
3230 PRINT:PRINT"ODDS";
3240 Y=0
3250 FOR X=1T010
3260 PRINTTAB(5+Y)INT(OD(EE(X)));
3270 Y=Y+6
3280 NEXTX
3290 PRINT:PRINT"$BET";
3300 Y=0
3310 FORX=1T010
3320 PRINTTAB(5+Y)BR(Z1,EE(X));
3330 Y=Y+6
3340 NEXTX
3350 GOSUB3600
3360 NEXTZ1
```

```

3370 RETURN
3380 CLS:PRINTTAB(12)"BATHURST BOOKIE - STATUS REPORT"
3390 GOSUB3810
3400 PRINTTAB(13)"AMOUNT BET/POTENTIAL WINNINGS"
3410 PRINT:PRINT"CAR"TAB(6)"LAPS"TAB(13)"ODDS"TAB(20)PL$(1);
3420 IFNR>1THENPRINTTAB(31)PL$(2);
3430 IFNR>2THENPRINTTAB(42)PL$(3);
3440 IF NR>3THENPRINTTAB(53)PL$(4);
3450 PRINT:PRINT"-----"
3460 FORX=1TO10
3470 GOSUB2420
3480 PRINTTAB(5)A(EE(X))TAB(12)INT(OD(EE(X)))":1"TAB(19)BR(1,EE(X))"/"WR(1,EE(X)
);
3490 IFNR>1THENPRINTTAB(30)BR(2,EE(X))"/"WR(2,EE(X));
3500 IFNR>2THENPRINTTAB(41)BR(3,EE(X))"/"WR(3,EE(X));
3510 IFNR>3THENPRINTTAB(52)BR(4,EE(X))"/"WR(4,EE(X));
3520 PRINT
3530 NEXTX
3540 IFA(EE(1))>120GOTO3560
3550 GOT03570
3560 GOSUB50:RETURN
3570 GOSUB50
3580 GOSUB3040
3590 RETURN
3600 PRINT:PRINT"BOOKIE NOW OPEN!! BETTOR NO."Z1", "PL$(Z1)", YOUR BETS PLEASE.
TYPE CAR NO., AMOUNT OF BET (EG 5,30). TYPE 'NEW LINE' TO END.":PRINT
3610 FORX=1TO10
3620 BS=0:Z2=0
3630 IF CR(Z1)<=0THEN3770
3640 PRINT"FUNDS AVAILABLE $"CR(Z1)". YOUR BET";
3650 INPUT Z2,BS
3660 IF BS=0ANDZ2=0GOTO3770
3670 IF Z2<0ORZ2>100RBS<10RBS>CR(Z1)GOTO3710
3680 IFOD(Z2)=999GOTO3720
3690 IFZ2=0GOTO3685
3700 GOT03730
3710 PRINT"INVALID ENTRY - TRY AGAIN":GOTO3620
3720 PRINT"SORRY, CAR WITHDRAWN - TRY AGAIN":GOTO3620
3730 CR(Z1)=CR(Z1)-BS
3740 BR(Z1,Z2)=BR(Z1,Z2)+BS
3750 WR(Z1,Z2)=WR(Z1,Z2)+(BS*OD(Z2))
3760 NEXTX
3770 RETURN
3780 PRINT:PRINT"PITTED FOR FUEL - TYRES CHECKED. 1 MIN. DELAY.":HA=1
3790 GOSUB3950
3800 T=1:RETURN
3810 FORX=1TO10
3820 OD(EE(X))=INT((2*X)+(A(EE(1))-A(EE(X))))/2
3830 IFAZ(EE(X))=1THENOD(EE(X))=999
3840 NEXTX
3850 RETURN
3860 PRINT:PRINT "PITTED FOR TYRES. 1 MIN. DELAY.":HA=1
3870 GOSUB3950
3880 T=1:RETURN
3890 CLS:PRINT@33PRINT,"LAP 163 HAS BEEN COMPLETED!!!"
3900 FORX=1TO1000:NEXTX:GOSUB4000
3910 PRINT:PRINT"IF MORE THAN ONE CAR COMPLETES LAP 163, THE CAR COVERING
THE GREATEST DISTANCE WILL BE DECLARED THE WINNER."
3920 PRINT@330,"FINAL SUMMARY FOLLOWS"
3930 FORX= 1TO2000:NEXTX
3940 ZZ=1:GOTO1630
3950 F(X)=100:PRINT"FUEL TOPPED UP.",
3960 FORY=1TO4
3970 IFT(X,Y)>50THENPRINT"TYRE "Y" CHANGED.",
3980 IFT(X,Y)>50THEN(X,Y)=0
3990 NEXTY:PRINT:RETURN
4000 CLS:FORY=1TO10
4010 FORX=1TO25
4020 SET(X,Y)
4030 NEXTX
4040 FORX=50TO75
4050 SET(X,Y)
4060 NEXTX
4070 FORX=100TO125

```

