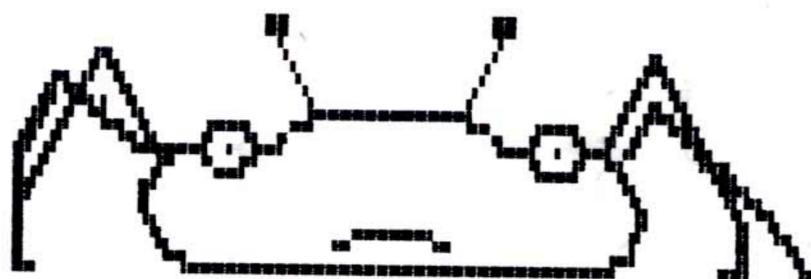




**BLOWFLY**  
A truly  
disgusting  
game



Also in this issue:

**HARDWARE**

Add Joysticks and Input/Output Ports to your '80 — Part 1

**PROGRAMMING:**

Better Basic Programming — Part 3

**SOFTWARE:**

Level 1 Cheque Book  
Disk Index  
Mileage Calculator  
Degree Conversions

Level 1 Share Graph  
Starshoot  
Genius  
Bingo

# MICRO-80

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

EDITOR:	IAN VAGG
SOFTWARE EDITOR:	CHARLIE BARTLETT
HARDWARE EDITOR:	EDWIN PAAY
U.K. CORRESPONDENT:	TONY EDWARDS

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
BETTER BASIC PROGRAMMING - PART 3	3
JOYSTICKS AND INPUT/OUTPUT PORTS FOR YOUR '80 - PART 1	7
MICRO BUGS	9
NEWS FROM MICRO-80 PRODUCTS	9
MARKET PLACE	10
<u>SOFTWARE SECTION</u>	
SHARE GRAPH.....L1/4K	11
CHEQUE BOOK DATA FILE.....L1/4K	12
BLOWFLY.....L2/4K	14
MILEAGE CALCULATOR.....L2/4K	15
CONVERSIONS.....L2/4K	22
STAR SHOOT.....L2/4K	24
BINGO.....L2/4K	26
GENIUS.....L2/16K	27
DISK INDEX.....DISK/32K	32
MICRO-80 PRODUCTS	17
NEXT MONTH'S ISSUE	35
CASSETTE EDITION INDEX	36
ORDER FORM	36

MICRO-80 is registered by Australia Post - Publication SQB 2207 Category B

AUSTRALIAN OFFICE AND EDITOR:

MICRO-80, P.O. BOX 213, GOODWOOD, SOUTH AUSTRALIA, 5034. TEL.(08) 211 7244

U.K. SUBSCRIPTION DEPT:

24 WOODHILL PARK, PEMBURY, TUNBRIDGE WELLS, KENT TN2 4NW.

Printed by:

Shovel & Bull Printers, 312A Unley Road, HYDE PARK, S.A. 5061.

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

★ ★ ★ FREE SOFTWARE OFFER ★ ★ ★

EVERY NEW AUSTRALASIAN SUBSCRIBER TO MICRO-80 WILL RECEIVE A FREE CASSETTE CONTAINING THREE LEVEL I AND THREE LEVEL II PROGRAMS PLUS COMPREHENSIVE DOCUMENTATION ... THE RETAIL VALUE OF THE SOFTWARE WOULD EXCEED THE COST OF THE SUBSCRIPTION!!!

\*\*\*\*\* EDITORIAL \*\*\*\*\*

## IS TANDY SLIPPING?

Latest word from the U.S.A. is that the Apple II is, in that quaint way Americans express these things "about to become the most installed computer in the world". Presumably this means that the Apple II is about to outsell the TRS-80. It is interesting to speculate how a computer, designed in a garage by six young men aged around 21, could outsell a computer designed by a team of professionals working for a corporation with an annual turnover in excess of US\$1,000,000,000 and having thousands of captive retail outlets around the world. Is it in the design of the computer itself, or does it have something to do with the marketing strategies of the two corporations? We are biased of course but, in our estimation, the TRS-80 is a better and more versatile computer than the Apple with the exception of its graphics resolution and ability to display in colour. Certainly, feature for feature, the TRS-80 is considerably cheaper than the Apple and it is a more economical computer to own; Apple peripherals and peripheral adaptors are quite expensive. So, if it isn't price and it isn't computing power, what is it?

The answer to that one, in our opinion, is marketing strategy. When the TRS-80 was launched, Tandy owned a formidable distribution network for high technology consumer goods. As a result, sales of the TRS-80 rocketed. Apple, on the other hand, had no such distribution network but there were independent computer stores including two or three franchised chains which were only too glad to handle such an advanced (for its time) well presented computer. So Apple sold to them. The cost of developing disk controllers, DOS's, etc. was and is high, but still only a drop in the bucket for Tandy, so Tandy did these things itself. So did Apple to some extent, but Apple realised that it could not fully exploit its own product during the limited lifetime such a high technology device would have, so it made system information readily available to all comers and encouraged independent suppliers to develop peripherals, software, plug-in cards etc. etc. I happened to be in the local Computerland store recently and idly picked up a fairly thick manual lying on the counter. It turned out to be a directory of Apple add-ons made by a wide range of independent suppliers. The directory was produced under the auspices of the Apple Corporation. Can you imagine Tandy doing that?

One of the nice things about Apple's distribution network was that the store managers and salesmen etc. were all interested and knowledgeable in computers. Tandy's store managers, as a group, were not. Why should they be? Even today, computers represent only 13% of Tandy's sales. Even more important, most independent computer stores are managed by their owner who has a very real profit motive. Tandy's answer to this was to establish a second, specialised distribution network - the Computer Centres. These Centres culled the more interested store managers from the general retail stores, gave them more training and allowed them to specialise - a sure recipe for success, right! Well, maybe. Remember, they still only had Tandy products to sell. They still did not have a real profit motive and being a big corporation, a lot of political infighting developed - no-one likes an elitist corp, unless he is part of it.

Then free enterprise started to sneak in to the Tandy scene. Despite a dearth of technical information, some cheeky upstarts actually started producing superior disk operating systems, better and cheaper disk add-ons, much, much better software and so on. The authorised dealer networks Tandy had established, protected by fair trading legislation, began cutting prices of TRS-80 computers and even supplying other, non-Tandy dealers. At this point, some corporations might have recognised the inevitable and opened up the doors to independent dealers. Not Tandy. Now saddled with an expensive distribution network dedicated to computers, Tandy attempted to tighten the reins. The real turning point came, however, when companies such as Microsoft, the developers of the TRS-80 BASIC Interpreter, until then almost synonymous with Tandy and the Z-80, actually started producing Apple products; and what products they were too. The Z-80 softcard, which converted the Apple to a Z-80 computer and makes most of Microsoft's very respectable library of Z-80 machine language programs suitable for use on the Apple. A genuine case of the mountain coming to Mohammed! Now the Apple can run under CP/M, too, an industry standard disk operating system which opens up a veritable mountain of applications software, particularly in the business arena.

Around about the same time came Visicalc. I know of several companies that have bought Visicalc first and foremost, and a computer to run it as a necessary evil! Sure, Tandy now sells Visicalc for the TRS-80 but how many people relate Visicalc to Tandy? Very few indeed. Think Visicalc, think Apple.

Then Tandy brought out the TRS-80 Model III. A very nice computer indeed. A refined version of the Model I but not totally compatible. In the 12 months or so since the Model III was released in the States, how many new programs have you seen advertised for it - very, very few. Even Microsoft, who will be releasing a disk version of Editor/Assembler Plus for the Model I any day now, does not have any firm plans for developing an Editor/Assembler for the Model III. One U.S. software house puts it this way - "dealers are not interested in selling TRS-80 software. 12 months ago we were supplying 250 dealers of whom 220 took TRS-80 software. Today, we are supplying over 1,000 dealers of whom less than 100 take TRS-80 software". It seems that Tandy's strategy is being successful, the independent dealers are abandoning sales of Tandy-related products for Apple and others. Of course, as a result Tandy is losing market share to Apple but, presumably small is beautiful and that must be part of Fort Worth's grand strategy (I'll bet!)

It seems Tandy has lost the battle. But it certainly hasn't lost the war of course. The Apple II is exhibiting all the signs of a mature product nearing the end of its life cycle. Its replacement, the Apple III, is already well behind schedule and seems to be a lemon. Apple is now a public company of some 1600 employees and is suffering extreme growing pains. It is not inconceivable that the Apple Corporation will go under (no, we are not trying to start a nasty rumour but Apple has enormous problems to overcome). Tandy has long ago learned how to conduct its affairs as a large corporation. The Model III will never achieve the market share of its predecessor, the Model I - there are too many other contenders - but it will be supported by many independents (not least ourselves). It does seem a shame, however, that such a good computer is being let down by misguided marketing policies.

\*\* NEWDOS 80 VERSION 2.0 \*\*

Newdos 80 Version 2.0 is now being delivered. It is a substantial rewrite of Version 1.0. The manual is even thicker, it comes in a pretty blue colour and covers both the Model I and Model III DOS's. However, you have to buy the Model III version separately from the Model I version. Double-density operation is now included as an option, you can read or write 40 track diskettes on 80 track drives and the system is even more flexible than previously. If you have Version 1.0 already, you may upgrade to Version 2.0 by sending your original disk plus US\$70 to Apparat Inc., 4401 South Tamarac Parkway, Denver, CO.80237 U.S.A. \$60 is for the upgrade and \$10 for airmail postage. This service is only available from Apparat so please do not send your disk to us here. Micro-80 Products will soon have stocks of Newdos 80 Version 2.0 for both the Model I and Model III for those who have not yet taken the plunge. We will shortly publish a comparative review of the new and old versions.

- 0000000000 -

\*\*\*\*\* BETTER BASIC PROGRAMMING - PART 3 - by Rod Stevenson \*\*\*\*\*

\*\*\*\*\* CHECKING INPUT \*\*\*\*\*

Before we start on this month's topic, there are a couple of matters to deal with from previous articles.

\*\*\* EXTRA EXPLANATION OF INKEY\$ STRING \*\*\*

When we asked for readers' feedback we really did mean it. So we are pleased to answer this point here, which was not covered in the article on strings. Incidentally, there are many other little hints, tips, and quirks, which we intend to cover in the scheduled article about the more advanced aspects of all these elementary subjects. But we are more than pleased to amplify any requests as they arise - at least it shows someone is reading!

When we discussed the INKEY\$ string function, we said that INKEY\$ string is cleared by each access, and will be lost unless a string variable has been allocated to hold its value. So much is true. What we didn't point out was that the INKEY\$ string routine constantly scans the keyboard for a key pressed: it doesn't just do it when the program gets to the particular line contained in the function. So what this means is that INKEY\$ string routine will accept an input at any time up until it is called, and only when it is called will it pass the value to the assigned string variable. In effect, it anticipates.

Whether or not this is a good side effect depends on the program's use of the INKEY\$ string function, and a moment's thought will open alarming possibilities. A good one: to be able to anticipate requests for a single letter input when one wants to skip through to a certain point in a program. A bad one: when one is using INKEY\$ string as a time delay to allow for reading of the screen - a keybounce will convince INKEY\$ string that 2 pages are required.

The cure is to put 0 into the INKEY\$ string storage location (4099HEX,16537DECIMAL) just before the line which accesses the INKEY\$ string function. This will void any keys pressed in advance and only accept them while actually handling the loop containing INKEY\$ string. An example: while in the above screen-reading period, immediately after presenting the current screen-display, have a delay loop such as FOR I = 1 TO 1000: NEXT which will cover any possible keybounce, then POKE 16537,0 to clear INKEY\$ string, then do the loop which will wait for the next key to be pressed to continue.

\*\* CORRECTION \*\*

In our first article on strings we advised you that RIGHT\$ starts counting from the left whereas of course it starts counting from the right, so that if you say RIGHT\$(I\$,2) you will get the two right hand characters in I\$ only. We offer our apologies for misleading you and hope that the damage is not irreparable.

Now on with this month's topic.

## INTRODUCTION

We consider this to be the MOST IMPORTANT aspect of programming. If we were standing in front of you, we would jump up and down and pound on anything in sight to impress our unqualified belief of its importance. Lack of such is the reason for the bad public image of computers, large and small. We've all heard of garbage in/garbage out! We've heard of human operator error! But how often have we heard of the programmer neglecting input-checking? Whatever the reason, (lack of time, lack of money to pay for time, lack of expertise, or simply disinterest) we can see no acceptable excuse for blaming someone else when your own program bombs out because of an unexpected (even if it is erroneous and should never have been) input. How many times have you given a program to one of your faithful followers and had to excuse yourself with: "well, it always ran all right for me". Or have you come back to a program some time later and forgotten what input is required? Or even such a simple thing as wanting to quickly skip through to a certain point and never getting there because in your haste you offered the wrong input?

We hope the above diatribe will impress on every reader that we mean it! Now to specifics.

## A SIMPLE FIRST STEP

With no effort at all you can put a prompting comment on the INPUT line, rather than just have a question mark appear on the screen without even the elementary advice whether a number or a letter is required. A line such as INPUT"INSTRUCTIONS(Y/N)";I\$, is not much more difficult surely than simply INPUTI\$? Yet it is amazing how often and how many don't bother. We know our BASIC is very forgiving with messages such as REDO,??, EXTRAIGNORED; at the very least it will mess up the (carefully planned?) screen display. And BASIC will only handle such outrageously wrong input as above - you really can't expect it to know you were looking for a four digit number with no digit greater than 6, and only one decimal point which should be after the third digit? But, and be honest, how often have you done just this (or the equivalent) and wondered why your program falls apart?

Even such simple (to computerists) messages as those above will be useless to an untrained operator. And why should a computer have to be operated by a programmer only? The program should never drop out into BASIC. Every possible irregularity should be covered. When checking-out your masterpiece, by far the greatest checking should comprise testing all manner of input. Obviously input as expected; but also right at the limits of that expected, just outside those limits, widely outside, and totally wrong in the wrong format. And, of course, no input at all. Just pressing ENTER will leave the input variable as it is - it depends on the program whether or not this is O.K., but certain beware that it will happen, and check. If it matters you can zero the variable just prior to the INPUT. Because, of course, any input checking will not detect that the input has not been changed.

## THE INPUT-ROUTINE

Before any input can be checked it must be input. And right from this point it is necessary to be aware of the traps for the unwary. Of course, the BREAK key will have been disabled (and more on this in a later article) but such things as accidental presses of the comma can be disastrous. So, really all input should be accepted, then sorted through after. No, it's not enough to input everything as a string and then convert it by VAL if required.

Disk-basic and level III users are fortunate in having LINE INPUT, which will accept anything until ENTER. So we have written a short and simple routine to provide this for level II users. All it does is provide input facility for up to 200 characters of any sort, then return to the program for input checking as required.

We will explain this routine fully and provide the source-code in the article on assembly language. Suffice it to say that we wrote the routine in assembly, and then converted it into BASIC to use as a routine. Certainly it can be improved - we are no Eddy Paay! But it does show the sort of thing that is within reach of everyone, and we hope by the time we've imparted the assembly episode, you will all agree.

