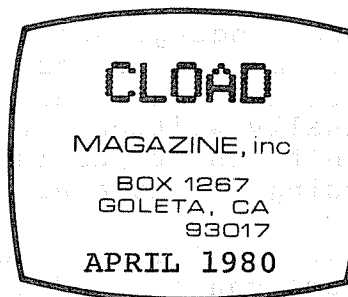


Here's April!

Subscribers who have been with us for some time will be happy to hear that we are shifting from copying our "yellow sheets" on our office copier to having them printed at a for real print shop. After several hundred thousand copies, our faithful copier is getting a break (mainly because it doesn't fold the sheets). Ah, success...



*****				
* Level Title Turns Count *				
* CTR-41 CTR-80 *				
*****				
**	Boxes Cover	20 & 263	11 & 155	*
**	Craps	61 & 295	36 & 174	*
**	Distance scales	136 & 352	80 & 207	*
**	Hi-Que	202 & 404	118 & 238	*
*****				
**	Boxes Cover	12 & 37	6 & 21	*
**	Craps	61 & 103	36 & 60	*
**	Distance scales	142 & 181	83 & 106	*
**	Typing	219 & 291	129 & 172	*
**	Syzygy	356 & 381	209 & 224	*
**	Change	404 & 425	238 & 251	*
*****				
System/change/30208)				
*****				

"Craps" is an interesting demonstration of how a computer flow chart operates. Each path in the algorithm is dynamically plotted as it is traversed. I have a friend who teaches a beginning programming class. He informs me that the old dice game of craps is the ideal program to use as an example problem to program. This version is sort of overkill on the subject, but it shows how the game works better than any explanation I've ever seen.

"Distance Scales" is good for beginning astronomers and space junkies of all types. It demonstrates that you can get there from here, but pack a big lunch and bring along some reading material, it's a long trip.

"Hi-Que" is the old pegboard game where you move each peg by jumping over another peg. A perfect game is one that ends with one peg left, in the exact center of the board.

"Typing" is a good one for those who want to learn touch typing. A computer is the ideal machine to learn typing on, for those of us so fortunate to have one. An observation I've made, however, is that few programmers of my acquaintance can type (or spell).

"Syzygy" is a worm game. In this case, the idea is to encircle your opponent with your worm. Each player is given five missiles (presumably because a worm has five hearts).

"Change" is a system tape that allows a programmer who has written a

machine language subroutine on a level II 16K machine to make a level I 4K system tape out of it (assuming that it doesn't need the level II ROM subroutines and it does not load higher than 4FFFFH). This one was originally written for in-house use, but turned out so nice that we just couldn't resist publishing it. There is a bit of explanation required for using it, however, so here goes:

The level I machine does not have a "system" tape setup as such. Programs can be loaded into RAM and directly executed, but the level I BASIC ROMs have to be "tricked" into letting the program in RAM take over. The technique is generally known as "crashing the stack". Here's how the trick works: when you load a tape program in the level I machine, you do so with the CLOAD command (catchy name, eh?). The ROM calls its internal tape loading subroutine to load what it thinks is a BASIC program (which is all it has ever been told about). Just before executing the tape load subroutine, however, it writes a return address down on the "stack" - an area of RAM memory set aside for short notes of this type.

The level I tape data format starts out with a memory address. The tape loading subroutine puts the data which follows into memory starting at this address. This way, a program can be written that loads itself into any memory area of the level I machine.

After loading the tape data, the tape loading subroutine goes back to the ROM program which CALLED it. It does this by the RETurn instruction, which gets the address to go back to from the stack. But wait! The stack is just another area in memory, and we can load data into any area of memory because the load address is on the tape. Aha! If we know where the level I's stack is (we do, it's at 41FEH and 41FFH), we can make a tape that starts loading at that place.

This is where it starts getting complex - we set up a tape to start loading at 41FEH. The first two bytes of "data" on the tape are 00H and 42H. When loaded into the stack area (41FEH and 41FFH), they form an address of 4200H. The data which follows is a machine language program which starts at 4200H. When the tape loading subroutine finishes loading what it thinks is a BASIC program, it RETURNS to the ROM, or so it thinks. It wrote the address to go back to on the stack, and it assumes that the address is still there. It isn't - we changed it to 4200H. The tape loader isn't swift enough to realize that it's been fooled, so it RETURNS to 4200H, passing control to the start of our program.

