



BITS, BYTES & PIXELS

LIMA AREA 99/4A USER GROUP
VOLUME II NO 10
OCTOBER 1986



COMMENTARY

The 99/4A is about to celebrate an anniversary. This is not the kind of anniversary that would cause one to blow horns and drink bubbly but it is a date to remember. In November of 1983 TI made the decision to abandon the market place for its 99/4A. As a result all of our computers and peripherals became orphans. Unlike the Edsel it is not the butt of jokes and epithets but an object to be sought and to undergo continuous development of applications.

This, we believe, speaks well for the machine and for the persons that own them. Unlike Timex, Atari, Coleco etc the TI survives and prospers.

We believe that it is because the TI 99/4A is basically a technology that at the time of its introduction was well ahead of its times. As a result it continues to challenge the minds of owners, software developers and inovaters who continue to place peripherals in the market.

So, if it is an orphan it seems to be in demand for adoption.

MISCELLANY

A railroad company used a set of questions to test each applicant for the job of flagman. The first question the interviewer asked was, "What would you do if you saw two trains coming at each other on the same track at 80 miles per hour?"

One applicant thought about it for a while and then said, "I'd run home and get my brother."

"Why would you do that?" asked the the supervisor.

"Well, my brother ain't never seen a train wreck before."

BRAIN TEASER

What is the largest sum of money in current United States coins (not including silver dollars that a person can have in his/her pocket without being able to give change for a dollar, half dollar, quarter, dime or nickle?

WANT AD

TI console with modulator and power pack. All for \$50.00
Contact: Chuck Alsbaugh 223 7646

*
*
* BITS, BYTES & PIXELS *
*
* Published by Lima *
*
* 99/4A User Group *
*
* Hal Sehnert Editor *
* Charles Good Tech. *
* Editor *
*
* Material contained herein *
* may be copied by any user *
* group as long as credit *
* is given to source. *
* Address: 2225 High Ridge *
* Lima, Oh 45805 *
*
* Published Monthly except *
* June, July, August. *
*

THE WALL STREET JOURNAL



"Just take a look at this onboard computer. It can figure square roots, percentages and the interest rate of your monthly installment."

WHAT IS YOUR SYSTEM WORTH?

The prices below are reprinted with slight modifications from the April 86 issue of Chicago Times. Our own experience indicates that these prices are probably still valid. The low price is the lowest known for used equipment. The high price is a price for for new equipment. New prices from dealers may be somewhat higher or lower.

Item Name	Low	High
Full height SS bare drive	\$ 30	\$ 60
Full height DS bare drive	\$ 60	\$100
Half height DS bare drive	\$ 80	\$130
Disk drive power supply	\$ 30	\$ 75
TI99/4A console	\$ 40	\$ 85
Expansion Box (empty)	\$ 80	\$160
32K Memory card	\$ 45	\$ 95
128K Memory card	\$185	\$210
512K Memory card	\$250	\$300
RS232 card	\$ 50	\$100
TI Disk Controller card	\$ 20	\$ 75
Corcomp Controller card	\$115	\$185
Myark Controller card	\$150	\$185
Speech Synthesizer	\$ 25	\$ 50
Monochrome Monitor	\$ 50	\$150
Color Monitor (composite)	\$ 75	\$150
300 Baud Modem	\$ 10	\$125
1200 Baud Modem	\$150	\$275
Corcomp Clock card	FREE	\$150
SG10 Printer	\$200	\$240
Gemini 10X printer	\$ 75	\$175

The following TI EXCHANGE PRICES were obtained from the TI-CARES phone service on Sept. 20. The first number is what TI charges to replace your defective OUT OF WARRANTY product with a new or rebuilt TI product. The second number is that TI charges for postage and handling. You also have to add your own state's sales tax. Since TI is licenced to do business in all 50 states, you have to pay state sales tax even though you are dealing with Lubbock Texas. Your total cost is TI's fee + TI's postage & handling + sales tax on TI's fee + your postage to Lubbock.

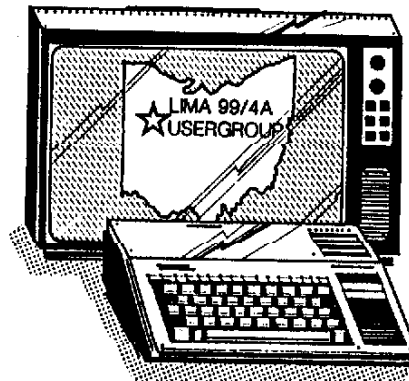
Recent personal experience indicates that turn around time is about 15 days. Those TI exchange centers sure were handy!! It is interesting to note that in some cases you can purchase the product new from catalogs for ~~less~~ than TI wants. Such products are marked with "*".