Now perhaps is a good time to advocate Bill Barden's book "Assembly Language Programming for the TRS-80" available from Tandy stores for \$3.95. Also to suggest that if you are anxious to get into assembly and haven't an assembler, that you invest in the MICROSOFT Editor/Assembler Plus which is so superior to Tandy's (and cheaper!) that we purchased it, even though we had the Tandy version! (Although MICRO-80 PRODUCTS does sell it, (\$37.50 plus postage) truly we do not recommend it only for this reason. For example, we also recommend PACKER which MICRO-80 PRODUCTS does not sell). And it comes with a superb manual which does not assume you are familiar with the workings of any assembler. It does not, though, teach assembly - Barden's book will get you well and truly down that road. But, yes, you will still need our forthcoming article on assembly, which will explain and perhaps concentrate more on how to use assembly language with BASIC, because as we have said earlier, we assume that you will have the very minimum knowledge of most of the functions which we mention in this series.

```

0 * ROUTINE TO GIVE THE DISK-BASIC "LINE-INPUT" FUNCTION IN LEVEL II. EXAMPLES
OF DIFFERENT TYPES OF INPUT-CHECKING.
1 CLS:POKE16561,32700AND255:POKE16562,INT(32700/256):CLEAR500:FORI=32700TO32733:
READML:POKEI,ML:NEXT:DATA217,205,127,10,14,0,205,73,0,254,13,40,13,205,51,0,119,
12,121,254,200,48,3,35,24,236,121,217,38,0,111,195,154,10
2 *CLS:CLEAR500:FORI=1TO34:READML:ML%=ML%+CHR$(ML):NEXT:DATA217,205,127,10,14,0,
205,73,0,254,13,40,13,205,51,0,119,12,121,254,200,48,3,35,24,236,121,217,38,0,11
1,195,154,10
100 *THIS IS START OF A SIMPLE PROGRAM TO TEST NUMERIC INPUT WITHIN EXPECTED RAN
GE.
110 PRINT:PRINT"THIS DEMONSTRATES CHECKING THAT INPUT IS BETWEEN 1 AND 56
120 GOSUB 32760 *GET INPUT
130 IN = VAL(IN$) *CHANGE INPUT TO NUMERIC VALUE
140 IF IN<1 OR IN>56 THEN PRINT"INVALID INPUT":GOTO120 *CHECK THAT INPUT IS IN
EXPECTED RANGE
150 PRINT"YOU INPUT"IN;"WHICH IS ACCEPTABLE" *DISPLAY INPUT
155 PRINT
160 PRINT"ANOTHER DEMONSTRATION? (Y/N)":GOSUB32760:IFLEFT$(IN$,1)="Y"THENGOTO110
*ALLOWS FOR ANOTHER OF THE SAME
170 PRINT
180 PRINT
200 PRINT"ROUTINE TO CHECK SINGLE-CHARACTER AGAINST ALL ALLOWABLE INPUT."
210 PRINT"INPUT ALL ALLOWABLE INPUT: ";GOSUB32760:AL$=IN$ *GETS ALL ALLOWABLE
INPUT
220 PRINT"INPUT TO BE CHECKED: ";GOSUB32760
225 IFLEN(IN$)<>1THENPRINT"REQUIRES SINGLE-CHARACTER, SO IS ";GOTO250
230 FORI=1TOLEN(AL$):IFIN$=MID$(AL$,I,1)THENI=LEN(AL$):NEXT:GOTO260
240 NEXTI
250 PRINT"INVALID INPUT":GOTO220
260 PRINT"YOU INPUT "IN$", WHICH IS ALLOWABLE INPUT"
270 PRINT
280 PRINT"ANOTHER DEMONSTRATION? (Y/N)":GOSUB32760:IFLEFT$(IN$,1)<>"Y"THENGOTO30
0 *ALLOWS FOR ANOTHER OF THE SAME
290 PRINT"SAME ALLOWED INPUT? (Y/N)": ";GOSUB32760:PRINT:IFLEFT$(IN$,1)="Y"THENG
OTO220ELSEGOTO200 *ALLOWS CHOICE OF CHANGING ALLOWED INPUT
300 PRINT:PRINT:PRINT"ROUTINE TO CHECK SETS OF INPUT AGAINST ALLOWED SETS."
310 PRINT"INPUT ALL ALLOWED INPUT: ";GOSUB32760:AL$=IN$
320 PRINT"INPUT ALLOWED NUMBER OF CHARACTERS: ";GOSUB32760:AL=VAL(IN$):IFAL<=00
RAL<>INT(AL)THENGOTO320
330 PRINT"INPUT CHARACTERS TO BE CHECKED: ";GOSUB32760:IFLEN(IN$)<>ALTHENPRINTA
L;"CHARACTERS REQUIRED, SO ";GOTO360
340 FORI=1TOLEN(AL$)STEPAL:IFMID$(AL$,I,AL)=IN$THENI=LEN(AL$):NEXT:GOTO370
350 NEXTI
360 PRINT"INVALID INPUT, ":GOTO330
370 PRINT"YOU INPUT "IN$" WHICH IS ALLOWABLE INPUT"
380 PRINT"ANOTHER DEMONSTRATION? (Y/N)": ";GOSUB32760:IFLEFT$(IN$,1)<>"Y"THENGOT
O400
390 PRINT"SAME ALLOWED INPUT? (Y/N)":GOSUB32760:PRINT:IFLEFT$(IN$,1)="Y"THENGOTO
330ELSEGOTO300
400 PRINT:PRINT:PRINT"ROUTINE TO CHECK MONETARY INPUT."
410 GOSUB32760:IFVAL(IN$)=0THENGOTO460
420 D=0:FORI=1TOLEN(IN$):IFMID$(IN$,I,1)="."ANDI<>LEN(IN$)-2THENGOTO460
422 IFVAL(MID$(IN$,I,1))=0THENGOTO460
425 IFMID$(IN$,I,1)="."THEND=D+1
430 NEXTI
440 IFD=0THENIN$=IN$+".00"
450 PRINT"YOU INPUT $"IN$" WHICH IS ALLOWABLE INPUT":GOTO470
460 PRINT"INVALID INPUT, ";GOTO410
470 PRINT"MORE OF THE SAME? (Y/N) ";GOSUB32760:PRINT:IFLEFT$(IN$,1)="Y"THENGOTO
410
480 X=RND(4):ONXGOTO100,200,300,400
32760 POKE16526,32700AND255:POKE16527,INT(32700/256):IN$=STRING$(200,65):AD=VARP
TR(IN$):AD=PEEK(AD+1)+PEEK(AD+2)*256:PRINT"INPUT UP TO 200 CHARACTERS?":X=USR(AD
):PRINT:PRINT"THE INPUT WAS: ";PRINTLEFT$(IN$,X):IN$=LEFT$(IN$,X)
32761 *AD=VARPTR(ML$):POKE16526,PEEK(AD+1):POKE16527,PEEK(AD+2):IN$=STRING$(200,
65):AD=VARPTR(IN$):AD=PEEK(AD+1)+PEEK(AD+2)*256:?"INPUT UP TO 200 CHARRACTERS?":
X=USR(AD):?:?"THE INPUT WAS: ";?:LEFT$(IN$,X):IN$=LEFT$(IN$,X)
32762 RETURN

```

The above program is intended only to illustrate the use of our input routine and how the routine is checked by the main program itself. We wouldn't expect you to use the whole thing - we have numbered the "bones" of it as line 1 and 32760 to allow any normal program to be inserted between.

Although we said above that we are not going to explain the workings, or how we called the routine from BASIC, we will explain why we've provided alternative lines 2 and 32761, which are protected by '(REM). As it is, the program sets its own memory size to 32700 then uses that protected area above 32700 for the machine-language routine. And theory tells us this is the way to do it. But we personally like to keep our machine-language in a string, which is looked after by BASIC, and is independent of memory size or disk memory space or other machine language routines already resident. This is what the two alternative lines do. Theory tells us they ought not to work - try them by removing the '(REM), and inserting one at the start of line 1 and line 32760.

Just why we say theory tells us they will fail, we'll explain in the later episode (yes, the assembly one!) We will also explain why some programs keep their machine-language in a string and some in REM statements - no, it's not individual preference!

Note also that after an IF THEN statement we say THEN GOTO which could perhaps be shortened to just THEN but, as we have pointed out to you in earlier articles, we suggest that you don't shorten unless you are absolutely sure of the ramifications of shortening, and that you do use the correct shortened version. It could also be shortened to IF GOTO or sometimes just IF and the number. Also, notice the lack of spaces between letters. This is how we write our programs because this is how we think of them. In fact, the computer doesn't mind whether you put the spaces in or not on most occasions. Beware, however of the expression GOSUBXXX: in a multi statement line. Some computers (but not all) will revert to Memory Size? Ready? on encountering the space before the colon. Make sure the colon is immediately behind the line number viz. GOSUBXXX:

#### A Warning.

We have found this program shows up a hardware fault in some CPUs. You may find that the first time you run it, the line which says: "THEINPUTWAS:" does not show what the input was, even though it was echoed to the screen as it was input. However, after BREAKing and RUNning again, all is well. We have no idea what is going on here. The owners of such CPUs tell us that the same thing happens with other programs. We must confess though, we were worried until we came to the conclusion that it is a hardware problem, therefore, not to be worried about (by us!)

Another point, though not actually a fault, is that it will show up keyboard bounce as would be expected. It will also do unpleasant things if the contact on the ENTER key is worn and you don't press it properly - a delicate flick of the key may not be enough! But this merely emphasises the validity of the practice of saving a program as it's typed in, and certainly before it's run - if one of the data numbers is typed incorrectly, you could lose the lot!

#### DETAILING THE ROUTINES

100 - 160 This is the first and simplest case. To check that simple numeric input is between acceptable limits. The limits of 1 and 56 mentioned in line 110, and checked for in line 140, are arbitrary - they could just as well have been input by the routine.

Line 130 uses VAL to convert a string to a numeric value, and here is a chance to test how various numbers are converted. Try mixing alpha characters. Try putting in a decimal point. Try for double precision. And try anything else that you would like because it really is a good opportunity to get your understanding of VAL really established.

Line 160 allows for a repeat of this same routine before progressing.

This program also shows our particular personal style of writing - i.e. we tend to handle each thought as a line on its own. So line 110 tells what is going on, line 120 gets the input, line 130 changes it, line 140 checks it, line 150 displays it, line 160 finishes off the section. No, we didn't really write lines 1 and 32760 as they appear - we wrote them in manageable thoughts, then used PACKER to create a single line for the practical convenience of our gentle readers.

200 - 290 This is only slightly more complex than the first routine. It allows you to input a set of data which represents the allowable inputs (e.g. ABCEX), then checks actual inputs against this data!

As in the first example, each line is basically one thought, which should be apparent by observation.

Line 210 gets allowable input as IN\$, but then changes it to AL\$ because the input routine will also get the input as IN\$.

Line 225 checks that only a single character is input, as this is what is wanted in this case.

Line 230 checks that the character is contained somewhere in AL\$, and falls through to "INVALID INPUT" if it is not. However, if it is, the FOR...NEXT loop is closed before going to line 260 which displays the input.

Line 280 allows for another of the same.

Yes, we anticipate the question: "Why not do the line 280 as a subroutine instead of 4 times in the program?" The reason is for clarity (we hope), and because of the different lines to go to if YES. The saving would be small anyway.

300 - 390 This routine is decidedly more complex as it checks sets of input, against expected sets. E.G. days of the week could be input as MONTUEWED, etc. then the length allowed set to 3. Or directions, NORSOUWESEAS. And we suspect this is actually the usual form input will be used in a program - even STAR TREK has LRS SRS NAV COM, etc.

Again, each line handles one basic thought. Line 310 gets allowed input and calls it AL\$, line 320 gets allowed number of characters and calls it AL. It also checks that input was a numerical integer. Note the use of meaningful variable names IN and IN\$ for input, AL and AL\$ for allowed characters. Only because we find it easier to write like this, not because the computer cares! Line 340 checks the input against the allowed input in steps of AL. If the length of input were not critical, the STEPAL could be omitted and the AL in the MID\$ changed to 1, so every character input would be checked against the allowed characters, but not for sequence.

Of course, this routine can still be used to check numeric input, but in a character- rather than numeric-mode. Such as to check for dates and days: "MON12TUE13SAT17".

400 - 470 A simple routine to check monetary input - really only to illustrate how to check the decimal point's position and that there is only one. Of course, the first routine could be incorporated to check input within a certain range, but still check for decimal point.

Line 480 is simply a 'frill' we added, rather than say STOP.

#### CONCLUSION

We hope this article has shown that input-checking is easy, is within the capability of all, and above all, is VITAL. We really can't stress it enough.

Next episode will be an easier and shorter one on the (almost) self-teaching subject of screen-formatting. Virtually only a trial and error process but with a few simple hints which perhaps you thought were not worth trying because of the certainty they wouldn't have worked!

- 000000000 -

\*\*\*\*\* JOYSTICKS AND INPUT/OUTPUT PORTS FOR YOUR '80 - PART 1

by Alan Dent \*\*\*\*\*

It is now over 12 months since we hinted at a project which would enable you to add joysticks to your '80. Unfortunately, that project turned out to be unreliable, so we did not proceed with it. We are now happy to announce that Allan Dent, one of our Adelaide readers, has, after months of work, developed an analogue-to-digital converter board which, amongst other things, will enable you to use joysticks with your '80. The complete theory of operation, constructional details and printed circuit board design, will be presented in a three part series, starting this month. Now, over to Allan for Part 1.

When I bought my secondhand TRS-80 MODEL 1 about two years ago, the previous owner also gave me a piece of hardware that he had made. It was a very simple output port and consisted of three standard chips. One was a 74LS42 one of ten BCD decoder wired as an address decoder, by using the three low addresses and the OUT pulse. The selected output address from this chip strobed two CD4042 CMOS quad latches whose inputs were wired across the DATA BUSS. The outputs were wired directly to LED indicators to show the state of the latched output port. Although I am a Technical Officer in electronics with quite a few years' experience, I was surprised to see such a simple but effective interface. I have used various microprocessor evaluation kits and their interfaces are usually much more complicated. My port did have one downfall though, its address repeated every 8 addresses, i.e. 1, 9, 17, 25 etc. After studying the TRS-80 TECHNICAL MANUAL and with my simple interface in my hand, I soon reaffirmed what I already knew. That is that computers are not a mysterious "black box" to be held in awe, but basically just an extension of existing digital technology compressed into a few LSI chips. My appetite was whetted for a more sophisticated interface, so I constructed one on a piece of VERO board. I then demonstrated both units at the next meeting of the ADELAIDE USERS' GROUP to show fellow '80 owners how simply it could be done. Since then, I have made several small interfaces and demonstrated them at our monthly meetings. One of my later interfaces was set up to display port 255 as a demonstration of the cassette output and motor control signals along with the 32/64 character per line signal, all of which use port 255. My main thrust is to try to make the hardware side of computing less daunting to the uninitiated.

I have since been "invited" onto the committee of the USERS' GROUP as the Projects and Hardware Co-ordinator. One of my recent demonstrations was an interface to perform ANALOGUE to DIGITAL conversion. It had sixteen analogue inputs which were individually addressable and gave a digital output between 0 and 255 (8 Binary Bits). My demonstration was fairly simple. I connected a JOYSTICK to analogue inputs 0 and 1 and with a simple program, moved a spot around the screen.

The response to my demonstration was very gratifying, and I was asked if I could design a printed circuit board so that other '80 owners in our group could connect joysticks to their computers. As I usually just throw together "one off" circuits on VERO board or wire wrap, this presented me with a challenge.

I costed the exercise and thought that it would be an even better value-for-money project if I included some INPUT and OUTPUT PORTS in the design. I decided it would be best to design the circuit to be as flexible as possible, allowing builders to include only the parts they were interested in. The final design contained eight inputs to the ANALOGUE to DIGITAL converter and six parallel INPUT/OUTPUT PORTS. The constructor has the choice of an eight input A/D converter and two groups of three parallel I/O ports. Any of the above can be included or left out of the finished P/C board as desired. This means that a person who only wants to play games with his JOYSTICKS is not paying for ports he probably will not use and vice versa. All A/D inputs and I/O ports are at PORT ADDRESSES. This makes programming relatively simple for either the BASIC or ASSEMBLY LANGUAGE programmer. No interrupts or special subroutines are required for the A/D conversion, one simply requests an input from the selected analogue input port address and the digital value of the analogue input to that port is returned to the CPU. At the moment the P/C board is designed for the TRS-80 Model I (my machine) with facilities to allow connections to either the SYSTEM-80 or the TRS-80 Model III - in fact, any 8080 or Z80 based microcomputer.

The A/D converter used in this project is a NATIONAL SEMICONDUCTOR chip type INS0809. It is usually operated in the ratiometric mode. This means that the reference voltage applied to the A/D chip is also used to supply the voltage to the joystick pots. The analogue voltage from these pots is then converted to a digital value which is a ratio of the percentage of the full scale value (255) and is not necessarily related to an actual voltage as would be the case in a digital voltmeter. For example, the reference voltage is a nominal 5 volts, the full scale digital output is 255. As you can see, there is no convenient voltage relationship between the input and the digital output (19.8mV per BIT). If the reference voltage was reduced to 2.55 Volts, then the digital output would have a direct relationship to actual voltage of 10mV per digital bit, and it could then be classed as a simple digital voltmeter. The circuit board has options included which allow the owner to set the top and bottom reference voltage levels. This means that one can set the A/D reference to suit the individual designer who wants to do more with his A/D inputs than just play games. The industrial controller, for example, can set his reference points to be 1 and 5 volts, thus setting up the system for a 4 to 20mA current input across the normal 250 ohm resistor used in such systems.

The six PARALLEL INPUT/OUTPUT PORTS are implemented by using two identical LSI chips which are programmable and may be set up to perform various functions. The chip used is a PROGRAMMABLE PERIPHERAL INTERFACE (PPI) chip type 8255. It was designed by INTEL as one of a series of support chips for their 8080 MICRO-PROCESSOR and, as such, is also Z80 compatible. The PPI chip has four internal 8 bit registers, three are used as parallel 8 bit I/O ports, A, B and C, with the fourth register being used as the CONTROL REGISTER. Port C can also be split into two, four bit ports, C upper and C lower. It may be programmed to perform three different MODES of operation, but for this exercise we will only be discussing MODE 0. This is the basic input/output mode and using the two 8 bit and two 4 bit ports, there are 16 possible I/O combinations. An example may be port A, B and C upper, programmed as input ports, with C lower as an output port. The combination of I/O ports is set up by writing a CONTROL WORD into the CONTROL REGISTER. This then sets up the internal BUSS structure within the 8255 PPI to perform the desired I/O function. Once programmed by the control word, the 8255 stays in that condition unless it is reprogrammed with a new control word or RESET by the system. Port C, when in the output condition, may also be programmed to either SET or RESET any of the individual 8 output pins with only one OUT instruction. This ability to change individual pins, independently of the other 7, reduces system software especially in control applications. The other MODES of operation are, Mode 1, Strobed Input/Output, and Mode 2, Bi-Directional Bus.

Anybody wishing to extend their knowledge on the above chips should consult the manufacturer's books or data sheets. The INS0809 is covered by two N/S publications that I know of, the DATA ACQUISITION HANDBOOK and the CMOS DATA HANDBOOK while the 8255 is described in the SERIES 8000 MICROPROCESSOR FAMILY HANDBOOK, also by N/S. I have no doubt that INTEL also have available a data book covering the 8255 PPI. Both N/S and INTEL should also have available DATA SHEETS and APPLICATION NOTES on their chips. These are usually available from the manufacturers or their agents.

When I gave a demonstration of my finished P/C board at a recent ADELAIDE USERS' GROUP meeting, I had several demonstration programs to put it through its paces. I started by simply moving a dot around the screen with a single joystick. This was followed by a modified version of the same program to draw pictures on the screen also from the joystick. Then I moved onto the parallel I/O ports and sensed input switch positions and output data to some HEX LED displays. My final demonstration program was one that switched 240 Volt lights under keyboard control. This was a rather dramatic indication to the members at the meeting that all the things that I had previously said were not just words. Their computer can actually be put to use as a controller of many things. As you can see, the uses for such an interface will only be limited by your imagination. For a little stimulation, try to envisage your model train layout under computer control. You could use some of your input ports to sense train positions around the track, your output ports could control your loco speed and direction and also switch the points. In fact, you could connect up several slide pots to your A/D inputs and use them as throttle controls for your trains. As I said, the possibilities are endless.

## \*\*\*\*\* MICRO-BUGS \*\*\*\*\*

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

80 COMPOSER - Issue 17 Pages 24-28.

The trouble we were having with our renumbering at the time, affected this program quite badly. In most conditional statements, a spurious 10 was inserted after the "THEN" and sometimes but not always, after the "ELSE". For example, line 310 reads:

```
310 1FRIGHT$(K$,1)="H"THEN10H=1 : etc.
```

It should read:

```
310 1FRIGHT$(K$,1)="H"THENH=1 : etc.
```

Line 330 should read:

```
330 FORC=1TO21:1FN$(C)="END"THEN390ELSEIFN$(C)<>K$THENNEXT
```

The following lines were affected by this bug:

```
250 THEN ELSE          370 THEN
310 THEN              460 ELSE
320 THEN              630 ELSE
330 ELSE THEN        640 ELSE
360 THEN
```

BETTER BASIC PROGRAMMING - PART 2.