```

4080 SET(X,Y)
4090 NEXTX
4100 NEXTY
4110 FORY=11T020
4120 FORX=26T049
4130 SET(X,Y)
4140 NEXTX
4150 FORX=76T099
4160 SET(X,Y)
4170 NEXTX
4180 NEXTY
4190 FORY=21T030
4200 FORX=1T025
4210 SET(X,Y)
4220 NEXTX
4230 FORX=50T075
4240 SET(X,Y)
4250 NEXTX
4260 FORX=100T0125
4270 SET(X,Y)
4280 NEXTX
4290 NEXTY
4300 FORY=31T040
4310 FORX=26T049
4320 SET(X,Y)
4330 NEXTX
4340 FORX=76T099
4350 SET(X,Y)
4360 NEXTX
4370 NEXTY
4380 FORX=1T0 2000:NEXT X:CLS
4390 RETURN
4400 CLS:FORX=1T014
4410 PRINT"***** " N$(EE(1))" WINS THE 1981 HARDIE-FERODO 1000 *****"
4420 NEXTX
4430 GOTO4620
4440 FORAA=1T010
4450 PP=1
4460 FORBB=1T010
4470 IFD(AA)<D(BB)THENPP=PP+1
4480 IFD(AA)=D(BB)ANDAA>BBTHENPP=PP+1
4490 NEXTBB
4500 EE(PP)=AA
4510 NEXTAA
4520 RETURN
4530 FORAA=1T010
4540 PP=1
4550 FORBB=1T010
4560 IFL(AA)>L(BB)THENPP=PP+1
4570 IFL(AA)=L(BB)ANDAA>BBTHENPP=PP+1
4580 NEXTBB
4590 EE(PP)=AA
4600 NEXTAA
4610 RETURN
4620 IFGR=1GOTO4650
4630 PRINT:PRINT"THE RACE IS FINISHED - THANKS FOR PARTICIPATING"
4640 GOTO4810
4650 GOSUB50
4660 CLS:PRINT"BATHURST BOOKIE - FINAL SUMMARY"
4670 PRINT:PRINT"BETTOR"TAB(20)"AMOUNT BET (TOTAL)"TAB(40)"AMOUNT WON"
4680 PRINT"-----"
4690 TA=100*NR
4700 FORX=1TONR
4710 PRINTPL$(X)TAB(20)(100-CR(X))TAB(40)WR(X,EE(1))
4720 WA=WA+WR(X,EE(1))
4730 TA=TA-CR(X)
4740 NEXT X
4750 PRINT"TOTAL AMOUNT BET $"TA
4760 PRINT"TOTAL AMOUNT WON $"WA
4770 IF(TA-WA)>0THEN4790
4780 PRINT"A SAD DAY FOR THE BATHURST BOOKIE":GOTO4630
4790 PRINT"YET ANOTHER VICTORY FOR THE BATHURST BOOKIE"
4800 GOTO4630
4810 END

```

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

Next month's issue will contain at least the following programs plus the usual features and articles.

\*\* AN ADVENTURE IN LEVEL I \*\*

That's right - someone has finally written an adventure program in LI BASIC and it fits into a 4K machine. Next month you'll be able to get lost in a forest.

\*\* FRUSTRATION - LI/4K \*\*

This game was in Issue 2 as a Level II program. Now all those people who wanted to test their reflexes way back then can do it.

\*\* SETTING NON-GRAPHICS LII/4K \*\*

As requested, here is a subroutine that you can add to your program to "SET" non-graphics.

\*\* GRAND PRIX LII/16K \*\*

Another game straight from the sideshows. You drive a high speed racing car down the screen, dodging cars coming the opposite way.

\*\* VARIABLE WORKSHEET LII/4K \*\*

This program will do away with the problem of forgetting which variables you have used in your program.

\*\* 80 COMPOSER LII/16K \*\*

This program will let you compose and play music on your 80 with full musical capabilities.

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

To MICRO-80 Tick where appropriate.

Please consider the enclosed program for...

- (i) Publication in MICRO-80 .....
- (ii) Publication on disk or cassette only .....
- (iii) Both .....
- Name.....
- Address .....
- .....
- .....Post Code.....

\*\*\*\* CHECK LIST \*\*\*\*

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

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

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

The changes or improvements that you think may improve it.

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





# MICRO-80

## LEVEL II ROM REFERENCE MANUAL

by Edwin Paay

Published by MICRO-80 PRODUCTS

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

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

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

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

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

# MICRO-80