TI99/4A console	\$30.50	+\$3.00
P.E. Box	\$55.00	+\$8.00
RS232 card	\$33.00	+\$3.00
Disk Controller card	\$44.00	+\$3.00
32K Memory card	\$44.00	+\$3.00
Stand Alone Drive	\$80.00	+\$3.00
Drive for P.E. Box	\$60.50	+\$3.00
TI-Writer package*	\$33.00	+\$3.00
Multiplan package*	\$33.00	+\$3.00
Logo II package*	\$33.00	+\$3.00
Extended Basic*	\$33.00	+\$3.00
Mini Memory*	\$33.00	+\$3.00
Terminal Emulator II*	\$24.95	+\$1.50
Speech Synthesiser*	\$30.00	+\$3.00
most Game Modules*	\$10.95	+\$1.50

TEXAS INSTRUMENTS PHONE NUMBERS

TI-CARES.....	800-842-2737
TI-RESPONSE.....	800-232-3200
Business Computers...	800-847-2787
Software.....	800-858-4075
Technical.....	806-741-2663
Manuals and Parts....	806-741-3064

Home Computer owners should use the TI-CARES number first. When calling about repairs it helps if you have a product number (such as PHP1850). Otherwise the operators sometimes have trouble finding your product on their long product lists.



ATTENTION: II-WRITER USERS

If you don't yet have a copy of the official II-Writer manual (because you are using FUNLWRITER or something else other than the "official" version) you can now get a copy directly from II for just the cost of postage (\$3.00) while supplies last. Call 800-II-CARES for details. You can also borrow the manual that is in the club's library, but it will cost you more than \$3.00 to xerox this manual.

FROM THE GRAPEVINE

The CorComp Triple Tech card has a problem. One of its circuits charges the lithium battery every time the PE box is on. But, lithium batteries are not supposed to be charged, and some of these cards have blown up.

Recommendation: if you don't use the clock feature of the card, remove the battery. Source: Chicago Times newsletter.

Some Myarc DS/DD disk controller cards (reviewed elsewhere in this issue) randomly blow disk sectors. This has been reported by several users. Source: Sept 86 99'er News, the newsletter of the Will County IL Users Group.

Three members of the Lima Ohio User Group have purchased flexible "Peripheral Extension Cables" from the Tenex catalog, and none of these cables have worked properly. After flexing too much some of the small wires in these ribbon cables break and the cables give irregular operation as they flex. Sometimes they work and sometimes they don't. In two cases, members blame the cables for causing computer peripherals to blow up.

CHARLIE SAYS:

Microsoft MULTIPLAN is the best spreadsheet for the II99/4A. It is a full featured program. If you learn to use it on the II it will be very easy to learn on most other computers. Presumably multiplan data files can be interchanged between different kinds of computers. And now, MULTIPLAN is cheap. The latest TRITON catalog lists multiplan for \$29.95. The club has the updated MULTIPLAN disk in its library, but you need to purchase the module to load the disk.

MULTIPLAN is initially set up for single drive operation and has a printer default of RS232.BA=300. It is much easier to use with two drives (or one drive and a RAMdisk), and most of us have printers other than the above default. If you want 2 drive operation or a different printer default it is necessary to set up MULTIPLAN each and every time you boot the program. You can modify the MULTIPLAN disk to permanently reconfigure your program to 2 drive operation and your printer default. Here's how.

Using a disk sector editor such as DISKO examine the second sector of the MPINIR file (this is sector >110 on my disk). Look in ASCII for "DSK1." followed by a single space. Changing this to "DSK2." makes all data files automatically load and save to drive 2. Look in this same sector for "RS232.BA=300" followed by several spaces. Change this to "PIO" or whatever your printer default is and type spaces over any left over letters from the original printer default. Your MULTIPLAN disk is now customized for your printer.

CHARLES GOOD

A DIFFERENT WAY TO TYPE IN PROGRAMS

by ANDREW FRUEH

Most BASIC commands can be entered with a single keystroke if you first put a REM or (in extended basic) ! immediately after a line number. From EXTENDED BASIC try this without using any spaces:

Type a line number such as 100

Type !

Type CTRL D (puts nothing on screen).