So much for the trick of crashing the stack. Now let's talk a bit about the program "Change". The simplest way to use it is to write a machine language program designed to run on a level I machine, at an address of 4200H. Load the program "Change", then load your machine code program. "Change" will ask you to put a blank tape in the recorder, and then it will dump it at the level I rate (250 baud), in level I format, with a stack crash pointer of 4200H. Suppose for some reason you don't wish to have your program run on the level I machine at 4200H. "Change" can handle this in two ways. Proceed as before, first loading "Change", then your machine language program. "Change" will sense that the load address is not 4200H, and will ask you to specify either null fills or a two program setup. The null fill option will create a tape that loads starting at the stack crash, continuing with 00H's (NOP command) from 4200H to the start of your program, and then your program. This can be a drag to load for those programs stuffed into the higher part of RAM. The other option creates two programs, the first one being your machine language program, and the second one being a stack crash pointer (that

is, a "program" consisting of the two bytes needed to pass control to wherever your program is at. This requires two CLOAD commands from the user of the level I machine.

For those of you who are getting into the machine language swing of things, the ability to CALL subroutines in the level II ROMs is an obvious advantage. This allows a USER function written in machine language to take advantage of the compact, powerful subroutines that are already there. The only problem has been that documentation on what is available and how to use it has been essentially nonexistent. The book from Mumford Micro Systems (see ad in this issue) has been written to fill this need. It is 65 pages long and contains 18 chapters, covering the topics from internal representation of numbers and protocol of calling subroutines to specific examples of linking BASIC and machine language programs to form a composite. At the risk of sounding like I have an interest in the project (I don't), I can truthfully say that Everything You Always Wanted To Know About The Level II ROMs But Were Tired Of Asking Is Right Here, Folks.

And for those of you who would like to sample the delights of working with assembler for the IBM 360 & 370 category of computers, Balcode Software has a programming system which allows a programmer to enter code in BAL (stands for Basic Assembly Language - no relation to BASIC) and assemble code that will run on the TRS-80. Opinions vary on the artistic merits of the large IBM mainframe computers, but anyone who claims that they are no longer the dominant computers of business and scientific data processing has been drinking from the bit bucket. Those who wish to pursue a career in computer science will find that these computers are worth knowing about.

See You Next Month

*Ralph*

Ralph McElroy, Publisher

## EVEN COMPUTERS GET THE BLUES

Has your **TRS-80** been sluggish lately? Slow to respond? Had excessive keyboard bounce?

The problem might be low voltage, or a BASIC misunderstanding, or **IRON POOR SOFTWARE!**

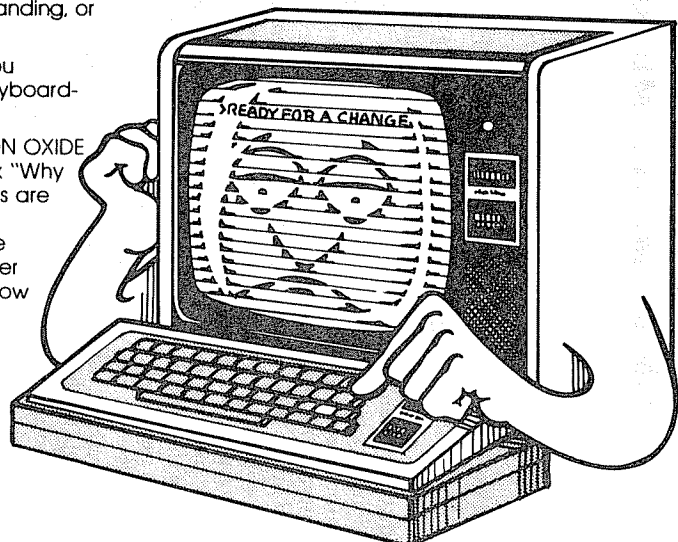
Do you serve your **TRS-80's** meals on paper sheets? Do you (shudder) write it yourself? Recent studies indicate that keyboard-feeding causes **MALIGNANT BUGS!**

**CLOAD Magazine** is published monthly on a magnetic **IRON OXIDE** tape, wound up inside a C-30 cassette. Now you may ask "Why bother?", but I can assure you that our computer cassettes are **DIRECTLY** readable, I repeat **DIRECTLY** readable by your computer. We have Thrills, Variety, and Absurdity. We have every program your computer has ever wanted to run after a hard day at the job. We even include our infamous "yellow sheets" with every issue, filled with lies about the **TRS-80** computing scene.

12 Monthly cassette issues (over 60 programs)	\$36.00 *
Single issues	\$ 3.50 *
Best of <b>CLOAD</b> (9 programs w/ listings)	\$10.00 *

\* CA residents please add 6% to non-subscription orders  
Please write for overseas rates