Somehow, in the cutting and pasting that goes into setting up the magazine prior to publication we managed to mangle the listing which illustrated the text in this article. The correct listings are shown below:

```
90 CLEAR 300 'PLENTY OF STRING SPACE
100 DIM A$(4,5) 'SIZE OF EXAMPLE ARRAY
110 FOR I = 1 TO 4 'X DIMENSION
120 FOR J = 1 TO 5 'Y DIMENSION

600 FOR I = 1 TO 4: FOR J = 1 TO 5 'DO EVERY ARRAY ELEMENT
610 FORL=LEN(A$(I,J))+1 TO 12 'PACK EACH ELEMENT TO 12
620 A$(I,J)=A$(I,J)+" " :NEXTL 'ADD ONE SPACE AT A TIME
630 NEXTJ, I
635 ' PRINTOUT THE ARRAY IN ITS NEW FORMAT
640 PRINT:FORI=1TO4: FORJ=1TO5: PRINTA$(I,J);:NEXTJ: PRINT:NEXTI
- 0000000000 -
```

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

The recent increases in Sales Tax, the continued strength of the US\$ and cost increases in general, have forced us to raise many of our prices. This is the first general increase since MICRO-80 PRODUCTS commenced business 18 months ago and we are sure you will agree that our new prices are still very good value for money. The price rises will be effective from October 1st, 1981. Any orders sent to us prior to that date (according to the postmark) will be accepted at the old prices.

The demand for our interfaces based on the LNW Research board far exceeded our expectations. This, coupled with back-ordering by LNW themselves and the slow down in US air traffic meant that some of our customers have had to wait a considerable time before receiving their units. We are happy to report that we are now catching up and hope to be able to supply from stock in a few weeks' time.

We have a number of exciting new products in stock which have arrived too late to be included in the advertisements in this issue. Firstly, there is Orchestra 80. You may have seen advertisements for this product in the US magazines. It is not just another music generating program but also includes a hardware sine-wave synthesizer which plugs on to the expansion port on the rear of the TRS-80 keyboard or the SYSPAND 80 (sorry, it is not suitable for connecting directly to a SYSTEM 80). This combination of hardware and software produces the most realistic music possible from your '80. It has a much wider frequency range than systems which use the cassette port and, at \$89 + \$2.00 p&p represents excellent value.

MICRO-80 PRODUCTS has been appointed Australian distributor for SNAPP INC. software. SNAPP INC. produce high quality utilities for both Model II and Model III TRS-80's. We have virtually the full range of SNAPP INC. software available ex stock - call for prices and details.

- 0000000000 -

\*\*\*\*\* 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

\*\*\*\*\*
- AXION 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 \$680.00 Mr. M. Gosbee  
Ph. (02) 88-4468

\*\*\*\*\*
- SYSTEM-80 L2/16K with full documentation, approx. 40 blank cassettes (data), books, magazines, and programs (most games) including EDTASM PLUS & Adventure No. 9 \$720.00 o.n.o. Mr. R. Glucz,  
1/1707 Dandenong Rd.  
Oakleigh, Vic. 3166  
Ph.(03)543 3302 A.H.

\*\*\*\*\*
- 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 perline. 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, incl. BMON, RPN Calc. Sargon 3.5 Chess, Space Invaders, 70 Maths/Finance Programs. Excellent condition \$680.00 G. Hobbes,  
42 Nareen Parade,  
Narabeen, N.S.W. 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" B&W 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

\*\*\*\*\*
- Tandy QUICK PRINTER 11. 120 LPM Upper/lower case on 2-3/8" aluminised paper. As new & 4 rolls of paper incl. \$200.00

INTERFACE to suit System-80 \$ 40.00

\*\*\*\*\*
- TEXAS INSTRUMENTS TI59 Calculator & PC100C Thermal printer. The calculator has a card reader, 960 program steps & up to 100 memories. I have 2 extra plug in modules (each with about 22 programs) & some other software for it \$300.00 Dennis Bareis,  
286 Lennox Street,  
Maryborough, Q. 4650  
Ph. (071)22 1699

\*\*\*\*\*
- ITOH DOT MATRIX PRINTER. 80 columns wide with cable for TRS-80. Good working condition. \$750.00 Mr. Pat Browne,  
90 Muriel Drive  
Pooraka. S.A. 5095  
Ph. 258 1555 Bus.  
262 4020 A/H

\*\*\*\*\*
- MICROTEK MT 32 with 32K memory, an Exatron Stringy Floppy and 15 wafers. All excellent condition \$500.00 the lot Brian J. Fillery,  
Box 1790, G.P.O.  
Brisbane, Q. 4001

\*\*\*\*\*
- Tandy Model III complete with 48K RAM, line printer (7) and System Desk. \$1850.00 Mike Mooney  
Ph. (02) 768112 A.H.  
(02) 821943 Bus.

\*\*\*\*\*
- SYSTEM 80 16K Computer with video screen, Audio amplifier and 15 issues of Micro-80 plus software cassettes plus Adventure, Space and strategy games, Light Pen 2 Sound off (value \$100's) \$900.00 freight free G. Jackson  
P.O. Box 1012,  
Nhulunbuy, N.T.  
Ph. (089) 87 2983

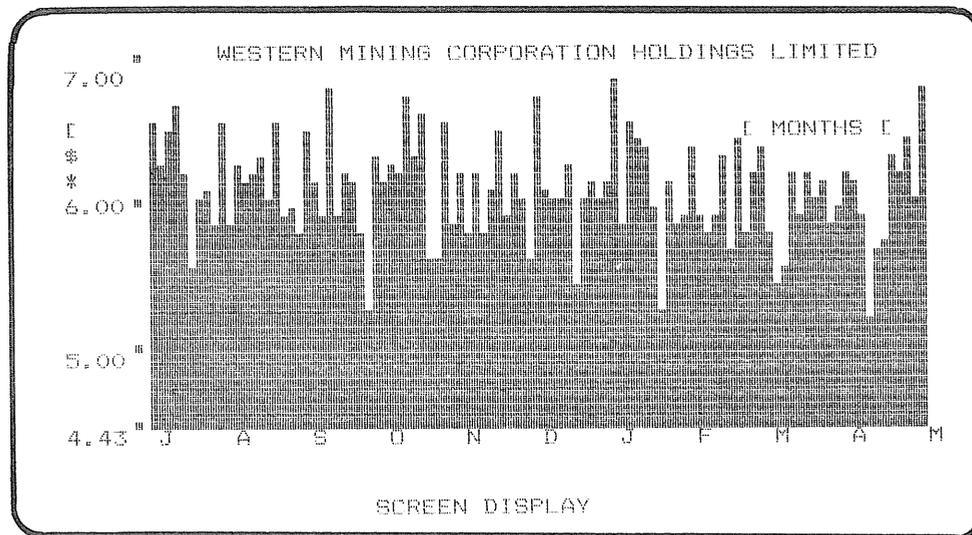
\*\*\*\*\*

\*\*\*\*\* SHARE GRAPH LI/4K

(C) L.W. Tappenden \*\*\*\*\*

This program is most useful in seeing when a share price is too expensive or too cheap. It is also handy for estimating percentage rises and falls over a given time. To adapt it for your own use, you must replace the data in the program with data of your own as each item of data represents the price of the share on that day.

Instructions for modifying the program to suit your own requirements are contained in the REM statements.



```

0 ' ***** SHARE GRAPH *****
  * L. TAPPENDEN *
  * 77 TADSTAN DR. *
  * TULLAMARINE *
  *****
1 REM EACH ITEM OF DATA REPRESENTS THE PRICE OF THE SHARE ON
2 REM THAT DAY. THIS INFORMATION IS THEN MODIFIED BY THE
3 REM PROGRAM AND PLACED BACK ON THE SCREEN IN GRAPHICAL FORM.
4 REM AS YOU CAN SEE WE HAVE LOTS OF MEMORY LEFT TO ADD MORE
5 REM DATA. AS YOU ADD MORE DATA FROM DAY TO DAY ALTER LINE
6 REM 100 SO D=(TOTAL NUMBER OF PIECES OF DATA). WHEN YOU HAVE
7 REM TOO MUCH DATA THE GRAPH WILL START TO OVER PRINT ON THE
8 REM LEFT OF THE SCREEN. ALTER THE 2'S IN LINES 105 160 AND
9 REM 195 TO READ 3. THUS PRESENTING ALL THE HISTORY AT ONCE
10 D.550,550,526,534,540,530,518,500,502,502,524,526,508,510
12 D.512,520,520,524,530,530,540,536,538,540,544,550,568,590
14 D.590,598,598,614,618,610,616,622,624,616,622,604,590,580
18 D.560,584,596,580,570,600,584,570,584,590,606,614,614,602
20 D.600,608,600,580,584,586,586,584,570,550,540,556,564,560
22 D.550,546,580,564,640,610,626,630,638,630,634,624,638,660
24 D.656,656,666,656,640,650,646,658,676,676,660,666,570,668
28 D.656,620,616,606,620,604,578,578,580,558,560,550,550,560
30 D.542,558,580,580,574,568,580,580,580,560,556,562,536,528
32 D.510,485,492,485,482,480,480,480,470,470,485,500,485,468
34 D.465,460,445,448,470,475,460,450,458,460,458,458,445,450
36 D.457,455,465,485,475,490,526,510,526,534,530,546,538,538
38 D.528,532,530,528,526,532,546,560,560,548,544,544,550,546
40 D.548,536,524,512,512,520,522,520,512,512,495,490,500,495
42 D.490,485,490,485,493,500,520
65 CLS
70 P.AT10,"WESTERN MINING CORPORATION HOLDINGS LIMITED"
74 P.AT64,"7.00";
75 SET(9,1)
76 P.AT200,"["
77 P.AT264,"$"
78 P.AT328,"\"
80 P.AT384,"6.00"
82 SET(9,18)
85 P.AT768,"5.00"
87 SET(9,35)
90 P.AT960,"4.43 J A S O N D J F M A M";
    
```

```

92 SET(9,44)
95 REM 70 TO 92 SETS HEADING AND COORDINATES
100 D=203
101 REM D IS THE NUMBER OF PIECES OF DATA YOU HAVE IN THE
102 REM PROGRAMME.
105 D=D/2
110 H=707
115 REM H SETS THE HIGH POINT ON THE GRAPH
120 X=10
125 REM 110 + 120 POSITION THE GRAPH ON THE SCREEN
130 F.N=1TOD
140 X=X+1
150 A=0
160 F.M=1T02
170 READ B
180 A=B+A
190 N.M
195 A=A/2
197 REM 150 TO 195 ADDS UP AND AVERAGES 2 PIECES OF DATA
200 A=A.(A-H)
202 REM 200 SETS THE DISTANCE FROM THE HIGH POINT ON THE GRAPH
205 A=A/6
207 REM 205 SQUASHES THE RESULT SO IT FITS ON THE SCREEN
210 GOS.4000
215 REM GOSUB 4000 DRAWS THE DATA ON THE SCREEN IN THE FORM OF
216 REM A BAR GRAPH
220 N.N
300 P.AT236,"1 MONTHS ^";
400 G.400
4000 F.Y=44TOAS.-1
4010 SET(X,Y)
4020 N.Y
4030 RETURN
5000 REM OF COURSE,, THE IDEA IS TO BUILD UP A HISTORY OF THE
5001 REM COMPANIES IN WHICH YOU ARE INTERESTED AND THIS
5002 REM PROGRAM SHOULD ASSIST YOUR BUY OR SELL DECISION.
5003 REM DON'T FORGET TO CSAVE YOUR UPDATED PROGRAM.
5004 REM KEEP DESTROYING REM STATEMENTS AS YOU NEED ROOM
5005 REM FOR MORE DATA.
5006 REM L. W. TAPPENDEN.
5007 REM 77 TADSTAN DRIVE,
5008 REM TULLAMARINE. VIC. 3043.
5009 REM PH. (03) 338 1505

```

---

\*\*\*\*\* CHEQUE BOOK DATA FILE LI/4K

(C) K. Ford \*\*\*\*\*

This program enables you to save all your cheque book transactions on a cassette data file and to retrieve them at will. It is best to use a separate cassette for your data file and not to rewind it after each entry. This means that you will not have to list all cheques before entering your latest cheque. When starting a new cheque book alter line 3050 so that it contains the number of cheques in the old book, e.g.

```

FROM          TO
F.T.=1T.P    F.T.=1T.P+25

```

This enables the computer to retrieve all past cheques using the numbers on your present book.

For those who may want to adapt the program to their own needs, here is an explanation of some of the lines.

LINE	FUNCTION
20	Do you want the previous cheques listed? If YES goto 30 and 40 if NO goto 45
30	Find out the number of cheques used.
40	gosub 3000-4000
45	Get current balance.
70	Holds program until ready to continue.
110	Any more cheques to deposit, if YES gosub 1000 if NO goto 900
900	Give current balance and END
1000	Collect all cheque data and return
2000	Instructs user how to set, then records the data and returns
3000	Instructions on how to set the cassette.
4000	Retrieves data and prints it.

```

1 REM          *** CHEQUE BOOK DATA FILE ***
2 REM          *      (C) K.L. FORD      *
3 REM          *      P.O. BOX 1457      *
4 REM          *      KATHERINE 5780     *
5 REM          *      N.T.               *
6 REM          *****
7 R=0:B=0:M=0:C=0
10 A=128:Y=1:N=0
15 P.A.23,"CHEQUE BOOK DATA FILE":F.X=46T.87:S.(X,3):N.X
20 P.A.910,"DO YOU WANT ALL PREVIOUS CHEQUES LISTED";:IN.Z
22 P.A.910,"
25 IFZ=0G.45
30 P.A.910,"WHAT IS THE NUMBER OF THE FIRST CHEQUE. ";:IN.K
35 P.A.910,"WHAT IS THE NUMBER OF THE LAST CHEQUE USED.":;IN.I
40 P=(I-K)+1:GOS.3000:G.50
45 P.A.910,"WHAT IS THE CURRENT BALANCE.":;IN.B
47 P.A.910,"
50 GOS.1000
60 GOS.4000
70 P.A.910,"PRESS ENTER WHEN READY TO CONTINUE.":;IN.A#
80 CLS
100 GOS.2000
110 P.A.910,"HAVE YOU ANY MORE CHEQUES TO ENTER";:IN.Z
120 IFZ=0G.900
125 CLS:R=0:GOS.4000
130 GOS.1000
140 GOS.4000
150 P.A.910,"PRESS ENTER WHEN READY TO CONTINUE.":;IN.A#
160 G.80
900 CLS:P.A.336,"YOUR CURRENT BALANCE IS $";B
999 E.
1000 P.A.947,"*";
1005 P.A.974,"(MAX. 16 CHARACTERS)";
1010 P.A.910,"DATE AND PARTICULARS";:IN.B#
1015 P.A.974,"
1020 P.A.910,"DEPOSITS IN DOLLARS AND CENTS.":;IN.D
1030 P.A.910,"
1100 P.A.910,"CHEQUE NUMBER";:IN.C
1120 P.A.910,"AMOUNT OF CHEQUE";:IN.M
1130 B=(B+D)-M
1140 IFB<1T.B=1.(B*100)/100
1150 RET.
2000 P.A.212,"SET CASSETTE TO END OF LAST CHEQUE"
2010 P.A.276,"ENTERED AND PRESS PLAY AND RECORD "
2020 P.A.340,"BUTTONS. THE COMPUTER WILL NOW"
2030 P.A.404,"RECORD YOUR LATEST FIGURES."
2045 P.A.660,"PRESS ENTER WHEN READY TO RECORD";:IN.A#
2047 P.A.910,"NOW RECORDING";
2050 P.#B#,"";C#,"";M#,"";D#,"";B
2060 RET.
3000 P.A.212,"REWIND CASSETTE RECORDER TO START OF"
3010 P.A.276,"DATA FILE AND PRESS PLAY BUTTON,"
3020 P.A.340,"THE COMPUTER WILL NOW RETRIEVE"
3030 P.A.404,"ALL CHEQUE BOOK DATA."
3040 P.A.532,"PRESS ENTER WHEN READY TO RETRIEVE.":;IN.A#
3045 CLS
3047 P.A.20,"RETRIEVING DATA"
3050 F.T=1T.P
3060 IN.#B#,C,M,D,B
3080 GOS.4000
3100 N.T
3110 RET.
4000 P.A.A+1,"NUMBER";:P.A.A+11,"PARTICULARS";:P.A.A+31,"AMOUNT";
4010 P.A.A+41,"DEPOSITS";:P.A.A+53,"BALANCE"
4020 R=R+64
4030 P.A.A+R+30,"$";:P.A.A+R+40,"$";:P.A.A+R+52,"$";
4040 P.A.A+R+1,C;:P.A.A+R+11,B#;:P.A.A+R+31,M;
4050 P.A.A+R+41,D;:P.A.A+R+53,B;
4055 P.A.A+R+64,"
4060 IF R=704T.R=0
4070 RET.

```

\*\*\*\*\* BLOWFLY LII/4K-16K (C) Carl Cranstone \*\*\*\*\*

The object of this very funny game is for one of the players to be first to get his fly to the top of the window. However, there are a few hazards.