Type X=0

Type CTRL O (puts nothing on screen)

Type CTRL G (puts nothing on screen)

Type 1000

Press ENTER

Type LIST and see:

"100 ! IF X=0 THEN GOSUB 1000"

After typing in your program, LIST the program and delete the "!" from each line. This technique has been published before, but I have never seen a complete list of the commands and symbols the technique will produce. Below is such a list. Most of these key combinations also work in TI BASIC, a fact not noted before. From TI BASIC without extra spaces:

Type a line number such as 100

Type REM

Type CTRL G (puts nothing on screen)

Type 1000

Press ENTER

Type LIST and see:

"100 REM GOSUB 1000"

This list includes both the CTRL and FCTN keys. Most BASIC commands are accessed using the CTRL key, although a few show up with the FCTN key as well. Leaving the ALPHA LOCK key up or pressing the SHIFT key has no effect on this list.

CTRL Produces	FCTN Produces
A	ELSE
B	:: (XOnly)
C	!
D	IF
E	GO
F	GOTO
G	GOSUB
H	RETURN
I	DEF
J	DIM
K	END
L	FOR
M	LET
N	BREAK
O	UNBREAK
P	TRACE
Q	UNTRACE
R	INPUT
S	DATA
T	RESTORE
U	RANDOMIZE
V	NEXT
W	READ
X	STOP
Y	DELETE
Z	REM
0	THEN
1	TO
2	STEP
3	,
4	;
5	:
6)
7	(
8	OPTION
9	OPEN
+	CALL
-	AND
:	PRINT
>	ON
<	garbage
A	!
B	-
C	,
D	CURSOR RT
E	SCROLL DN
F	{
G	}
H	<
I	?
J	>
K	+
L	-
M	*
N	/
O	,
P	"
Q	^
R	[
S	CURSOR LT
T]
U	_
V	!
W	~
X	SCROLL UN
Y	garbage
Z	\
0	CURSOR RT
1	Insert
2	Delete
3	Erase
4	Clear
5	nothing
6	nothing
7	nothing
8	Redo
9	nothing
+	Quit
-	OR { XB }
:	NOT{only}
>	garbage
<	&

A REVIEW OF ORPHAN CHRONICLES AND THE
MYARC DISK CONTROLLER

BY TONY MCGOVERN, H. V. 99ERS

This is reprinted from the newsletter of the Hunter Valley (Australia) User Group. The author also is co-author of the fairware program FUNLWRITER

The time has come once more to put fingers to keyboard and do something for the HV99 Newsletter. I am of the opinion that monthly issue is just too frequent for a small group to support and will inevitably lead to too much dilution of local content. That can be fixed by having more local contributions too. Anyway for this edition, apart from the general gossip and progress reports we will have a review of Orphan Chronicles by Ron Albright and first impressions of the Myarc disk controller card.

On to the general gossip first. Funlwriter Vn 3.3 is progressing very satisfactorily and I am even writing some new document files to go with it. The group members going to Melbourne for TI-Fair will after all have something to take with them, unless of course Will uncovers some previously unsuspected fatal bug a day before they leave. Vn 3.1 of DM1000 was received from Bob Boone in Ottawa and the updated MGR1/2 have been interfaced to F'Wr Vn 3.2 to make an interim update until Vn 3.3 is released. Clint Pulley has sent the latest release of his small-c compiler with additional graphics, floating point ... libraries contributed from other Canadian sources. The Province of Ontario is a real fount of top grade 'fairware' material. Edgar Dohmann from JSC/UG near Houston tells me that a new version of Superbug II is on the way soon also.

5
Clint has made REL2 of c99 compatible with XB loader programs, and we have taken it a step further and made it re-entrant to F'Wr with filename transfer back to the Editor. Vn 3.3 will also allow return to F'Wr from DM1000 without going through the title screen. We have also closed the circle and made a version that will load from the utility option of TI-Writer. Really had to dig to find the module to test that one out. Lots more improvements go to make up a major update, but time enough to see those when it is released. Enough maybe to mention that a disk directory is available from a press of AID.

Now to the Orphan Chronicles, published by Miller's Graphics. The copy reviewed was on a round robin to gather an inscribed record of its travels. Thanks Dick, thanks Rex. Before anything else let me say that it is a thoroughly absorbing read for anyone who has spent a lot of time wrestling with the orphan. The book contains a measure of prehistory, and a description of the rise and fall, but most of it is oriented to the ongoing support in programming and hardware that will keep the machine viable in the near future even without TI. There is much information on the great growth in bulletin board and areas on commercial database systems. Unfortunately in Australia we are isolated by the tyranny of telecom costs and distance from the US networks. I would have liked to see more discussion of the engineering and corporate decisions and battles that went into making the machine the flawed gem that it is. But then again a writeup like that might not stir the same interest in readers without an engineering background. A question implicitly posed that is not answered and probably never can be, is whether the right microprocessor was chosen. I think it was, and that it is the TMS9900 processor that gives the machine its continuing life after death. A Z80 machine would have been cheaper for TI to build, though this is confused by the transfer pricing games that went on within TI over the 9900, but I can't help feeling that a Z80 based machine would have