Master Charge / Visa Welcome Also Cash & Gold



© Copyright CLOAD MAGAZINE 1980

## ORIGINAL HARDWARE AND SOFTWARE FOR THE TRS-80 FROM MUMFORD MICRO SYSTEMS

### SPEED UP YOUR TRS-80

The SK-2 is a sophisticated clock modification which allows the user to select either normal operation at 1.77MHz, a 50% increase to 2.66MHz, or a 50% decrease to .88MHz. In addition, instructions are included for trying a 100% speed increase to 3.54MHz (not recommended for the TRS-80). Speed may be changed AT ANY TIME without interrupting program execution or "bombing" the computer. The SK-2 may be configured by the user to change speeds on software command, with a toggle switch, or both. It may also be tied to the expansion interface so it will automatically return to normal speed anytime a disk is active. Provision has even been made for adding an LED to indicate when the computer is not at normal speed. The high speed shortens CLOAD and CSAVE times as well as computational and decision making time (as in chess programs and sorts), and adds a new level of difficulty to action games. The slow speed simplifies debugging of rapid sequences, slows a LIST down to a readable rate, and eliminates keyboard bounce without software overhead. The kit consists of a small printed circuit board with 5 socketed IC's. It mounts inside the keyboard unit with only 4 necessary connections to the existing circuit for easy removal in case service is ever required on the computer. It comes fully assembled and tested with instructions for implementing the various options. **SK-2....\$24.95**

### THE PROGRAMMERS GUIDE TO LEVEL II ROMS

INSIDE LEVEL II is a comprehensive guide to the internal operations of the LEVEL II Basic software which allows the machine language programmer to easily access and use the sophisticated routines already resident in the ROMs. It clearly explains set-ups, calling sequences, variable passage, and I/O routines. Special instructions are given to allow disk users to fully utilize the Basic routines when Disk Basic may not be resident. Part II presents a new composite program structure which loads under the SYSTEM command and executes in both Basic and machine code with the speed and efficiency of a compiler. In addition, the 18 chapters include a large body of other information useful to the programmer. **INSIDE LEVEL II....\$15.95**

### DUPLICATE SYSTEM TAPES

CLONE is a machine language program used for making duplicate copies of any tape written for Level II (CLONE itself excepted) directly from your computer. They may be SYSTEM tapes (continuous or not) or data lists. If you can load it you can CLONE it. It is not necessary to know the file name or loading and starting addresses. CLONE displays the file name of programs being copied and prints the ASCII representation of every byte on the video screen in sequential order (up to 896 bytes at a time). Checksums are verified to guarantee a good load and the start and execute addresses of each program are displayed. Program text may then be modified before the copy is produced. Used with the clock speed—up above, you may easily generate fast loading versions of long machine language programs. **CLONE....\$16.95**

### ORGANIZE YOUR PROGRAM LIBRARY

DISK INDEX is a comprehensive system for generating an alphabetized index of your entire program library from disk directories. The program reads directory entries and free space available directly from the video screen into an index file which may then be alphabetized (with a fast Shell/Metzner sort) by user assigned disk names or by program names. The list may also be searched for any disk name or program name, programs or disks added or deleted, and the whole list or any part sent to the line printer. Finally, the list itself may be stored on disk for future access and update. One drive and 32K required. For a more complete description see page 34 of the January issue of '80 Microcomputing. **INDEX....\$19.95**

### EDIT BASIC PROGRAMS WITH ELECTRIC PENCIL

PENPATCH is a utility program used to modify disk files of ASCII information so that they may be loaded into the disk version of Electric Pencil for editing. These files may be Basic program texts or general data lists. When working on Basic programs, line numbers may be edited, program segments moved or duplicated, and the occurrence of any group of characters or variables searched for. On data files, complete editing freedom of information is allowed. In both cases, the modified file generated by Electric Pencil will load back into Basic or the program which created it. PENPATCH adds the single word command "PATCH" to your operating system which will quickly modify ASCII files to ELECTRIC Pencil format. **PENPATCH....\$9.95**

### SPOOLER FOR PARALLEL PRINTERS

SPOOLER is a full feature print formatting package for parallel printers. It supports user definable line and page lengths (with linefeeds inserted between words or after punctuation), pause between pages, automatic indent, screen dump, keyboard debounce, and printer halt control. In addition, printing is done from a 4K expandable buffer area so that the LPRINT or LLIST commands return control to the user while printing is being done. Ideal for Selectric or other slow printers. Stop waiting for your printer to catch up to your computer by allowing printing and processing to run simultaneously. **SPOOLER....\$16.95**

### RAM TEST PROGRAM

RAMTEST is a Level II program that runs in machine code to test memory locations for open or shorted address or data lines, as well as intermittents. In addition to testing each bit for validity, each byte is tested in actual execution of an instruction cycle as in real program execution. All bad addresses are displayed along with the bad data and the proper data. A full test of 48K RAM takes just 14 seconds. It also has a separate test for errors produced by power line glitches from other appliances. **RAMTEST....\$9.95**

Please include .75 postage. California residents add 6% sales tax. All programs are usually shipped on cassette. Add \$4 for disk. Complete satisfaction or full refund is guaranteed.

## MUMFORD MICRO SYSTEMS

Box 435—B Summerland, California 93067 (805) 969-4557