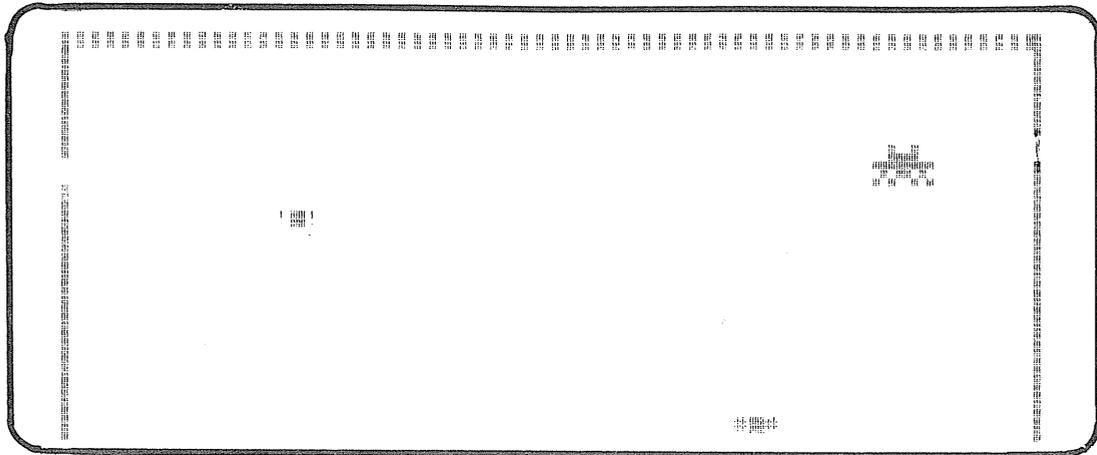
Use the following keys to control your fly:-

DIRECTION	TRS-80	SYSTEM-80
UP	UP ARROW (left player)	ESC (left player)
UP	U (right player)	U (right player)
DOWN	DOWN ARROW (left player)	CNTRL (left player)
DOWN	J (right player)	J (right player)

Using these keys you must climb your fly up the window without getting swatted or gobbled up by the spider. If you stay in one spot for too long you will start to slide down the window.

What else can I say about this program but...YUK !!!

The graphics on the front cover are only illustrative of this game. The actual screen presentation is shown below.



```

10 ***** BLOWFLY *****
   * C. CRANSTONE *
   * 17 HELEN ST *
   * CHRISTIES DOWNS *
   *****
20 CLS:PRINT@20,"B L O W F L Y";
CARL CRANSTONE 1981 JULY **** "BLOWFLY" (C)
   N.B. 4K USERS DELETE LINES 10 TO 80
30 PRINT:PRINT"THIS GAME IS FOR TWO PLAYERS.":PRINT"EACH PLAYER
HAS A FLY WITH WHICH THEY MUST CRAWL TO THE TOP OF THE WINDOW."
40 PRINT"LEFT PLAYER USES 'L' (ESC) TO CRAWL UP AND RIGHT PLAYER
   USES 'U'"
50 PRINT:PRINT"YOU MUST CRAWL UP THE WINDOW DODGING THE SWATS.":
PRINT"A SPIDER LIVES ON THE WINDOW AND MAY COME DOWN TO GOBBLE YO
U UP!"
60 PRINT"IF YOU STAY IN ONE SPOT FOR TOO LONG, YOU WILL START TO
   SLIDE DOWN!"
70 PRINT"YOU CAN ALSO MOVE DOWN IF YOU GET COLD FEELERS!":PRINT"
LEFT PLAYER PRESS DOWN ARROW (CNTRL) AND RIGHT PLAYER 'J'."
80 INPUT"PRESS ENTER TO START";LK$
90 CLS: CLEAR200:P1$="!" +CHR$(143)+"!";P2$="#" +CHR$(143)+"#";P1=9
74:P2=1004:SP$=" " +CHR$(184)+CHR$(180)+" " +CHR$(26)+STRING$(5,2
4)+CHR$(167)+CHR$(167)+CHR$(155)+CHR$(155)
100 PS$=" " +CHR$(180)+CHR$(184)+" " +CHR$(26)+STRING$(5,24)+CHR
$(155)+CHR$(155)+CHR$(167)+CHR$(167)
110 PRINTCHR$(143)" LEVEL 0 - 3 ";CHR$(143);:INPUTLEV:IFLEV<0ORL
EV>3THEN90
120 PRINT@960+14,P1$;:PRINT@960+44,P2$;:PRINT@0,STRING$(64,133);
130 IFK$="PASS"THENRETURN
140 FORQ=1TOLEV*10:NEXTQ:IR$=INKEY$:IFIR$="L"THENGOSUB250:GOSUB2
70
150 IFPEEK(14440)=16THENGOSUB250:GOSUB290
160 IFIR$="U"THENGOSUB260:GOSUB310
170 IFPEEK(14383)=4THENGOSUB260:GOSUB330
180 IFRND(2)=2THENFORQ=1TOLEV+4:JK=RND(832)+63:PRINT@JK,"SWAT!";
:FORQ=1TOLEV*10:NEXTQ:PRINT@JK," ";:GOSUB220:NEXTQ
190 IFRND(8)=2THENGOSUB480
200 IFRND(4)=2THENGOSUB250:GOSUB290ELSE IFRND(4)=1THENGOSUB260:GO
SUB330
    
```

```

210 GOTO140
220 IFF1=JKORP1=JK+10RP1=JK+20RP1=JK+30RP1=JK+40THENLO$="PLAYER ONE":GOSUB380:RETURN
230 IFF2=JKORP2=JK+10RP2=JK+20RP2=JK+30RP2=JK+40THENLO$="PLAYER TWO":GOSUB380:RETURN
240 RETURN
250 PRINT@P1," ";:RETURN
260 PRINT@P2," ";:RETURN
270 P1=P1-128:IFF1<14THENWI$="PLAYER ONE":GOSUB350ELSEPRINT@P1,P1$;:RETURNELSERETURN
280 RETURN
290 P1=P1+64:IFF1>974THENP1=974ELSEPRINT@P1,P1$;:RETURNELSERETURN
300 PRINT@974,P1$;:RETURN
310 P2=P2-128:IFF2<44THENWI$="PLAYER TWO":GOSUB350ELSEPRINT@P2,P2$;:RETURN
320 RETURN
330 P2=P2+64:IFF2>1004THENP2=1004ELSEPRINT@P2,P2$;:RETURN
340 PRINT@1004,P2$;:RETURN
350 IFWI$="PLAYER ONE"THENK$="PASS":CLS:GOSUB120:K$="":W1=W1+1:IFW1>=10THEN430ELSEPRINT@26,W1;:PRINT@1,Q1;:PRINT@36,W2;:PRINT@60,Q2;:P1=974:P2=1004:RETURN
360 IFWI$="PLAYER TWO"THENK$="PASS":CLS:GOSUB120:K$="":W2=W2+1:IFW2>=10THEN430ELSEPRINT@36,W2;:PRINT@60,Q2;:PRINT@1,Q1;:PRINT@26,W1;:P2=1004:P1=974:RETURN
370 RETURNELSERETURN
380 IFLO$="PLAYER ONE"THENGOSUB250:P1=1000:GOSUB290:Q1=Q1+1:IFQ1>=10THEN400ELSEPRINT@1,Q1;:RETURN
390 IFLO$="PLAYER TWO"THENGOSUB260:P2=1005:GOSUB330:Q2=Q2+1:IFQ2>=10THEN400ELSEPRINT@60,Q2;:RETURN
400 CLS:PRINTCHR$(143)" BLOWIE NUMBER ";RIGHT$(LO$,3)" LOST! ";CHR$(143)
410 IFRIGHT$(LO$,3)="TWO"THENWI$="PLAYER ONE"ELSEIFRIGHT$(LO$,3)="ONE"THENWI$="PLAYER TWO"
420 BV=2:GOTO430
430 IFBV<>2THENCLS
440 PRINTCHR$(143)" THE WINNING BLOWIE WAS ";CHR$(143);:BV=0
450 PRINT" FLY-NUMBER "RIGHT$(WI$,3)" ";CHR$(143):PRINT:PRINT"PLAY AGAIN?"
460 PRINT"(FLY TALK FOR PLAY AGAIN!) BZZZ.. BZZZ..";:INPUTFLY$
470 IFLLEFT$(FL$,1)="Y"THENRUNELSEEND
480 SP=RND(56)+64
490 FORH=SPTOSP+1024STEP64
500 PRINT@H,SP$;:FORJ=1TO20:NEXTJ:PRINT@H,PS$;:FORJ=1TO20:NEXTJ:PRINT@H," ";CHR$(26);STRING$(6,24)" ";
510 IFH>=832THENS50
520 IFPEEK(P1+15360)<>ASC("!")ORPEEK(P1+15361)<>143ORPEEK(P1+15362)<>ASC("!")THENPRINT@H,"BURP!!";:FORP=1TO50:NEXTP:PRINT@H," ";LO$="PLAYER ONE":GOSUB380:GOTO140
530 IFPEEK(P2+15360)<>ASC("#")ORPEEK(P2+15361)<>143ORPEEK(P2+15362)<>ASC("#")THENPRINT@H,"BURP!!";:FORP=1TO50:NEXTP:PRINT@H," ";LO$="PLAYER TWO":GOSUB390:GOTO140
540 NEXTH
550 RETURNELSERETURN

```

\*\*\*\*\* MILEAGE CALCULATOR LII/4K (c) F. Greco \*\*\*\*\*

This program is basically in 4 parts each accessible through a menu driven routine. The menu is shown below.

- 1 - INPUT DATA ROUTINE
- 2 - PROJECTION CHART
- 3 - SUMMARY
- 4 - END

The menu display is controlled by the INKEY\$ function. Entering a 1 directs the computer to the first section of the program. This section requires that the user enter the following :-

- (1) month number
- (2) km at start
- (3) km at end
- (4) litres
- (5) cost

The program then works out :-

- (1) kms travelled
- (2) miles travelled (for those of us who still don't understand metrics).
- (3) kms/litre
- (4) mpg
- (5) litres per 100 km.

After the data for this session has been completed the user is asked if he wants to save the data on tape. After copying the data to tape, the program will ask if you want the data saved again. If you answer "Y" then the program will dump the same data again. Function 2 will take you to the projection chart. Here the user is required to enter the average km/litre that his vehicle does and then how many kms it is to be projected for. The program will then display how many litres will be needed for the trip.

Function 3 is the summary mode. In this part of the program you have two options :-

OPTION 1 - To summarise a particular month.

The program will ask the user which month. After the month is given the program will search the data tape (provided one has been created) and when it strikes the month specified it will load it in and then print out :-

TOTAL COST    TOTAL KM(TRAVELLED)    AVG KM/LITRES    TOTAL LITRES

with the results under each heading. After all the data has been displayed you may press any key to return to the menu.

OPTION 2 - To summarise for a full year.

This section inputs all months from the data tape then prints out the results under the headings above. BUT this section also accumulates the appropriate variables for the whole year and then prints out the totals and also divides the km/litre to give the average for the year.

##NOTE##

When the tape is being read in in the yearly mode, the last month read in must be a 12 or an OD error will result.

```

10 '          *** MILEAGE CALCULATOR ***
          *      FRANK GRECO      *
          *  RIVERVIEW COURT    *
          *  MARIBYRNONG VIC.    *
          *****

20 CLS:PRINT@456,"KILOMETERS PER LITRE/MILES PER GALLON CONVERSI
ON."
30 PRINT@589,"COMPLETELY WRITTEN BY FRANK GRECO."
40 PRINT@717,"MARIBYRNONG VICTORIA,AUSTRALIA,3032"
50 FORX=1 TO 3000:NEXTX
60 GOTO 500
70 CLS:INPUT"PLEASE INPUT MONTH NUMBER ((1 TO 12))";M
80 INPUT"PLEASE INPUT KILOMETERS AT THE START";A
90 INPUT"PLEASE INPUT KILOMETERS AT THE END";B
100 INPUT"PLEASE INPUT HOW MANY LITRES YOU HAVE PUT IN THE TANK"
;F
110 INPUT"PLEASE INPUT COST";CC
120 C=B-A
130 PRINT:PRINT
140 PRINT"YOU HAVE TRAVELLED ";C;" KILOMETERS."
150 PRINT:PRINT
160 REM* CONV KM TO MILES *
170 D=C*0.62
180 PRINT"THAT'S ABOUT ";D;" MILES."
190 PRINT:PRINT"NOW I AM WORKING OUT YOUR PETROL CONSUMPTION."
200 FORX=1 TO 1000:NEXTX
210 CLS
220 REM** PETROL CONS **
230 E=C/F
240 G=E*0.62
250 H=G*4.5
260 REM * LITRES PER 100 KM *
270 FF=F/(C/100)
280 PRINT"--- TOTAL DISTANCE TRAVELLED IS ";C;" KMS."
290 PRINT:PRINT"--- WHICH IS ABOUT ";D;" MILES."
300 PRINT:PRINT"--- PETROL CONS FOR ";C;" KMS IS ";E;" KMS/LITRE
."

```

## 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 \$349**  
**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 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 \$619**

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

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

**SYSPAND-80 - \$97.50 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 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.

**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.

**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.

**DISK DRIVE CABLES SUITABLE FOR ANY DISK DRIVES**

DC-2 2 Drive Connector Cable . . . . . \$39 incl. p&p  
DC-4 4 Drive Connector Cable . . . . . \$49 incl. p&p

**DOUBLE THE SPEED AND CAPACITY OF YOUR DISK DRIVES**

**PERCOM DOUBLER ONLY \$220 plus \$2.00 p&p**

Installing a Doublers 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 installs in the TRS-80 expansion interface, the System-80 expansion interface and the LNW Research expansion interface in a few minutes without any soldering, cutting of tracks, etc. It comes complete with its own TRSDOS compatible double density operating system.

**DOUBLE-ZAP II - DOUBLE DENSITY PATCH FOR NEWDOS 80**

**ONLY \$53.00 plus \$1.00 p&p**

~~DISCONTINUED~~  
If you are using NEWDOS 80, then you also need DOUBLE-ZAP II on diskette. This program upgrades your NEWDOS 80 to double density with ADR (automatic density recognition). 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-ZapII 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!

**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.

**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.

**LEARNING LEVEL II**

**\$16.95 + \$1.20 p&p**

Written by Daniel Lien, the author of the TRS-80 Level I Handbook, this book teaches you, step-by-step, how to get the most from your Level II machine. Invaluable supplement to either the TRS-80 Level II Manual or the System-80 Manuals.

**HIGH QUALITY DISKETTES ALL PRICES INCLUDE P&P**

**HIGH QUAL**

**40 TRACK NASHUA SINGLE SIDE/SINGLE DENSITY ... \$45.00 box of 10**

**40 TRACK VERBATIM DOUBLE SIDE**

# MORE AUSTRALIAN SOFTWARE

All programs designed to run on both the TRS-80 or the SYSTEM 80 without modification. Most programs include sound

## TRIAD VOL 1 – L2/16K Cassette \$10.95 Disk \$15.95 + 60c p&p

Three separate games which test your powers of memory and concentration. The programs combine graphic displays and sound:

**SIMON-SEZ:** Just like the electronic music puzzles on sale for more than \$20. Numbers are flashed on the screen and sounded in a sequence determined by the computer. Your task is to reproduce the sequence, correctly.

**LINE?:** Rather like a super, complicated version of noughts and crosses. You may play against another player or against the computer itself. But beware, the computer cheats!

**SUPER CONCENTRATION:** Just like the card game but with more options. You must find the hidden pairs. You may play against other people, play against the computer, play on your own, or even let the '80 play on its own.

## TRIAD VOL 2 – L2/16K Cassette \$10.95 Disk \$15.95 + 60c p&p

Remember those "NUMERO" puzzles in which you had a matrix of numbers (or letters) with one blank space and you had to shuffle the numbers around one at a time until you had made a particular pattern? Well, **SHUFFLEBOARD**, the first program in this triad, is just this, except that the computer counts the number of moves you take to match the pattern it has generated – so it is not possible to cheat.

**MIMIC** is just like **SHUFFLEBOARD** except that you only see the computer's pattern for a brief span at the beginning of the game, then you must remember it!

In **MATCHEM**, you have to manoeuvre 20 pegs from the centre of the screen to their respective holes in the top or bottom rows. Your score is determined by the time taken to select a peg, the route taken from the centre of the screen to the hole and your ability to direct the peg into the hole without hitting any other peg or the boundary.

## VISURAMA L2/16K Cassette \$10.95 Disk \$15.95 + 60c p&p

Two programs which give fascinating, ever-changing patterns on the screen.

**LIFE** is the fastest implementation of the Game of Life you will see on your '80. Machine language routines create up to 1200 new generations per minute for small patterns or up to 100 per minute for the full 128 x 48 screen matrix. Features full horizontal and vertical wraparound.

**EPICYCLES** will fascinate you for hours. The ever-changing ever-moving patterns give a 3D effect and were inspired by the ancient Greek theories of Ptolemy and his model of the Solar system.

## EDUCATION AND FUN – L1/4K, L2/16K Cassette \$10.95 Disk \$15.95 + 60c p&p

Written by a primary school teacher to make learning enjoyable for his pupils, there are five programs in both Level I and Level II to suit all systems:

**BUG-A-LUG:** a mathematics game, in which you must get the sum correct before you can move.

**AUSTRALIAN GEOGRAPHY:** learn about Australian States and towns, etc.

**SUBTRACTION GAME:** build a tower with correct answers.

**HOW GOOD IS YOUR MATHS?** Select the function (+, -, ÷ or X) and degree of difficulty.

**HANGMAN:** That well known word game now on your computer.

*Recommended for children from 6 to 9 years.*

## COSMIC FIGHTER & SPACE JUNK – L2/16K Cassette \$10.95 Disk \$15.95 + 60c p&p

Both programs have sound to complement their excellent graphics. In **COSMIC FIGHTER**, you must defend the earth against seven different types of alien aircraft. It is unlikely that you will be successful but you will have a lot of fun trying!

Your mission in **SPACE JUNK** is to clean up all the debris left floating around in space by those other space games. It is not as simple as it sounds and space junk can be quite dangerous unless you are very careful.

## SPACE DRIVE L2/4K & 16K Cassette \$8.95 Disk \$13.95 + 60c p&p

Try to manoeuvre your space ship through the meteor storms then land it carefully at the space port without running out of fuel or crashing. Complete with realistic graphics.

## STARFIRE AND NOVA INVASION L2/16K Cassette \$10.95 Disk \$15.95 + 60c p&p

Both programs include sound to improve their realism.

**STARFIRE** seats you in the cockpit of an X-wing fighter as you engage in battle with the deadly Darth Vader's Tie-fighters. Beware of the evil one himself and may the Force be with you.