no more continuing fascination than a Dick Smith VZ200. More like MSX which hasn't really turned people on either. As Ron Albright indicates, the real tragedy is that TI lost their nerve after the 99/4a and didn't continue with the 99/8. There must have been a lot of people like us who held back from buying a 99/4a expansion system while waiting for the upgrade of the whole system. The author is altogether too kind to some of the characters who floated to the surface - the pompous and vacuous editor of a magazine of very little content and the most amazing disappearing acts, a shonky software dealer masquerading as a user group, and so on. I do note with approval, as do the rest of your committee, that in the Appendix listing User Groups around the world the HV99 group is the very first mentioned.

We now have a Myarc disk controller operating in our PE box thanks to appreciation of F'Wr in the US of A. If Ron K's 3 holer mini PE box project comes good we may be able to get a second system going as well. The following discussion will be of first impressions only. The card gives double density operation of disk drives and whatever faults it may have, seeing 1433 sectors free on a disk more than makes up for them. We thought we were in hog heaven when we installed the Chinon DS drives in the PE box a few months ago, but DD is even nicer still. The comparison of interest is with the Corcomp card. I have not heard any bad news about reliability of Myarc cards, but local experience with Corcomp has not been good. On the other hand reports from the US are that Corcomp have solved many of their problems, but that may not help too much if you have a lemon. One thing they do have in common that they can take advantage of faster track stepping rates of newer disk drives with a permanent switch setting. We have been so busy with F'Wr that we haven't had time to check it out on the Chinon F502 drives yet. Will says he tried 6ms in software on the TI original Shugarts from the TI disk card and they didn't like it one little bit, but that was to be expected.

The Myarc comes with a manager program on disk which works very well and is easy to use. Some other utilities are provided but we have not had occasion to use them yet. One could say that the original TI controller card does the job that it was designed to do to perfection even if with limited performance. The Myarc card by handling double density does more, but there are some catches apparent that make it fall short of the TI standard of engineering.

The first thing you notice is that you don't notice anything. I like this much better than the way the Corcomp captures the machine on power up. As a matter of fact it has no power up routine at all, and unlike with the TI card the drives don't spin briefly on return to the title screen. That's fine if it never needs one, but I have a suspicion that on occasion it does - not in normal operation but when something has gone very, very wrong, a return to the title screen is not always enough to clear things up and the whole machine, PE box included must be powered down. That's more likely to happen around our place where Will has spent a lot of effort mastering the art of talking directly to disk controller chips, than it is in a more normal household. Another problem, which I have also not yet investigated fully, is a propensity to blow holes in the directory. This might occur if an assembly is aborted with closing the object file. With the TI card this results a nuisance Dis/Fix 0 file entry which must be deleted before re-assembly to the same file name. This happened with the Myarc and I thought at first that I had lost half the source files of F'Wr. None of SD, DM2 module or DM1000 could find them, but the Myarc DM showed them after a hole in the directory list and would copy them to another disk. Presumably the Myarc directory builder doesn't stop at an imbedded null entry on sector #1 unlike most programs which work fine with the TI card. If the Myarc card really does have this nasty habit, then it is essential to use the Myarc DM with it. A definite bug in the Myarc card is that it doesn't support all Status calls correctly. F'Wr Vn 3.0-2 will not assemble to

a pre-existing object file if the Myarc is used; and the hangup is in the Status calls made by F'Wr before it actually opens the file. A real "Wellington lady" of a program, it checks the status of files before it will associate with them.

Whereas the Corcomp card seems to use VDP RAM in much the same way as the TI original, the Myarc after setting aside VDP buffer space for compatibility with other software then proceeds to ignore it and uses its own RAM in the card. I have not been able yet to do any comparative timing tests between Myarc and Corcomp, so I don't know whether this speeds things up any. It still has to get things into VDP where normal software expects them, but it doesn't have to do it doubly. The Myarc's use of its own RAM does lead to some incompatibility with some important software, as Myarc have not bothered to adhere to that part of the TI disk system specification that calls for the last disk name to be stored near the top of VDP RAM. The latest version (3.1) of DM1000 bombs out completely with Myarc for this reason, and the boot disk tracking feature of F'Wr Vn 3.0-2 doesn't work. The soon to be released Vn 3.3 of F'Wr takes account of the quirks of Myarc for its own account, and loads DM1000 as well. This problem afflicts some RAMdisk DSRs too. From Bernie Elsner's last call from Perth I know that they have run into the same problem with the Horizon RAMdisk, a device which has not come to Newcastle yet, and are going to do something about it there. The Myarc has another quirk that renders DM1000 less than completely useful. This would appear to be a hangover from when the design was frozen, when only 16 sectors per track were written in double density (I seem to recall that the first IBM PCs came out with 320K disks compared with the later and now standard 360K). Unfortunately 16 sectors per track is built into the Myarc DSR ROM in sub-program >11, the formatting routine. Programs such as DM1000, TI-Forth, the DM2 module which call this routine end up with only 16 sectors per track for a total of 1280 sectors as against the now normal 1440. That's OK by disk