In **NOVA INVASION**, you must protect your home planet of Hiberna from the invading NOVADIANS. You have two fixed guns at each side of the screen and a moveable one at the bottom. Apart from shooting down as many invaders as possible, you must protect your precious hoard of Vitaminium or perish!

## AIR ATTACK AND NAG RACE – L2/16K Cassette \$10.95 Disk \$15.95 + 60c p&p

An unlikely combination of programs but they share the same author who has a keen sense of humour.

**AIR ATTACK** includes sound and realistic graphics. The aircraft even have rotating propellers! But they also drop bombs on you, so it's kill or be killed!

**NAG RACE** lets you pander to your gambling instinct without actually losing real money. Up to five punters can join in the fun. Each race results in a photo-finish whilst there is a visible race commentary at the bottom of the screen throughout the race. Happy punting!

## FOUR LETTER MASTERMIND L2/16K Cassette \$8.95 Disk \$13.95 + 60c p&p

There are 550 four-letter words from which the computer can make its choice. You have 12 chances to enter the correct word. After each try, the computer informs you of the number of correct letters and those in the correct position. You can peek at the list of possible words but it will cost you points. Makes learning to spell fun.

## MUSIC IV – L2/16K Cassette \$8.95 Disk \$13.95 + 60c p&p

Music IV is a music compiler for your '80. It allows you to compose or reproduce music with your computer that will surprise you with its range and quality. You have control over duration (full beat to 1/16 beat) with modifications to extend the duration by half or one third for triplets. Both sharps and flats are catered for as are rests. Notes on whole sections may be repeated. The program comes with sample data for a well-known tune to illustrate how it is done.

### \*\*\* SAVE 00\$'s \*\*\* SAVE 00\$'s \*\*\* SAVE 00\$'s \*\*\* MICRO-80 EXPANSION INTERFACE \*\*\*

MICRO-80's expansion interface utilises the proven LNW Research Expansion board. It is supplied fully built up and tested in an attractive cabinet with a self contained power supply, ready to plug in and go. The expansion interface carries MICRO-80's full, no hassle, 90-day warranty.

**Features include:-** ● Sockets for up to 32K of memory expansion ● Disk controller for up to 4 disk drives ● Parallel printer port ● Serial RS232C/20mA I/O port ● Second cassette (optional)

The expansion interface connects directly to your TRS-80 L2/16K keyboard or, via SYSPAND-80 to your SYSTEM-80/VIDEO GENIE  
Prices: HD-010-A Expansion Interfaces with Ø K : \$499.00 HD-010-B Expansion Interfaces with 32K : \$549.00 HD-011 Data separator fitted (recommended) : add \$29.00 HD-012 Dual cassette Interfaces fitted : add \$19.00

The MICRO-80 Expansion Interface is also available in kit form.

Prices: HD-013 Kit consisting of LNW Research PC board and manual, ALL components including cabinet & power supply : \$375.00 HD-011 Data separator for above \$25.00 HD-013 Dual cassette Interface kit : \$15.00

**TURN  
THIS**

**into  
this**

**for \$49.00** plus \$2.00 p & p

A choice of upper and lower case display is easier to read, gives greater versatility.

The Micro-80 lower case modification gives you this facility, plus the symbols for the 4 playing-card suits for \$49.00 + \$2.00 p. & p.

The Micro-80 modification features true below-the-line descenders and a block cursor.

Each kit comes with comprehensive fitting instructions and two universal lower-case drive routines on cassette to enable you to display lower case in BASIC programs.

The driver routines are self-relocating, self-protecting and will co-reside with other machine language programs such as Keyboard-debounce, serial interface driver programs etc.

Both programs give your TRS-80™ Model I or System 80™ an optional typewriter capability, i.e. shift for upper case.

The second programme also includes Keyboard-debounce and a flashing cursor.

You fit it. Or we can.

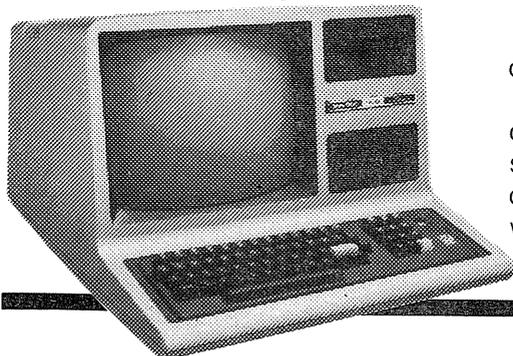
Fitting the modification requires soldering inside the computer. This should only be carried out by an experienced hobbyist or technician.

If you are at all dubious, a fitting service is available in all capital cities for only \$20.00.

A list of installers is included with each kit.

**Save \$120 now.**

**ADD A DISK DRIVE TO YOUR TRS-80™ MODEL III FOR ONLY \$875.00 OR ADD TWO FOR ONLY \$1199.**



The Micro-80 disk drive upgrade for the TRS-80™ Model III contains the following high quality components:

1 or 2 MPI 40-track single head disk drives, 1 VR Data double-density disk controller board and 1 dual drive power supply plus all the necessary mounting hardware, cables and comprehensive fitting instructions, which can be carried out with a minimum of fuss by any average computer owner.

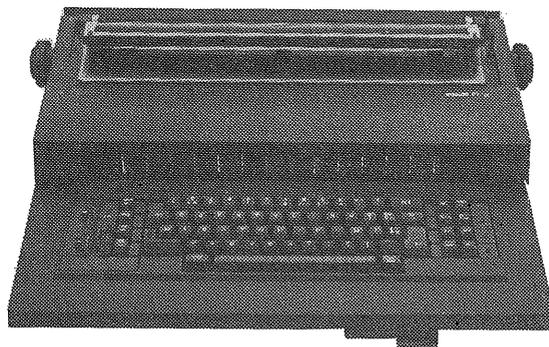
Fitting service is available for \$25.00 in most capital cities.

### **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.

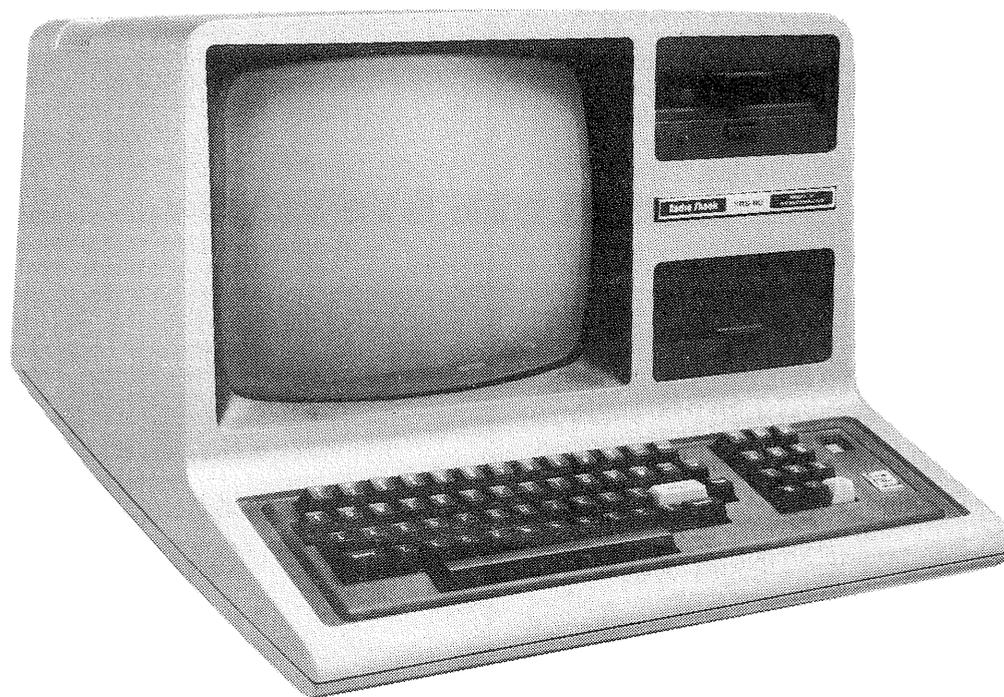
How good is it? - This part of our advertisement was typeset using an ET-121 driven by a TRS-80. Write and ask for full details.

**ONLY \$2049 INC. S.T.**



---

**1.4 MEGABYTES ON LINE + 48K RAM**  
**for \$3800** incl. Sales Tax



**MICRO-80's**  
**MODEL 380 +**

**MICRO-80** has equipped the TRS-80 with two high reliability dual-head 80 track mini-floppy disk drives made by MPI, one of America's leading mini-disk drive manufacturers.

This turns the mild-mannered Model 3 into a powerhouse able to handle the most difficult business programs. The TRS-80 is one of the best-supported microcomputers in the world. MICRO-80 has been supporting the TRS-80 in Australia for 18 months and is one of Australia's leading dealers in MPI disk drives.

**2.8 MEGABYTES FOR \$5300** incl. Sales Tax

If you need even more file space you can add MICRO-80's external dual-drive cabinet enclosing two more dual-head 80 track drives for an additional \$1500.

---

# COMPUTER PRICES

## MODEL 340

2 40 TRACK SINGLE HEAD DRIVES GIVING  
350K FORMATTED STORAGE, 48K RAM

\$2990 INCL. SALES TAX

## MODEL 340 +

2 40 TRACK DUAL-HEAD DRIVES GIVING  
700K FORMATTED STORAGE, 48K RAM

\$3350 INCL. SALES TAX

## MODEL 380

2 80 TRACK SINGLE HEAD DRIVES GIVING  
700K FORMATTED STORAGE, 48K RAM

\$3350 INCL. SALES TAX

## MODEL 380 +

2 80 TRACK DUAL-HEAD DRIVES GIVING  
1.4 MEGABYTE FORMATTED STORAGE, 48K RAM

\$3750 INCL. SALES TAX

# COMPLETE SYSTEMS

## 350K SYSTEM

MODEL 340, EPSON MX-80 PRINTER  
DOSPLUS DISK OPERATING SYSTEM

\$4040 INCL. SALES TAX

## 700K SYSTEM (40 Track)

MODEL 340 + , EPSON MX-80 PRINTER  
DOSPLUS DISK OPERATING SYSTEM

\$4400 INCL. SALES TAX

## 700K SYSTEM (80 Track)

MODEL 380, EPSON MX-80 PRINTER  
DOSPLUS DISK OPERATING SYSTEM

\$4400 INCL. SALES TAX

## 1.4 MEGABYTE SYSTEM

MODEL 380 + , EPSON MX-80 PRINTER  
DOSPLUS DISK OPERATING SYSTEM

\$4850 INCL. SALES TAX

## 2.8 MEGABYTE SYSTEM

MODEL 380 + , DUAL EXTERNAL DRIVES,  
MX-80 PRINTER, DOSPLUS DISK OPERATING SYSTEM

\$6350 INCL. SALES TAX

# 40 TRACK COMPATIBILITY

**MAKE 80** — PROGRAM TO CONVERT  
40 TRACK DISKS TO 80 TRACK

\$19.95

**COMPAT-80** — HARDWARE DEVICE TO SWITCH  
80 TRACK DRIVES TO 40 TRACK. GIVES FULL  
BI-DIRECTIONAL COMPATIBILITY  
(AVAILABLE SEPTEMBER '81)

\$65.00

All prices subject to change without notice. Prices are F.O.B. Adelaide. All computers and systems carry MICRO-80's  
90-day Warranty covering parts and labour.



# HIGH QUALITY DISKETTES

**40 Track Scotch Brand Single Side / Single Density** ..... \$59 box of 10  
**40 Track Verbatim Double Side / Double Density** ..... \$59 box of 10  
**77 Track Verbatim Single Side / Single Density** ..... \$59 box of 10



## 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)

\$14.95 \$ 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) \$8.95 + 60c 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) \$8.95 + 60c 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)

\$10.95 + 60c 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.

7 DISKETTES

DOUBLE DENSITY ... \$59.00 box of 10

HIGH QUALITY DISKETTES

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

**AUSTRALIAN SOFTWARE (Cont.)****GAMES****SHEEPDOG (L2/16K) \$8.95 + 60c p&p**

Ever wondered how a sheepdog manages to drive all those awkward sheep into a pen? Well, here is your chance to find out just how difficult it is and have a lot of fun at the same time. You control the sheepdog, the computer controls the sheep! As if that isn't enough, look out for the dingoes lurking in the bush!

**U BOAT \$8.95 + 60c 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 WITH SOUND \$8.95 + 60c 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) \$8.95 + 60c p&p**

Pit your skills of 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) \$8.95 + 60c 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) \$8.95 + 60c 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 RXR III loading board, if fitted.

**LOWER CASE FOR YOUR TRS-80/SYSTEM 80****Kit only \$49.00 plus \$2.00 p&p**

Give your TRS-80 or SYSTEM 80 a lower case display with proper descenders and a block cursor (similar to the TRS-80 Model III). Also includes symbols for the four suits of cards. Includes full fitting instructions, all necessary components and a special machine language driver program to enable lower case in BASIC. The modification is similar to the Tandy model 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 install 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 are also arranging to have installers in each State. See elsewhere in this issue for their names and addresses.

**PRICES**

Cat No.

HD-020 Lower case mod kit for TRS-80

\$49.00 plus \$2.00 p&amp;p

HD-021 Lower case mod kit for SYSTEM-80

\$49.00 plus \$2.00 p&amp;p

**EPSON MX-80 PRINTER****ONLY \*\$949 Inc. Cable for TRS-80 and p&p****(\*Printer only — \$940 incl. p&p)**

The EPSON MX-80 printer is compact, quiet, has features unheard of only 2-3 years ago in a printer at any price and, above all, is ultra-reliable. All available print modes may be selected under software control. Features include:

- high quality 9x9 dot-matrix character formation
- 3 character densities
  - . 80 characters per line at 10 chars/inch
  - . 132 characters per line at 16.5 chars/inch
  - . 40 characters per line at 5 chars/inch
- 2 line spacings
  - . 6 lines per inch      8 lines per inch
- 80 characters per second print speed
- bi-directional printing
- logical seeking of shortest path for printing
- lower case with descenders
- TRS-80 graphics characters built in
- standard Centronics printer port

The bi-directional printing coupled with the logical seeking of the shortest print path (which means that the print head will commence printing the next line from the end which requires the least travel, thereby minimising unutilised time) gives this printer a much higher throughput rate than many other printers quoting print speeds of 120 c.p.s. or even higher.

**GREEN SCREEN SIMULATOR****\$9.50 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.

```

310 PRINT:PRINT"--- THAT'S ABOUT ";H;" MILES PER GALLON."
320 PRINT:PRINT"--- THAT IS ";FF;"LITRES PER 100 KM."
330 PRINT@716,"--- WANT TO SAVE IT ON TAPE? <Y/N> --- ";
340 GOSUB 440
350 CLS
360 PRINT@274,"MAKE PREPARATIONS FOR DUMPING"
370 PRINT@470,"";:INPUT"HIT ENTER WHEN READY";A
380 CLS:PRINT@470,"DUMPING DATA"
390 FORX=1 TO 1000:OUT255,4:NEXT:PRINT#-1,M,CC,A,B,F,D,E,H,C,FF
400 CLS:PRINT@470,"DATA HAS BEEN DUMPED"
410 CLS:PRINT@470,"--- MAKE ANOTHER COPY <Y/N> ---";
420 GOSUB 440
430 GOTO 390
440 C$="":C$=INKEY$:IF C$="" THEN 440
450 IF C$="Y" THEN RETURN ELSE IF C$="N" THEN CLS
460 CLS:PRINT@470,"--- ANY MORE DATA? <Y/N> ---";
470 G$="":G$=INKEY$:IF G$="" THEN 470
480 IF G$="Y" THEN 70
490 IF G$="N" THEN CLS:GOTO 500
500 CLS:PRINT@24,"M E N U "
510 PRINT@24,"SELECT 1"
520 PRINT@207,"1 - INPUT DATA ROUTINE"
530 PRINTTAB(15);"2 - PROJECTION CHART"
540 PRINTTAB(15);"3 - SUMMARY"
550 PRINTTAB(15);"4 - END"
560 A$="":A$=INKEY$:IF A$="" THEN 560
570 B=VAL(A$):ON B GOTO 70 ,580 ,680 ,1070
580 CLS:PRINT@24,"PROJECTION CHART"
590 PRINT:PRINT"ON THE AVERAGE HOW MANY KM/LITRE DOES YOUR CAR D
O ?";
600 INPUT P
610 PRINT"HOW MANY KM DO YOU WANT IT PROJECTED FOR ?";
620 INPUT PP
630 XX=PP/P
640 PRINT:PRINT"ON ";PP;"KM YOU WOULD NEED ";XX;" LITRES OF FUEL
"
650 PRINT@906," PRESS ANY KEY TO RETURN TO MENU"
660 D$=INKEY$:IF D$="" THEN 660 ELSE 670
670 GOTO 500
680 CLS:PRINT@24,"SUMMARY"
690 PRINT:PRINT:PRINT
700 PRINT@270,"MONTHLY OR YEARLY --- M OR Y"
710 E$="":E$=INKEY$:IF E$="" THEN 710
720 IF E$="M"THEN740 ELSE850
730 STOP
740 CLS:PRINT@24,"MONTHLY"
750 PRINT:PRINT:PRINT:INPUT"MONTH NUMBER = ?";MM
760 M=0
770 CLS:PRINT@470,"READING -- MONTHLY #";M+1
780 INPUT #-1,M,CC,A,B,F,D,E,H,C,FF
790 IF M=MM THEN800 ELSE IF M<>MM THEN 770
800 CLS:PRINT@24,"MONTH NUMBER";M:PRINT:PRINT:PRINT
810 GOSUB1080
820 CLS:PRINT@470,"--- ANY MORE MONTHS ? <Y/N> ---"
830 GOSUB 440
840 GOTO 740
850 CLS:PRINT@24,"YEARLY"
860 PRINT@275,"MAKE PREPARATIONS"
870 A=0:PRINT:INPUT" PRESS ENTER WHEN READY";A
880 CLS:PRINT@470,"READING -- YEARLY"
890 M=0
900 PRINT@538,"MONTH #";M+1
910 INPUT #-1,M,CC,A,B,F,D,E,H,C,FF
920 Z=Z+CC
930 Y=Y+F
940 QQ=QQ+C
950 RR=RR+E
960 IF M=12 THEN 970 ELSE 900
970 CLS:PRINT@24,"COSTS"
980 PRINT:PRINT
990 PRINT,"TOTAL LITRES","TOTAL COST"
1000 PRINT,Y,Z
1010 PRINT"TOTAL KMS",,, "AVG KM/L"
1020 R=RR/12
1030 PRINTQQ,,R
1040 PRINT:PRINT:PRINT" PRESS ANY KEY TO RETURN TO
MENU"

```

```

1050 GOSUB 470
1060 RETURN
1070 CLS:PRINT@470,"--- THE END ---":END
1080 PRINT,"TOTAL LITRES","TOTAL COST"
1090 PRINT,F,CC
1100 PRINT"TOTAL KMS",,,"AVG KM/L"
1110 PRINTC,,E
1120 PRINT@910,"PRESS ANY KEY TO RETURN TO MENU"
1130 GOSUB 470

```

\*\*\*\*\* CONVERSIONS LII/4K-16K (C) S. Greco \*\*\*\*\*

This program gives the user the choice of four different types of conversion :-

TEMPERATURE.

Fahrenheit to centigrade and centigrade to Fahrenheit.

LENGTH.

Feet to centimetres and centimetres to feet.

Inches to centimetres and centimetres to inches.

DISTANCE.

Kilometres to miles and miles to kilometres.

WEIGHT.

Stones to kilogrammes and kilogrammes to stones.

The menu display is controlled by the INKEY\$ function, so that it is not necessary to press the ENTER/NEWLINE key unless you are typing in more than one digit, like temperature.

To use the program simply select the type of conversion that you want performed from the menu and then enter the values that you want converted.

TO RUN THIS PROGRAM IN A 4K SYSTEM.

Delete all of the instructions and remove all REM lines.

```

10 '          *** DEGREE CONVERSIONS ***
           *   SILVIO GRECO   *
           * RIVERVIEW COURT *
           * MARIBYRNONG VIC. *
           *****

20 CLS
30 GOSUB 1170
40 REM*** COPYRIGHT(C) 1980
50 CLEAR200
60 CLS:PRINT@284,"MENU"
70 PRINT@348,STRING$(4,"*");
80 PRINT@465,"1. TEMPERATURE (F,C)
           2. DISTANCE (M,K)
           3. LENGTH (CM, IN + CM, FEET)
           4. WEIGHT (S,KG)"
90 Q$=INKEY$:IF Q$=""THEN 90 ELSE 100
100 Q=VAL(Q$):ON Q GOTO110 ,330 ,520 ,970 :IF Q>4THEN90
110 CLS
120 PRINT @200,"FOR FARENEIGHT TO CENTIGRADE CONVERSION PRESS 'F
' "
130 PRINT@264,STRING$(49,"-");
140 PRINT @456,"FOR CENTIGRADE TO FARENEIGHT CONVERSION PRESS 'C
' "
150 PRINT@520,STRING$(49,"-");
160 A$=INKEY$:IF A$="F" THEN240 ELSE IF A$="C" THEN 180
170 GOTO 160
180 CLS
190 PRINT@212,"INPUT CENTIGRADE VALUE"
200 PRINT:PRINT TAB(26);:INPUT C
210 F=(9*C)/5+32

```

```

220 PRINT@461, C; "CENTIGRADE ="; F; "FARENEIGHT"
230 GOTO 290
240 CLS
250 PRINT @212, "INPUT FARENEIGHT VALUE"
260 PRINT:PRINT TAB(29);:INPUT F
270 C=(F-32)*5/9
280 PRINT@461, F; "FARENEIGHT ="; C; "CENTIGRADE"
290 FOR X=1TO 1000 :NEXT X
300 PRINT@850, "ANY MORE CONVERSIONS <Y/N>"
310 S#=INKEY$: IF S#="Y" THEN 110 ELSE IF S#="N" 80
320 GOTO 310
330 CLS:PRINT@208, "FOR MILES TO KILOMETRES PRESS 'M'"
340 PRINT@272, STRING$(32, "-");
350 PRINT@464, "FOR KILOMETRES TO MILES PRESS 'K'"
360 PRINT@528, STRING$(32, "-");
370 Q#=INKEY$: IF Q#="M" THEN 470 ELSE IF Q#="K" THEN 390
380 GOTO 370
390 CLS:PRINT@212, "INPUT KILOMETRE VALUE"
400 PRINT :PRINT TAB(28);:INPUT K
410 M=K*.621
420 PRINT@466, K; "KILOMETRES ="; M; "MILES"
430 FOR X=1TO1000:NEXT
440 PRINT@850, "ANY MORE CONVERSIONS <Y/N>"
450 Q#=INKEY$: IF Q#="Y" THEN 330 ELSE IF Q#="N" THEN 60
460 GOTO 450
470 CLS:PRINT@214, "INPUT MILE VALUE"
480 PRINT:PRINT TAB(28);:INPUT M
490 K=M*1.61
500 PRINT@463, M; "MILES ="; K; "KILOMETRES"
510 GOTO430
520 CLS:PRINT@200, "FOR FEET AND CENTIMETRE CONVERSIONS PRESS 'T'"
530 PRINT@264, STRING$(45, "-");
540 PRINT@456, "FOR CENTIMETRE AND INCH CONVERSION PRESS 'S'"
550 PRINT@520, STRING$(44, "-");
560 Q#=INKEY$: IF Q#="T" THEN 770 ELSE IF Q#="S" THEN 580
570 GOTO 560
580 CLS:PRINT@208, "FOR CENTIMETRE TO INCHES PRESS 'C'"
590 PRINT@272, STRING$(34, "-");
600 PRINT@464, "FOR INCHES TO CENTIMETRES PRESS 'I'"
610 PRINT@528, STRING$(34, "-");
620 Q#=INKEY$: IF Q#="C" THEN 720 ELSE IF Q#="I" THEN 640
630 GOTO 620
640 CLS:PRINT@212, "INPUT INCH VALUE"
650 PRINT:PRINT TAB(26);:INPUT I
660 C=I*2.54
670 PRINT @462, I; "INCHES ="; C; "CENTIMETRES"
680 FOR X=1TO1000: NEXT
690 PRINT@ 848, "ANY MORE CONVERSIONS <Y/N>"
700 Q#=INKEY$: IF Q#="Y" THEN 520 ELSE IF Q#="N" THEN 60
710 GOTO 700
720 CLS:PRINT@212, "INPUT CENTIMETRE VALUE"
730 PRINT:PRINT TAB(26);:INPUT C
740 I=C*.394
750 PRINT @462, C; "CENTIMETRES ="; I; "INCHES"
760 GOTO 690
770 CLS:PRINT@208, "FOR FEET TO CENTIMETRES PRESS 'F'"
780 PRINT@272, STRING$(33, "-");
790 PRINT@464, "FOR CENTIMETRES TO FEET PRESS 'M'"
800 PRINT@528, STRING$(33, "-");
810 Q#=INKEY$: IF Q#="F" THEN 920 ELSE IF Q#="M" THEN 830
820 GOTO 810
830 CLS
840 PRINT@212, "INPUT CENTIMETRES VALUE"
850 PRINT TAB(26);:INPUT M
860 F=(M*.394)/12
870 PRINT@ 466, M; "CENTIMETRES ="; F; "FEET"
880 FOR X=1TO1000:NEXT
890 PRINT@850, "ANY MORE CONVERSIONS <Y/N>"
900 Q#=INKEY$: IF Q#="Y" THEN 770 ELSE IF Q#="N" THEN 60
910 GOTO900
920 CLS:PRINT@214, "INPUT FEET VALUE"
930 PRINT TAB(26);:INPUT F
940 M=F*30.5
950 PRINT@ 466, F; "FEET ="; M; "CENTIMETRES"
960 GOTO880

```

```

970 CLS:PRINT@208,"FOR STONES TO KILOGRAMES PRESS 'S'"
980 PRINT@272,STRING$(33,"-");
990 PRINT@400,"FOR KILOGRAMES TO STONES PRESS 'K'"
1000 PRINT@464,STRING$(33,"-");
1010 Q$=INKEY$:IF Q$="S" THEN 1030 ELSE IF Q$="K" THEN 1110
1020 GOTO 1010
1030 CLS:PRINT@214,"INPUT STONES VALUE"
1040 PRINT TAB(26);:INPUT S
1050 K=S*.35
1060 PRINT@465,S;"STONES =";K;"KILOGRAMES"
1070 FOR X=1TO 1000:NEXT
1080 PRINT@850,"ANY MORE CONVERSIONS <Y/N>"
1090 Q$=INKEY$:IF Q$="Y" THEN 970 ELSE IF Q$="N" THEN 60
1100 GOTO1090
1110 CLS:PRINT@212,"INPUT KILOGRAMES VALUE"
1120 PRINTTAB(26);:INPUT K
1130 S=K*.157
1140 PRINT@464,K;"KILOGRAMES";S;"STONES"
1150 GOTO 1070
1160 END
1170 PRINT@153,"INSTRUCTIONS
*****"
1180 PRINT@322,"THIS PROGRAM WILL ENABLE YOU TO MAKE 4 DIFFERENT
CONVERSIONS."
1190 PRINT@386,"THE MENU SHOWS YOU THE DIFFERENT CONVERSIONS, IF
YOU WANT A "
1200 PRINT@450,"PARTICULAR CONVERSION, THEN YOU PRESS THE CORRIS
PONDING"
1210 PRINT@514,"NUMBER. THE LETTERS BE SIDE THE CONVERSIONS ARE
TO INDECAT E "
1220 PRINT@578,"WHAT TYPE OF CONVERSION, EG. (F,C) MEANS FARENHEI
GHT AND
CENTIGRADE."
1230 FOR X=1TO 1000:NEXT:PRINT@914,"PRESS SPACE BAR TO CONTINUE"
1240 Q$=INKEY$:IF Q$="" THEN 1260
1250 GOTO 1240
1260 GOTO 50

```

\*\*\*\*\* STARSHOOT LII/4K (C) D. Zwart \*\*\*\*\*

The object of this game is to shoot stars numbered from 1 to 9. The stars are numbered in this fashion :-

```

1      2      3
4      5      6
7      8      9

```

although these numbers are not displayed on the screen. When the game starts there is one star on the screen. You must shoot stars so as to get from this :-

```

"      "      "
"      X      "
"      "      "

```

to this :-

```

X      X      X
X      "      X
X      X      X

```

To shoot a star, type in a number corresponding to the star's position. As stars are shot different patterns of stars will appear. The appearance of any particular new star is determined by the star you have just shot and its neighbours.

During the progress of the game the instructions can be recalled by typing "I", or a display that illustrates the effect that shooting a particular star will have on the other stars can be called up by pressing any other key. System-80 owners, when typing in the program listing, can omit all references to CHR\$(23).

```

10 '          ***** STARSHOOT *****
              *   D. ZWART   *
              * PROOF RANGE *
              * PT. WAKEFIELD *
              * S.A. 5550 *
              *****

20 DIMB$(16)
30 FORA=1TO9:B$(A)=" ":NEXTA:B$(5)="X"
40 GOTO90
50 FORQ=1TO2000:NEXT:RUN
60 U=U-1:PRINT"PRESS ANY KEY WHEN READY"
70 A$=INKEY$:IFA$=""THEN70
80 U=U+1:Z=VAL(A$):RETURN
90 CLS:PRINTCHR$(23):PRINT@146,"# STARSHOOT #"
100 PRINT@210,STRING$(13,"*")
110 PRINT@576,"DO YOU NEED INSTRUCTIONS? (Y/N)"
120 GOSUB70:U=U-1
130 IFA$="N"THEN250
140 PRINT"
THE GAME STARTS WITH
      . . .
      . X .
      . . ."

150 PRINT"
AND YOU MUST FINISH WITH
      X X X
      X . X
      X X X"

160 PRINT"
BY SHOOTING AT STARS NUMBERED
      1 2 3
      4 5 6
      7 8 9"

170 GOSUB60
180 PRINT"
THE STAR YOU HIT WILL EXPLODE
AND AFFECT THE AREA AROUND IT.
WHERE THERE IS A STAR IT WILL
DISAPPEAR AND WHERE THERE IS NO STAR A NEW ONE WILL APPEAR."
190 PRINT"THE FOLLOWING CHART, WHICH CAN
BE RECALLED DURING THE GAME WITH ANY UNUSED KEY, IS DIVIDED INTO
NINE PARTS. EACH PART SHOWS WHAT AREA IS AFFECTED BY WHICH STAR.
(THESE INSTRUCTIONS MAY BE RE-
CALLED WITH AN 'I')
"
200 GOSUB60
210 PRINT"
  1 2 .      1 2 3      . 2 3
1 4 5 .      2 . . .      3 . 5 6      . . .      . . .      . . . ."
220 PRINT"
  1 . .      . 2 .      . . 3 4 4 . .      5 4 5 6      6 . . 6
  7 . .      . 8 .      . . 9"
230 PRINT"
. . .      . . .      . . . 7 4 5 .      8 . . .      9 . 5 6
7 8 .      7 8 9      . 8 9
"
240 GOSUB60
250 CLS:PRINTCHR$(23)
260 FORA=1TO9STEP3
270 PRINT:PRINT:PRINT"          "B$(A)" " "B$(A+1)" " "B$(A+2)
)
280 NEXT:PRINT:IFB$(5)="X"THEN330
290 B$(5)="X":FORT=1TO16:IFB$(T)=" ":THENB$(5)=" ":GOTO310
300 NEXT:PRINT"YOU TOOK ";U;" TURNS":GOTO50
310 FORP=1TO16:IFB$(P)="X"THEN330
320 NEXT:PRINT"NO STARS LEFT YOU LOSE":GOTO50
330 PRINT@10,"SHOOTING AT WHICH STAR?"
340 GOSUB70
350 IFZ<1ORZ>9THENU=U-1ELSE370
360 IFA$="I"THENGOTO140ELSE210
370 IFB$(Z)=" ":U=U-1:PRINT"NO STAR THERE TRY AGAIN":GOTO340
380 ONZGOTO390,400,410,420,430,440,450,460,470
390 A=1:GOSUB480:A=2:GOSUB480:A=4:GOSUB480:A=5:GOSUB480:GOTO250

```

```

400 A=1:GOSUB480:A=2:GOSUB480:A=3:GOSUB480:GOTO250
410 A=2:GOSUB480:A=3:GOSUB480:A=5:GOSUB480:A=6:GOSUB480:GOTO250
420 A=1:GOSUB480:A=4:GOSUB480:A=7:GOSUB480:GOTO250
430 A=2:GOSUB480:A=4:GOSUB480:A=5:GOSUB480:A=6:GOSUB480:A=8:GOSU
B480:GOTO250
440 A=3:GOSUB480:A=6:GOSUB480:A=9:GOSUB480:GOTO250
450 A=4:GOSUB480:A=5:GOSUB480:A=7:GOSUB480:A=8:GOSUB480:GOTO250
460 A=7:GOSUB480:A=8:GOSUB480:A=9:GOSUB480:GOTO250
470 A=5:GOSUB480:A=6:GOSUB480:A=8:GOSUB480:A=9:GOSUB480:GOTO250
480 IFB$(A)=". "THENB$(A)="X"ELSE:B$(A)=". "
490 RETURN

```

\*\*\*\*\* BINGO LII/4K (C) Geoff Egel \*\*\*\*\*

This program first saw the light of day when a service club approached the author to see if they could use a computer to raise some finance for a community project. In that capacity, the program was set up as a bingo caller, simply displaying the drawn number on the screen. This version has been slightly modified to enable two players to select seven numbers different from each other. The program then "calls out" the numbers using standard bingo calls. The first player to have all his numbers called out is the winner. Numbers selected should be in the range from 1 to 90. The original function of the program as a bingo caller has been retained and may be selected from the menu displayed at the start of the program.

```

10 * ***** BINGO *****
*      GEOFFREY EGEL      *
*      18 STURT ST.      *
*      LOXTON 5333      *
* ***** *****
20 CLS: CLEAR 1000: RANDOM: PRINT TAB(25) "BINGO": PRINT STRING$(63,
191): T$="/": DIM A(180)
30 PRINT@128, "THIS IS A GAME OF PURE CHANCE FOR TWO PLAYERS WHO
MUST ENTER THEIR NAME AND SEVEN NUMBERS BETWEEN 1- 90 THE PLA
YERS WHOS NUMBERS COME UP FIRST IS THE WINNER OR THIS PROGRAM
CAN BE USED FOR ACTUAL BINGO CALLING PRESS ENTER TO CONTINUE";
40 INPUT A$
50 T=0: INPUT " (1) FUN RAISER (2) GAME "; T: IF (T=0) OR (T>2) THEN
50
60 CLS: M=1: J=0
70 FOR F=1 TO 90
80 A(F)=F: D=F+90: A(D)=0
90 NEXT F: IF T=2 THEN GOTO 310
100 IF J=90 THEN 250
110 IF M=0 THEN 130
120 CLS: PRINT TAB(25) " BINGO": PRINT STRING$(63,140)
130 M=0: C=RND(90)
140 IF A(C)=0 THEN 130
150 D=C+90: A(D)=A(C): A(C)=0: J=J+1
160 IF T=2 THEN GOSUB 390
170 PRINT@512, STRING$(63,140)
180 FOR Z=1 TO C
190 READ B$
200 NEXT Z
210 PRINT@ 576, TAB(15); C; " * "; B$; " * " ; C; TAB(58) J: RESTO
RE ; PRINT@640, STRING$(63,140)
220 IF T=2 THEN GOTO 430
230 PRINT: PRINT "OPTIONS (1) NEXT NUMBER (2) CHECK (3) NEW GAME
"
240 A$=INKEY$: IF A$="" THEN 240: ELSE K=VAL(A$): IF (K=0) OR (K>3) T
HEN 240: ELSE ON K GOTO 100,260,60
250 PRINT " ALL NUMBERS HAVE BEEN CALLED "
260 FOR S=91 TO 180
270 IF A(S)=0 THEN 290
280 PRINT A(S);
290 NEXT S: M=1
300 GOTO 230
310 INPUT "FIRST NAME"; C$
320 FOR N=1 TO 7: C(N)=0
330 PRINT "NUMBER"; N; : INPUT C(N): IF (C(N)=0) OR (C(N)>90) THEN 330
340 NEXT N