7
DSRs until you run into software that assumes 1440 sectors for DSDD such as disk sector copy programs - DM1000 for instance. Myarc get around this in a fashion that is only partially satisfactory in use by having 18 sector initializing done by the manager program from disk. I consider it reprehensible that Myarc has not by this late stage updated the card firmware or offered a replacement EPROM. Will is looking into the possibility of patching the Myarc DSR code so that 18 sector initialization is obtained, and maybe if that is possible we may go to the trouble of blowing a replacement EPROM. If any of our interstate or overseas readers know of this having been done already, we would appreciate hearing about it, or even better a sample of a revised EPROM.

The Myarc card also offers some new CALLs available from command mode from XB or also as statements from Basic even with no module present. They were probably a good idea at the time but the coming of Funlwriter has made most of them largely obsolete, and they would be better replaced by more powerful disk related functions. The one that is useful is CALL DIR(x) to get an instant disk directory with no module insertion or program loading needed. The others are a set of CALLs, ILR, LR, LLR that replace the INIT, LOAD, LINK from E/A Basic. There are limitations, some of them surprising, for instance the E/A DEBUG which is a program which is as transparent as possible returns to a locked up computer, either from a Basic program or from command mode. Because they are invoked from Basic, even in the command mode, programs are faced with VDP register settings and table locations appropriate for Basic, so only programs designed for this will give proper screen display. DEBUG has this flexibility but most utilities for E/A, as distinct from E/A Basic, don't.

The utilities loaded by ILR are similar in function to those loaded by E/A but occupy a little more of low-mem and the REF/DEF table entries point to different locations. This is no problem for programs which REF the utilities

but means that the ILR utilities are not usable by programs that rely on the E/A utilities being preloaded for cold starting of memory image files. One pleasant surprise is that GPLLNK works even in a bare console. This is probably worth discussing for its tutorial value while I am at it. As you know the bare console contains no mechanism for escaping from the clutches of GPL out into the 32K RAM expansion. Programs coming in from the 32K have more of a chance to fiddle conditions so that if they call a GPL routine, they can regain control from GPL. The point of the GPLLNK routine is to trick GPL into handing back control. As discussed in the E/A manual the GPL routines that can be used are those which end in a GPL return instruction (and otherwise do something useful for you) so that GPL goes back to the stack in console scratchpad RAM to retrieve the next address it should branch to. The GPLLNK tricks GPL by first pushing the address of a suitable GPL return instruction on to the stack before handing over, so that when GPL has done the routine called it hands back to GPLLNK which then returns to the calling program after resetting the GROM address to where it was to start with. Modules such as E/A and XB do have suitable exits built which GPL uses in CALL LINK or CALL LOAD (not XB) or from the E/A screen. GPL hands over to machine code when it encounters an XML (eXecute Machine Language) instruction (byte >0F) followed by a byte pointing to a standard table entry. These tables are partially listed in E/A or Minimem manuals. Most of the XML instructions in GPL point to console or cartridge ROM locations which is no good for returning to a program in RAM. So how to do it from the bare console? It turns out (though TI never saw fit to mention it explicitly) that there is one console GPL routine which does hand over to RAM. This is the GPL bit reversal routine in GROM 0 which unloads some machine code into the scratchpad RAM at >8300 and then transfers control to it at >8300 (>F0 table entry) to execute before returning to GPL. Since at least 8 versions of GROM 0 have been sighted, a program cannot rely on this being at a fixed location but has to search for it. There is

another possible exit point in GROM 0, but this is accidental in the sense that data bytes happen to form a XML instruction into the scratchpad RAM at an inconvenient location. Myarc uses this console GROM return point, doing the search every time.

We should mention in passing that E/A's GPLLNK is partially realized in GPL in the module and doesn't look like our description above. Trouble is it has a bug and does not always work. When F'Wr is simulating E/A the GPLLNK always works, thanks to a mixture of brute force and low cunning.