```

```

350 CLS: INPUT "SECOND NAME"; D#
360 FOR N=1 TO 7: B(N)=0
370 PRINT "NUMBER"; N;: INPUT B(N): IF (B(N)=0) OR (B(N)>90) THEN 370
380 NEXT N: GOTO 100
390 FOR N=1 TO 7
400 IF C(N)=C THEN C(N)=0
410 IF B(N)=C THEN B(N)=0
420 NEXT N: RETURN
430 PRINT @320, TAB(63): PRINT @320, C#: TAB(15)
440 L=7: FOR N=1 TO 7: IF C(N)=0 THEN L=L-1: GOTO 460
450 PRINT C(N);
460 NEXT N: IF L=0 THEN PRINT TAB(47) "WINNER": GOTO 520
470 PRINT @ 384, TAB(63): PRINT @384, D#: TAB(15)
480 L=7: FOR N=1 TO 7: IF B(N)=0 THEN L=L-1: GOTO 500
490 PRINT B(N);
500 NEXT N: IF L=0 THEN PRINT TAB(47) "WINNER": GOTO 520
510 FOR Q=1 TO 1500: NEXT Q: GOTO 130
520 GOTO 520
530 DATA KELLY EYE, ONE LITTLE DUCK, UP A TREE, OFF THE FLOOR, LOOK
  ALIVE
540 DATA IN A FIX, YOU'RE IN HEAVEN, 1 FAT LADY, DOCTORS ORDERS, DOWNING
  ST, LEGS
550 DATA ONE DOZ, UNLUCKY FOR SOME, 1&4, 1&5, SWEET
560 DATA OFTEN BEEN KISSED, AGE TO VOTE, 1&9, BLIND, KEY TO THE DOOR
570 DATA 2 LITTLE DUCKS, 2&3, 2 DOZ, QUARTER OF A DOLLAR, HALF A CROWN
580 DATA 2&7, 2&8, 2&9, BLIND, 3&1, 3&2, FEATHERS, 3&4, 3&5, 3&6, 3&7, 3&8
590 DATA ALL THE STEPS, BLIND, 4&1, 4&2, 4&3, DIANA DORS, HALFWAY THERE,
  4&6
600 DATA 4&7, 4&8, 4&9, GOLDEN YEARS, 5&1, 5&2, 5&3, 5&4, ALL THE FIVE
  S, 5&6
610 DATA ALL THE VARIETIES, 5&8, BRIGHTON LINE, BLIND, 6&1, 6&2, 6&3,
  6&4
620 DATA AGE TO RETIRE, CLICKITY CLICK, 6&7, 6&8, EITHER WAY, BLIND,
  7&1
630 DATA 7&2, 7&3, 7&4, 7&5, TROMBONES, SUNSET STRIP, 7&8, 7&9, BLIND,
  8&1, 8&2
640 DATA 8&3, 8&4, 8&5, 8&6, FAT LADY WITH WALKING STICK, TWO FAT LADIES
650 DATA ALL BUT, TOP OFF THE WASA

```

---

\*\*\*\*\* GENIUS LII/16K

(C) T. Fraser \*\*\*\*\*

How many times, when you have visitors, do you switch on your computer and load in a mind bending program to show how smart your computer is. Then when they try the program you are given the ego destroying comment "that was simple". Well here is a program specially for your genius friends. They have a choice of three different games. The first game WORD BLOCK has four levels of play. A block of squares is drawn on the screen, the size of the blocks depends on the level of play selected. Each square is filled with a letter. The player's object is to rearrange the letters to form a word or sentence as the case may be. The letters are moved by using the up and down arrows, (ESC and CTRL on a System 80) and the ">" and "<" keys without shift. When one of these keys is pressed, the program checks to see if the square to which you are trying to move is empty and, if it is, the letter is moved to that square. There is a time limit in this game and during the game the word will be printed one letter at a time at the top of the screen. But for each letter printed you lose points from your score.

The second game is similar to the word block program, the difference being you don't have to use the arrow keys for movement. The game prints the numbers 1 to 9 in three columns of three. You have to end up with all the numbers in one column, (any column), going in order from 1 on the bottom to 9 on the top. The part that makes it difficult is that you are not allowed to put a smaller number on top of a bigger number as you move them around.

The third game is the old number guessing game with a twist in it for all the smarties. A random number between 1 and 100 is selected by the computer, you are then given a clue as to what the number is and from this clue you enter your first guess. After the first clue you should only have a few numbers which will fit the clue. From then on you have to be careful; at no time are you told whether your number is higher or lower than the answer. At the start of the game you are asked to enter the allowable number of tries. After each try, the number is reduced by one.

Typical screen displays are shown below.



```

150 PRINT@0, "I'M WORKING OUT THE WORD":GOTO220
160 FORX=29T099:SET(X,13):SET(X,19):SET(X,25):SET(X,31):SET(X,37
):SET(X,43):NEXT:FORY=13T043:SET(29,Y):SET(43,Y):SET(57,Y):SET(71
,Y):SET(85,Y):SET(99,Y):NEXT
170 FORX=203T0245:PRINT@X,"*":PRINT@X+768,"*":NEXT:FORX=203T09
71STEP64:PRINT@X,"*":PRINT@X+42,"*":NEXT:BA=300:BB=73:HB=240
180 POKE16238,63:X=RND(24):IFV=4THENX=25
190 FORL=1T0125+X:READZ(L):ZT=ZN(L):NEXT:RESTORE:E=1:G=24
200 IFV=3THENZB$="SENTENCE"ELSEIFV=4THENZB$="LETTERS"
210 PRINT@0, "I'M WORKING OUT THE ";ZB$
220 FORK=1T06:ZQ=MID$(ZT,K,1):ZV(K)=ZQ:NEXT
230 FORR=1T06-1:FORS=R+1T06
240 IFZV(R)<=ZV(S)THEN260
250 ZG=ZV(R):ZV(R)=ZV(S):ZV(S)=ZG
260 NEXTS
270 NEXTR
280 FORD=1T06
290 FORL=1T06:READF(L):H=F(L):NEXT:RESTORE
300 ZY=ZV(D):U=ASC(ZY):POKEH,U:E=E+1:NEXT
310 ONV60T0320,330,340,340
320 M=16103:POKEM,32:GOTO350
330 M=16231:POKEM,32:GOTO350
340 M=16238:POKEM,32:GOTO350
350 PRINT@0, "LETTERS SET"
360 IFBB<2THEN490ELSEB$=""
370 SX=G*10+200:FORTL=1T0SX:B$=INKEY$:IFB$=","THEN380ELSEIFB$="
"THEN390ELSEIFB$=CHR$(10)THEN400ELSEIFB$=CHR$(91)THEN410ELSEIFB$=
"C"THEN500ELSEIFB$="@"THENCLS:GOTO500ELSENEXT:GOSUB460:IFTT>6-1TH
EN490ELSE370
380 GOSUB420:C=PEEK(M+7):IFC=42THEN360ELSEPOKEM+7,32:POKEM,C:M=M
+7:GOTO360
390 GOSUB420:C=PEEK(M-7):IFC=42THEN360ELSEPOKEM-7,32:POKEM,C:M=M
-7:GOTO360
400 GOSUB420:C=PEEK(M-128):IFC=42THEN360ELSEPOKEM-128,32:POKEM,C
:M=M-128:GOTO360
410 GOSUB420:C=PEEK(M+128):IFC=42THEN360ELSEPOKEM+128,32:POKEM,C
:M=M+128:GOTO360
420 IFDD=1THEN440
430 CB=CB+1:IFCB=BATHEN450ELSERETURN
440 CB=CB+1:IFCB=BBTHEN450ELSERETURN
450 CB=0:DD=1:BB=BB-3:IFBB<2THENHB=HB-10:RETURNELSE460
460 FOREA=1T0FA:ZZ=MID$(ZT,EA,1):NEXT
470 PRINT@FA+25,ZZ:FA=FA+1:HB=HB-10
480 TT=TT+1:RETURNTT
490 PRINT@0, "YOUR TIME IS UP":GOTO550
500 HB=0:PRINTCHR$(23):PRINT@0, "YOU HAVE GIVEN UP":GOTO520
510 IFHB=0THENHB=10
520 PRINT:PRINT"THE CORRECT ANSWER IS ":PRINT:PRINTZT:PRINT:PRI
NT"YOUR SCORE IS ";HB
530 FORX=1T05000:NEXT:CLS:INPUT"DO YOU WANT TO PLAY THIS GAME AG
AIN (Y/N)";A$:IFA$="Y"THEN70ELSERUN
540 END
550 PRINT@64, "I'M CHECKING YOUR ANSWER":FORJA=1T06:READJB(JA):N
EXT:FORJC=1T06-1:FORJD=JC+1T06
560 IFJB(JC)<=JB(JD)THEN580
570 JE=JB(JC):JB(JC)=JB(JD):JB(JD)=JE
580 NEXT:NEXT:FORJA=1T06:JF=JB(JA):JG=PEEK(JF):ZW=CHR$(JG):ZU=ZU
+ZW:NEXT
590 CLS:PRINTCHR$(23):IFZU=ZTTHENPRINT@0, "YOU ARE CORRECT":GOTO5
10ELSEHB=0:PRINT@0, "SORRY YOU ARE WRONG YOU HAVE
"ZU:GOTO520
600 DATA15833,15840,15847,15961,15968,15975,16089,16096,16103,15
826,15954,16082,16210,16217,16224,16231,15698,15705,15712,15719,1
5726,15854,15982,16110,16238
610 DATAFARMYARD,SHEPHERD,LANGUAGE,PLEASANT,BENEFITS,SWIMSUIT,AD
DITION,CASSETTE,EXAMINED,FUNCTION,ANYTHING,MOUNTAIN,MATERIAL,POSS
IBLE,ASSEMBLE,RECORDER,FOLLOWED,STANDARD,GREETING,TAXATION,IMAGIN
ED,APPROACH,COMPLETE,PROMPTED,ELEPHANT
620 DATACHANGING,FOUNTAIN,FEATURED,DIRECTED,REVIEWED,TROUSERS,SC
ISSORS,ACCIDENT,CONTINUE,CARRAIGE,MAINTAIN,STOPPING,STARTING,PRAC
TICE,SMALLEST,NEGATIVE,POSITIVE,MESSAGES,SENTENCE,REPEATED,DECLAR
ED,INCIDENT,MULTIPLE,PREPARED,WHEREVER
630 DATABUSINESS,INCREASE,DECREASE,INDICATE,OPTIONAL,CONVERSE,NA
TIONAL,REQUESTS,ALPHABET,RESERVED,ACKNOWLEDGEMENT,PREFERENTIALISM
,ACCLIMATIZATION,AEROPHOTOGRAPHY,ANTISEGREGATION,BIOLUMINESCENCE,
BIOPSYCHOLOGIST,CATECHISTICALLY

```

```

640 DATACHRONOLOGICALLY,CIRCUMSPECTNESS,DECERTIFICATION,DECONTAM
INATION,ELECTROACOUSTIC,ELECTROMAGNETIC,FAMILIARIZATION,FUNCTIONA
LISTIC,GASTROENTERITIS,GEOMAGNETICALLY,HISTOPHYSIOLOGY,HOLOSYMMET
RICAL,IDIOSYNCRATICAL,IMMORTALIZATION
650 DATAJURISPRUDENTIAL,KALEIDOSCOPICAL,KNICKERBOCKERED,MATERIAL
IZATION,MELODRAMATIZING,METALINGUISTICS,NEUROPHYSIOLOGY,NONCONVEN
TIONAL,OBJECTIFICATION,OCEANOGRAPHICAL,PARASYMPATHETIC,PERFECTION
ISTIC,RADIOBIOLOGICAL,RECOGNIZABILITY
660 DATASANCTIMONIOUSLY,TECHNOLOGICALLY,ZOOGEOGRAPHICAL,SYMPATHE
TICALLY,THE/LAST/MILE/IS/LONGEST,YOU/GROW/GREY/GRACEFULLY,REMEMBE
R/TIME'S/PRECIOUS,TRY/TO/WORK/THIS/ONE/OUT,WE/BET/ON/A/DONKEY/AGA
IN,MARMALADES/FOR/BREAKFAST
670 DATAI/HAVE/FUN/PLAYING/GAMES,YOU/LOOK/BEFORE/YOU/LEAP,DO/YOU
/GET/MANY/PROBLEMS,HAVE/A/SMALL/DRINK/ON/ME,WORRIES/CAUSE/GREY/HAI
RS,SOME/PEOPLE/HAVE/IT/EASY,BE/KIND/TO/EVERY/ANIMAL!,A/LIFE/ON/T
HE/OCEAN/WAVE,MINCE/ON/CRUMPET/IS/NICE
680 DATAIT/PAYS/YOU/TO/ADVERTISE,LET/US/ALL/GO/TO/A/PARTY,EXERCISE/IS/GOOD/FOR/YOU,DOES/WRITING/PAPER/WRITE,YOU/ARE/A/GENIUS/AT/T
HIS,MY/HORSE/HAS/A/FLAT/FOOT,TICK/TOCK/GOES/THE/CLOCK,AN/APPLE/FO
R/THE/TEACHER,WE/LEARN/BY/OUR/MISTAKES
690 DATAABCDEFGHIJKLMNPOQRSTUVWXYZ
700 CLS:PRINT"THE IDEA OF THIS GAME IS TO UNSCRAMBLE THE LETTERS
AND FORM THE WORD OR SENTENCE, OR IN THE ALPHABET GAME PUT THE L
ETTERS IN ALPHABETICAL ORDER."
710 PRINT"ONCE THE LETTERS ARE SET USE THE ARROW KEYS (ESC & CTR
L)
FOR UP AND DOWN MOVEMENT AND '<' AND '>' FOR LEFT AND RIGHT
MOVEMENT TO RE-ARRANGE THE LETTERS. THE WORD SHOULD BE COMPLETED
READING FROM LEFT TO RIGHT DOWN THE ";
720 PRINT"SQUARE WITH THE BOTTOM RIGHT HAND SQUARE LEFT BLANK.
WHEN THE WORD IS COMPLETED PRESS > Q < AND THE COMPUTER WILL
CHECK YOUR ANSWER OR IF YOU WISH TO GIVE UP PRESS > Q <.
":GOSUB780
730 PRINT"YOU HAVE THREE DIFFERENT GAMES : EIGHT LETTER WORDS WH
ICH IS NORMAL PLAY OR IF YOU THINK YOU ARE A GENIUS TRY THE FI
FTEEN LETTER WORDS OR THE SENTENCES.
YOU WILL BE PLAYING AGAINST TIME IN EACH GAME BUT TO HELP YOU
AS YOU PLAY ";
740 PRINT"THE LETTERS OF THE WORD WILL BE PRINTED OUT.
EACH TIME A LETTER IS PRINTED YOUR SCORE WILL BE REDUCED SO THE
IDEA IS TO GET THE WORD OUT AS QUICK AS POSSIBLE.
IF YOU DO THE SENTENCES THEN EACH WORD MUST BE SEPERATED BY A
> / <. ";
750 PRINT"IF YOU COUNT THE > / < IN THE SQUARE AND ADD ONE THAT
WILL TELL YOU HOW MANY WORDS ARE IN THE SENTENCE.":GOSUB780
760 PRINT"HERES A COUPLE OF TIPS,
THE HARDER THE WORD THE MORE TIME YOU ARE ALLOWED
YOU WILL HAVE A FEW SECONDS TO MAKE EACH MOVE BEFORE A LETTER
APPEARS.
IF YOU GET ALL THE LETTERS SET THEN FIND THAT THE LAST TWO ARE
THE WRONG WAY ";
770 PRINT"THEN YOU WILL HAVE TO TAKE TWO LETTERS WHICH ARE THE
SAME AND SWAP THEM AROUND AND THEN RESET THE REST OF THE LETTE
RS":GOSUB780:GOTO640
780 PRINT:INPUT"PRESS ENTER TO CONTINUE";B#:CLS:RETURN
790 CLS:PRINT@20,"N U M B E R S U P"
800 A#="":PRINT:PRINT"PRESS > Y < IF YOU WANT INSTRUCTIONS OR
> N < IF NOT"
810 A#=INKEY#:IFA#="Y"THEN1240ELSEIFA#<>"N"THEN810
820 CLEAR60
830 CLS:FORX=15638TO16403STEP64:FORX=YTOY+18STEP6:POKEX,149:NEXT
:NEXT
840 FORX=15638TO15655:POKEX,191:POKEX+128,191:NEXT:POKEX,149:X=1
850 PRINT@220,"COLUMN #";
860 POKE15705,49:POKE15711,50:POKE15717,51
870 Y=RND(9):QNYGOTO880,890,900,920,930,940,950,960,970
880 M=16217:GOTO980
890 M=16223:GOTO980
900 M=16229:GOTO980
910 POKE15705,55
920 M=16281:GOTO980
930 M=16287:GOTO980
940 M=16293:GOTO980
950 M=16345:GOTO980
960 M=16351:GOTO980