All in all I don't think the Myarc extra CALLs (other than DIR) are particularly useful or well chosen. After all if you have the system and resources to afford a Myarc card you probably have E/A or XB anyway. Now F'Wr has rendered it completely obsolete in this regard. I think the Corcomp designers' decision to include a program file loader shows more design sense, but I'm not instantly familiar with the details. The extra CALLs in both of them mean that programs which use them lose a lot of portability because neither disk card is universally used.

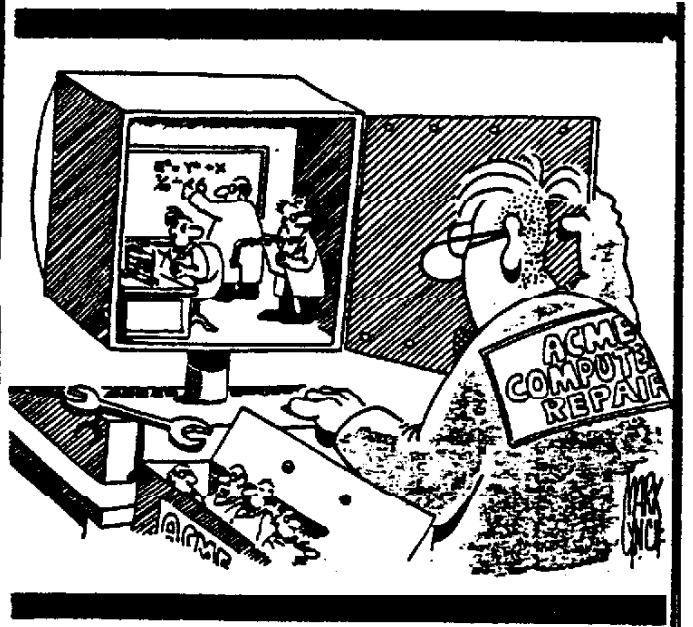
The Myarc controller uses a Western Digital 1770 controller chip and has 2K of RAM and 8K of ROM. How does that fit in the 8K DSR space allocation? The 2K RAM is located at >5000 and the ROM is paged in two 4K halves by CRU bit addressing, each located at >4000 - >5000. Will found that paging in the second half of the ROM is done by a cute piece of code which loads workspace registers with the transfer code, then does a direct branch to the first register of the block just loaded so that it is now executing out of RAM in its own workspace, and this code pages the ROM, sets up the new branch address from a branch table in the newly paged ROM, and finally does an indirect branch to it. That illustrates why the 99/4a survives. Not only is 9900 code elegantly simple but also it allows you to do the most amazing things.

After all that you might get the impression that we're less than entranced with the Myarc. Well,

that just ain't so. It's just fantastic to have all that disk capacity open up, and it just sits there and does its job. No problems with not recognizing drives that you know are there. The only real difficulty is the formatting problem, and who knows - maybe that's fixable too. Vn 3.3 of F'Wr has been partially re-written to accomodate the Myarc controller with no loss of function. I am debating with myself whether to alter the DM1000 files included on the F'Wr disk so that they bypass or refuse to format disks in double density on a Myarc controller, requiring pre-formatted disks for disk copies. Of course if you have to load the Myarc manager program to format a disk, you might as well continue to use it to do the disk cloning.

A request that I would like to pass on is from Bonnie Snyder, President of the Front Range 99er Computer Club in Colorado Springs, who is looking for programs, particularly those using speech that would be suitable for use at a school for the blind. If you have such programs I can forward them or put you in touch for direct exchanges.

Tony McGovern
Funnelweb Farm
May/86



TIPS FROM THE TIGERCUB

#38

Copyright 1986

TIGERCUB SOFTWARE
156 Collingwood Ave.
Columbus, OH 43213

Distributed by Tigercub Software to TI-99/4A Users Groups for promotional purposes and in exchange for their newsletters. May be reprinted by non-profit users groups, with credit to Tigercub Software.

Over 138 original programs in Basic and Extended Basic, available on cassette or disk, only \$3.88 each plus \$1.58 per order for PPM. Entertainment, education, programmer's utilities. Descriptive catalog \$1.88, deductible from your first order.

Tips from The Tigercub, a full disk containing the complete contents of this newsletter Nos. 1 through 14, 56 original programs and files, just \$15 postpaid.

Tips from the Tigercub Vol. 2, another diskfull, complete contents of Nos. 15 through 24, over 60 files and programs, also just \$15 postpaid.