```

Phone 16553 255

```

970 M=16357:GOTO980
980 N=PEEK(M):IFN<>32THEN870ELSEPOKEM,X+48:X=X+1
990 IFX<10THEN870
1000 A=4:B=4:FORW=15833TO15845STEP6:R=PEEK(W):IFR=57THEN1190ELSE
NEXT
1010 PRINT@0,"TOP NUMBER FROM COLUMN ";:INPUTA
1020 PRINT@32,"TO TOP OF COLUMN ";:INPUTB
1030 PRINT@64," "
1040 IFA>3ORB>3ORA<1ORB<1THEN1170
1050 PRINT@0,STRING$(55," ")
1060 U=U+1:ONAGOTO1070,1080,1090
1070 X=15833:GOTO1100
1080 X=15839:GOTO1100
1090 X=15845
1100 FORX=XTOX+576STEP64:N=PEEK(X):IFN=32THENNEXTELSE1110
1110 ONBGOTO1120,1130,1140
1120 Y=15833:GOTO1150
1130 Y=15839:GOTO1150
1140 Y=15845:GOTO1150
1150 FORY=YTOY+576STEP64:P=PEEK(Y):IFP=32THENNEXTELSE1160
1160 IFN>PTHEN1180
1170 PRINT@64,"INPUT ERROR TRY AGAIN":GOTO1000
1180 POKEX,32:POKEY-64,N:GOTO1000
1190 Z=58:FORW=WTOW+512STEP64:Q=PEEK(W):Z=Z-1:IFQ=ZTHENNEXTELSE1
000
1200 CLS:PRINT@210,"CONGRATULATIONS YOU HAVE DONE IT"
1210 PRINT@465,"YOU HAVE TAKEN ";U;" MOVES TO DO IT."
1220 PRINT:INPUT"DO YOU WANT TO PLAY THIS GAME AGAIN (Y/N)";A#:I
FA#="Y"THEN830ELSERUN
1230 CLS:PRINT"COME BACK ANYTIME YOU FEEL BRAINY":FORX=1TO5000:N
EXT:CLS:RUN
1240 CLS:PRINT"NUMBERS 1 TO 9 ARE ARRANGED IN THREE COLUMNS
OF THREE AND BYMOVING NUMBERS BETWEEN COLUMNS YOU HAVE TO ARRANGE
THEM SO THEY ARE ALL IN ONE COLUMN WITH 1 ON THE BOTTOM AND IN
ORDER TO 9 ON THE TOP."
1250 PRINT"TO MOVE THE NUMBERS, ONLY THE TOP NUMBER CAN BE MOVE
D FROM ONE COLUMN TO ANOTHER AND ONLY TO A COLUMN WHOSE TOP NUMBE
R IS LOWER THAN THE ONE YOU ARE MOVING.":PRINT:PRINT"TO MOVE A NUM
BER TYPE IN THE COLUMN NUMBER OF THE NUMBER ";
1260 PRINT"YOU WISH TO MOVE - PRESS ENTER - THEN TYPE IN THE
COLUMN NUMBER YOU WISH TO MOVE THE NUMBER TO AND PRESS ENTER.":PR
INT:PRINT"DEPENDING ON HOW THE NUMBERS ARE SET YOU SHOULD BE ABLE
TO DO IT IN LESS THAN 100 MOVES."
1270 PRINT:INPUT"NOW PRESS ENTER TO BEGIN ";A#:GOTO830
1280 CLS:INPUT"ENTER HOW MANY TRIES DO YOU WANT TO GUESS THE NUM
BER (UP TO 6)";T:IFT<1ORT>6THEN1280
1290 P=0:Z=Z+1:J#="NUMBER":CLS:N=RND(100)
1300 IFT<2THENT=2
1310 PRINT"YOU WILL HAVE ";T;" TRIES TO GUESS THIS NUMBER"
1320 D=RND(6)+6:B=N:GOSUB1380:GOTO1400
1330 J#="ANSWER":IFP=T+1THEN1540ELSEPRINT:PRINT"YOUR GUESSES SO
FAR ARE ";:FORX=1TOP-1:PRINTG(X);":":NEXT
1340 U=U+1:PRINT:PRINT"YOUR GUESS NO ";P;" IS ";:INPUTG(P)
1350 IFG(P)=NTHEN1500ELSEIFG(P)>NTHEN1370
1360 D=RND(11)+1:B=N-G(P):GOSUB1380:GOTO1390
1370 D=RND(11)+1:B=G(P)-N:GOSUB1380:GOTO1390
1380 C=INT(B/D):E=B-(C*D):RETURN
1390 PRINT@128," IF YOU TAKE THE FOLLOWING TWO NUMBERS-
YOUR LAST GUESS AND THE NUMBER I THOUGHT OF
SUBTRACT THE SMALLEST FROM THE LARGEST":GOTO1410
1400 PRINT@192,"THE NUMBER I AM THINKING OF IS AS FOLLOWS:-":GOT
O1420
1410 PRINT"THE ANSWER IS AS FOLLOWS:-"
1420 PRINT:PRINT"THE ";J#;" DIVIDED BY ";D;" LEAVES A REMAINDER
OF ";E
1430 FORF=2TO8:H=INT(B/F):I=B-H*F:IFI=0THEN1450ELSENEXT
1440 PRINT:PRINT"THE ";J#;" IS A PRIME NUMBER":GOTO1460
1450 IFF*1=BTHENNEXTELSEPRINT:PRINT"THE ";J#;" IS NOT A PRIME NU
MBER"
1460 V=INT(B/2):W=B-(V*2):IFW=1THEN1480
1470 PRINT:PRINT"THE ";J#;" IS AN EVEN NUMBER":GOTO1490
1480 PRINT:PRINT"THE ";J#;" IS AN ODD NUMBER"
1490 P=P+1:GOTO1330
1500 T=T-1:Q=Q+1:CLS:PRINT@25,"YOU GOT IT"
1510 PRINT"THE NUMBER WAS ";N;" AND YOU TOOK ";P;" TRIES TO GET
IT"

```

```

1520 PRINT:PRINT"YOUR SCORE SO FAR IS
";Q;" NUMBERS GUESSED OUT OF ";Z;"
WITH AN AVERAGE OF ";U/Z;" TRIES":PRINT:INPUT"DO YOU WANT TO PLA
Y THIS GAME AGAIN (Y/N)";A#:IFA#="Y"THEN1530ELSERUN
1530 FORX=1TOP-1:G(X)=0:NEXT:GOTO1290
1540 T=T+1:CLS:PRINT@25,"SORRY YOU MISSED OUT":PRINT"THE NUMBER
WAS ";N:GOTO1520
    
```

\*\*\*\*\* DISK INDEX

(C) Bruce H. & Y.L. Bussenschutt \*\*\*\*\*

The ideas from this program should enable you to have a master index of all the BASIC programs on any disk, or even for your complete disk system. You can construct your own directory (which can be in English) and the program will let you select load/run by inputting a 2 figure number. INDEX is in BASIC.

To operate, when you commence loading from any disk, type BASIC ENTER etc., in the usual way, and then RUN "INDEX" or RUN "INDEX:1" or :2" or :3" as appropriate. Your index of up to 36 programs is displayed, and a bottom line prompt invites you to select your program number. When the number is ENTERed, INDEX checks that the selection is possible and then LOADs and RUNs the selected program.

As LISTed the program is incomplete. It cannot work until you modify it in 2 places for each program to be selected (see below). If the program is RUN as it now stands, it will display:

```

                INDEX ... DISK NO. ##
1. #####          13. #####          25. #####
2. #####          14. #####          26. #####
3. #####          15. #####          27. #####
4. #####          16. #####          28. #####
5. #####          17. #####          29. #####
6. #####          18. #####          30. #####
7. #####          19. #####          31. #####
8. #####          20. #####          32. #####
9. #####          21. #####          33. #####
10. #####         22. #####          34. #####
11. #####         23. #####          35. INDEX - DISK ##
12. #####         24. #####          36. RETURN TO BASIC
..... SELECT PROGRAM NUMBER AND >ENTER< ?
    
```

The #####s in the display are places for you to include your own program names.

Notice also that lines 100 to 3600 are a series of RUNs which have "#####/BAS" after them. Each RUN (when modified) will LOAD and RUN the program names. For easy identification, the RUN line number is 100 times the INDEX number displayed - e.g. program No. 27 RUNs from line 2700.

The selection No. prompt on the bottom line has a flashing cursor. The routine for this is in lines 4000 to 4080. You can leave this out if it is not desired. You would of course need to delete GOSUB4000:N=VAL(F\$) from line 70 and substitute INPUT N. If you want a different cursor, change the 136 in line 4020 to something else, say 191 or 95 or even 42.

To customise the program for each disk:

1. Obtain a written list of the programs on the disk. Use DIR or DIR :1 as appropriate.
2. Load INDEX and then change the #'s (using EDIT of course) in 2 places to coincide with your written list. The names need not be identical.
  - (a) Change the #####s in the display line to your program name, or to any English descriptive word or phrase up to 15 characters.
  - (b) Change the #####s in the corresponding RUN statements to coincide with your program names as SAVED onto disk. Assuming that each name has the maximum 8 letters, the 8 #'s can be EDITed to become the 8 letters of the name. Also it is assumed that your BASIC programs have been given the /BAS subscript, so this has been already typed to save you doing it.

When you have done this, your disk might have, say, 1. SPACE INVADERS in the display area and INVADER/BAS in the RUN line.

There is an inbuilt error trapping routine, in case you type in a program number that only refers to the #'s in the RUN lines. This is in lines 3 and 3900.

If your disk fills with less than 36 programs, the appropriate selection numbers and #####s can be deleted altogether from the display and their corresponding RUN lines EDITed out. An easier way is to leave all the #'s in place, and change the 36 in line 3600 to your maximum program number.

The program can also work as an index for more than one disk or more than 36 programs. For example, you may have 2 drives, and two disks, and the disk in drive 0 has DOS, INDEX and games programs, while the disk in drive 1 has only games. If all programs are INDEXed and if your RUN statement has the form RUN "program/BAS:1" where program/BAS is your actual program name, then it will load from disk 1. The only limit is the amount of display on the screen for selection purposes. At the moment there is room for 36 names, each of 15 characters. By changing the TABS from 21 and 42, to 16 and 32, would allow another statement on each line, starting with line No. 10 which would have PRINTTAB (48) "37. #####" and so on. This would limit your index display to a maximum of 11 characters for the program name. Change lines 3700 onwards to start from 4900, and put in lines 3700 to 4800 as RUN lines. The display will be fairly cramped when it is all completed though.

Another possibility is to link INDEXes. SAVE each disk INDEX program with its title corresponding to the diskette number. The index on diskette 1 could be called INDEX 1. If you have four disks full of games, INDEX 01 could link to the others by having:

```

34 GAME DISK 2 on the display with the corresponding 3400 RUN "INDEX02:1"
35 GAME DISK 3 " " 3500 RUN "INDEX03:1"
36 GAME DISK 4 " " 3600 RUN "INDEX04:1"

```

Lines 3400, 3500, 3600 could also have a built in prompt to ensure that you have the disk in the drive, e.g.

```
3400 INPUT "INSERT DISK 2 IN DRIVE 1";EN$:RUN "INDEX02:1"
```

Another neat way of operating is to have your programs link back to the INDEX when you have finished. Probably the tidiest way is to have the following at the end of every program on your disk instead of the existing END statement:

```

50000 CLS:PRINT @ 550,"";:INPUT"RETURN TO INDEX (Y/N) ";I$:IF LEFT$(I$,1)="Y" GOTO 50020
50010 GOTO 10 (or to your first line number of that program)
50020 RUN "INDEX"
50030 END

```

To avoid confusion, change I\$ to something else if your program has an existing I\$. If any of the existing END statements of your BASIC programs are of the non-essential type you may need to include a program statement that says GOTO 50000 so that the RETURN to INDEX will work. It is suggested that each INDEX program selection 36 will be a RETURN TO BASIC to allow you to get out of the INDEX process. This is achieved by GOTO 3999 which gives you the normal READY prompt. The INDEX is still in memory and can be RUN again if needed, or cancelled when loading another program or typing NEW.

The possible linkups are endless, and are only limited by your imagination.

```

10 ' ***** DISK INDEX *****
* B.H. & Y.L. BUSSENSCHUTT *
* VALLEYVIEW DRIVE *
* HIGHBURY S.A. *
*****
20 ON ERROR GOTO 550
30 CLS:PRINT@12,"INDEX ... DISK NO. ## ":PRINT
40 PRINT" 1. #####";:PRINTTAB(21)"13. #####";:PRINTTAB(42)
"25. #####"
50 PRINT" 2. #####";:PRINTTAB(21)"14. #####";:PRINTTAB(42)
"26. #####"

```

```

60 PRINT " 3. #####";:PRINTTAB(21)"15. #####";:PRINTTAB(42)
"27. #####"
70 PRINT " 4. #####";:PRINTTAB(21)"16. #####";:PRINTTAB(42)
"28. #####"
80 PRINT " 5. #####";:PRINTTAB(21)"17. #####";:PRINTTAB(42)
"29. #####"
90 PRINT " 6. #####";:PRINTTAB(21)"18. #####";:PRINTTAB(42)
"30. #####"
100 PRINT " 7. #####";:PRINTTAB(21)"19. #####";:PRINTTAB(42)
)"31. #####"
110 PRINT " 8. #####";:PRINTTAB(21)"20. #####";:PRINTTAB(42)
)"32. #####"
120 PRINT " 9. #####";:PRINTTAB(21)"21. #####";:PRINTTAB(42)
)"33. #####"
130 PRINT "10. #####";:PRINTTAB(21)"22. #####";:PRINTTAB(42)
)"34. #####"
140 PRINT "11. #####";:PRINTTAB(21)"23. #####";:PRINTTAB(42)
)"35. INDEX - DISK ##"
150 PRINT "12. #####";:PRINTTAB(21)"24. #####";:PRINTTAB(42)
)"36. RETURN TO BASIC"
160 PRINT:PRINT " ..... SELECT PROGRAM NUMBER AND >ENTER< ?";:G
OSUB570 :N=VAL(F#)
170 IF N>36 THEN CLS:PRINT@512,CHR$(23)"NUMBER BETWEEN 1 AND 36
PLEASE";:FOR X=1 TO 1000:NEXTX:GOTO30
180 ON N GOTO 190,200,210,220,230,240,250,260,270,280,290,300,31
0,320,330,340,350,360,370,380,390,400,410,420,430,440,450,460,470
,480,490,500,510,520,530,540
190 RUN "#####/BAS"
200 RUN "#####/BAS"
210 RUN "#####/BAS"
220 RUN "#####/BAS"
230 RUN "#####/BAS"
240 RUN "#####/BAS"
250 RUN "#####/BAS"
260 RUN "#####/BAS"
270 RUN "#####/BAS"
280 RUN "#####/BAS"
290 RUN "#####/BAS"
300 RUN "#####/BAS"
310 RUN "#####/BAS"
320 RUN "#####/BAS"
330 RUN "#####/BAS"
340 RUN "#####/BAS"
350 RUN "#####/BAS"
360 RUN "#####/BAS"
370 RUN "#####/BAS"
380 RUN "#####/BAS"
390 RUN "#####/BAS"
400 RUN "#####/BAS"
410 RUN "#####/BAS"
420 RUN "#####/BAS"
430 RUN "#####/BAS"
440 RUN "#####/BAS"
450 RUN "#####/BAS"
460 RUN "#####/BAS"
470 RUN "#####/BAS"
480 RUN "#####/BAS"
490 RUN "#####/BAS"
500 RUN "#####/BAS"
510 RUN "#####/BAS"
520 RUN "#####/BAS"
530 IFN=35THENCLS:INPUT"INSERT DISK ## IN DRIVE 1 AND >ENTER< ";
EN#:RUN"INDEX##:1"
540 IF N=36 THEN CLS:GOTO560
550 CLS:PRINT@512,CHR$(23)"PROGRAM SELECTED IS NOT INDEXED";:FOR
X=1 TO 1000:NEXTX:CLS:RESUME30
560 END
570 F#="":'BLINKING CURSOR SUB-ROUTINE
580 P=256*(PEEK(16417)-60)+PEEK(16416)
590 PRINT@P,CHR$(136);:FOR W=1 TO 10:NEXT
600 IN#=INKEY#
610 IF IN#=""THEN PRINT@P," ";:FOR W=1 TO 10:NEXT:GOTO590
620 IF ASC(IN#)=13 THEN PRINT@P,"":RETURN
630 IF ASC(IN#)<32 THEN POKE 15360+P,32
640 PRINT@P,IN#;
650 F#=F#+IN#:GOTO580

```

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

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

\*\* BIRTHDAY LI/4K \*\*

Tell this program today's date followed by the date of your birth and the program will tell you how old you are, how many years so far you have worked, slept, eaten, rested and when you are due to retire - also it will tell you, based on Australian averages, how many sheep, cattle and chickens you have eaten and how many hectares of grain and plants you have eaten your way through.

\*\* UTIL1 : UTIL2 : UTIL3 : UTIL4 : \*\*  
LII/4K-16K m/L

A real bonus here; each of these utility programs in machine language is slightly different to suit individual systems. Some of them include all of the functions below and some don't, some are at the bottom of 4K, some are at the top of 4K and they have been written to be easily relocatable for 16K plus systems. They provide a flashing cursor, auto character repeat, slow scroll and lower-case driver.

\*\* COSMIC WARS 32K/DISK \*\*

Just so they don't feel left out, here is a space war game for disk users. It simulates an arcade game called COSMIC KNOCKOUT. The game has a built in timer which only gives you 90 seconds to play; well at least you don't have to put in 20 cents each time.

\*\* GUZINTA LI/4K \*\*

This program is a space game for young children, but before they get to play they have to solve a problem. What sort of problem? What does GUZINTA mean? Say "4 GUZINTA 8 TWICE" a few times and you might catch on.

\*\* BINARY CONVERTER LII/4K \*\*

We have had all sorts of conversion programs, hex to dec, miles to kilometres etc. now this program which, when you type in a binary number, i.e. 10110110 will convert it and provide the equivalent number in decimal and hexadecimal. This program will be useful when the number in a book you are referring to is only supplied in binary.

\*\* BIG LETTERS LII/16K \*\*

This program draws big letters on the screen in either a normal or negative mode which may be mixed on the screen at the same time. The program also allows the message to be saved to tape and loaded back in at a later date.

\*\* LINEAR REGRESSION LII/4K \*\*

As promised, this is the second program in the series of scientific programs. As the name implies, this program computes the linear relationship between two sets of data.

APPLICATION FOR PUBLICATION  
OF A PROGRAM  
IN MICRO-80

Date .....

Tick where appropriate

To MICRO-80

Please consider the enclosed program for ...

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

Name .....

Address .....

Postcode .....

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

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

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

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

The changes or improvements that you think may improve it.

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





# 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