* Tips from the Tigercub *
* Vol. 3 is now ready. *
* Another 62 programs, *
* routines, tips, tricks. *
* from Nos. 25 thru 32. *
* Also \$15 postpaid. Any *
* two Tips disks \$27 or *
* all 3 for \$35 postpaid. *
*

Nuts & Bolts (No. 1), a full disk of 188 Extended Basic utility subprograms in merge format, ready to merge into your own programs. Plus the Tigercub Menuloader, a tutorial on using subprograms,

and 5 pages of documentation with an example of the use of each subprogram. All for just \$19.95 postpaid.

Nuts & Bolts No. 2, another full disk of 188 utility subprograms in merge format, all new and fully compatible with the last, and with 18 pages of documentation and examples. Also \$19.95 postpaid, or both Nuts Bolts disks for \$37 postpaid.

Tigercub Full Disk Collections, just \$12 postpaid! Each of these contains either 5 or 6 of my regular \$3 catalog programs, and the remaining disk space has been filled with some of the best public domain programs of the same category. I am NOT selling public domain programs - my own programs on these disks are greatly discounted from their usual price, and the public domain is a FREE bonus!

TIGERCUB'S BEST, PROGRAM-TUTOR, PROGRAMMER'S UTILITIES, BRAIN GAMES, BRAIN TEASERS, BRAIN BUSTERS!, MANEUVERING GAMES, ACTION REFLEX AND CONCENTRATION, TWO-PLAYER GAMES, KID'S GAMES, MORE GAMES, WORD GAMES, ELEMENTARY MATH, MIDDLE/HIGH SCHOOL MATH, VOCABULARY AND READING, MUSICAL EDUCATION, KALEIDOSCOPIES AND DISPLAYS

For descriptions of these send a dollar for my catalog!

I have discovered a rare bug in the 28-Coluan Converter, published in Tips #18, which will cause an I/O 25 ERROR if the very last line of the program being converted happens to have exactly 88 characters. You can fix it by adding a line -
215 IF EOF(1)=1 THEN 268

There is also a rare bug in the SIDWAYS subroutine on my Nuts & Bolts #2 disk, which prevents turning some

redefined character sets sideways. If you are one of those who BOUGHT that disk from me, you can fix it by changing the L=LEN(B#) in line 21639 to L=64.

I was in too much of a hurry to go fishing when I put the last couple of Tips together. In the Gordian Knot in Tips #35, I left out some essential instructions. Please add -
131 DISPLAY AT(11,1):" When you cross your track," : "press S to go over, U to go under, C to go across."

To make that fit, you will have to change the DISPLAY AT in line 138 to (8,1), in line 148 to (15,1) and in line 158 to (28,1), also the ACCEPT At in 168 to (28,11). And this change will prevent a lockup when you reach a border -
288 D=D-1 :: IF ABS(D-D2)=2 OR R+(D=1)=8 OR R-(D=3)=25 OR C+(D=4)=2 OR C-(D=2)=31 THEN 188 :: GOSUB 518 :: IF D<>D2 THEN GOSUB 458

I wrote the dulcimer music in Tips #36 in Basic, but I forgot to test it in Basic. It actually runs much better in Extended Basic, but will run fairly well in Basic if you delete the delays in lines 288 and 388.

If you liked the ESCHER ART in Tips #37, these modifications will improve it considerably -

```
118 DISPLAY AT(12,1):"Press
-": " D for new pattern":
B to change background": " F
to change foreground": " R to
reverse colors": " : "Any key
to start"
288 A=INT(6*RND+3):: H=INT(2
4/A):: RX=24-H*A :: HC=INT(2
B/A):: CX=28-HC*A :: W=ABS(H
C/2=INT(HC/2))-(RX)#8):: DIM
M(8,8):: FOR P=1 TO A
338 IF K<>66 THEN 346
348 BC=BC+1+(BC=16)#15 :: IF
BC=F THEN 348 ELSE 347
```

```
346 IF K<>78 THEN 368 :: F=F
+1+(F=16)#15 :: IF F=BC THEN
346
347 FOR S=7 TO 14 :: CALL CO
LOR(G,F,BC):: NEXT S :: GOTO
318
```

```
358 ! **DELETED LINE **
368 IF K<>ASC("R")THEN 318 :
: T=F :: F=BC :: BC=T :: GOTO
0 347
688 GOSUB 988 :: FOR T=1 TO
A :: DISPLAY AT(R-1+T,C):M#(
V,T):: NEXT T :: NEXT C
681 IF CX>8 THEN AA=A :: GOS
UB 888
685 GOSUB 1888 :: NEXT R
686 IF RX=8 THEN 618
687 GOSUB 1888 :: FOR C=1 TO
A#HC STEP A :: GOSUB 988 ::
FOR T=1 TO RX :: DISPLAY AT
(R-1+T,C):M#(V,T):: NEXT T :
: NEXT C
688 IF CX>8 THEN AA=RX :: GO
SUB 888
888 GOSUB 988 :: FOR T=1 TO
AA :: DISPLAY AT(R-1+T,C):SE
6#(M#(V,T),1,CX):: NEXT T :
: RETURN
988 V=V+1+(V=4)#4 :: RETURN
1888 V=V+W :: V=V+(V>4)#4 ::
RETURN
```

I had a letter from a teacher who was using the PRK module to keep student grades, and wanted to know how to average them. It can be done, but is so impractical that I wrote this program. While I was at it, I speeded up the loading and saving to cassette greatly by converting the grades to an ASCII string and combining the student's name and all grades into one record.

```
188 DIM N$(58),T(58,28)
118 CALL CLEAR
128 PRINT " TEACHER'S
HELPER": : :
138 REM - by Jim Peterson
148 PRINT "(1)CREATE A FILE?
": "(2)ADD TO FILE?": "(3)LOAD
A FILE?": "(4)SAVE A FILE?":
"(5)PRINT A FILE?"
158 PRINT "(6)CORRECT A FILE
?": "(7)COMPUTE AVERAGES?": "(
8)QUIT?"
168 CALL KEY(8,K,S)
```

```

170 IF (S=0)+(K<49)+(K>50)TH
EN 160
180 ON K-48 GOTO 190,250,610
,800,380,990,1120,1510
190 X=0
200 INPUT "SUBJECT? "I$
210 GOSUB 1370
220 INPUT "TEST #? "I
230 GOSUB 1440
240 GOTO 140
250 PRINT ;;:"(1)ADD NAMES?"
;"(2)ADD GRADES?"
260 CALL KEY(I,K,S)
270 IF (S=0)+(K<49)+(K>50)TH
EN 260
280 ON K-48 GOTO 290,310
290 GOSUB 1370
300 GOTO 140
310 INPUT "TEST #? "I
320 IF T(I,0)=0 THEN 350
330 PRINT ;;:"TEST #";STR$(0
);" ALREADY RECORDED"
340 GOTO 140
350 N=0
360 GOSUB 1440
370 GOTO 140
380 CALL CLEAR
390 PRINT "OUTPUT TO:"(1)SC
REEN?;"(2)PRINTER?"
400 CALL KEY(I,K,S)
410 IF (S=0)+(K<49)+(K>50)TH
EN 400
420 IF K=49 THEN 460
430 INPUT "PRINTER DESIGNATI
ON? "I$
440 OPEN #2:P$
450 F0=2
460 PRINT "PRESS ANY KEY TO
PAUSE":
470 PRINT #F0:S0:
480 FOR J=1 TO X
490 PRINT #F0:"";N$(J)&" ";T
AB(I0);
500 FOR K=1 TO HN
510 PRINT #F0:T(J,K);
520 NEXT K
530 CALL KEY(I,K,S)
540 IF S<>0 THEN 530
550 NEXT J
560 PRINT #F0
570 IF F0=0 THEN 140
580 F0=0
590 CLOSE #2
600 GOTO 140
610 PRINT ;;:"(1)CASSETTE?";
(2)DISK?"
620 CALL KEY(I,K,S)
630 IF (S=0)+(K<49)+(K>50)TH
EN 620
640 ON K-48 GOTO 650,670

```

```

650 OPEN #2:"CS1",INPUT ,FIX
ED
660 GOTO 690
670 INPUT "FILENAME? DSK":F$
680 OPEN #2:"DSK"&F$,INPUT
690 INPUT #2:Y,HN,S$
700 FOR J=1 TO X
710 INPUT #2:K$
720 N$(J)=SEG$(K$,1,POS(K$,C
HR$(255),1)-1)
730 K$=SEG$(K$,POS(K$,CHR$(2
55),1)+1,255)
740 FOR K=1 TO HN
750 T(J,K)=ASC(SEG$(K$,K,1))
-50
760 NEXT K
770 NEXT J
780 CLOSE #2
790 GOTO 140
800 PRINT ;;:"(1)CASSETTE?";
(2)DISK?"
810 CALL KEY(I,K,S)
820 IF (S=0)+(K<49)+(K>50)TH
EN 810
830 ON K-48 GOTO 840,860
840 OPEN #2:"CS1",OUTPUT,FIX
ED
850 GOTO 880
860 INPUT "FILENAME? DSK":F$
870 OPEN #2:"DSK"&F$,OUTPUT
880 PRINT #2:Y:HN:S$
890 FOR J=1 TO X
900 K$=""
910 FOR K=1 TO HN
920 K$=K$&CHR$(T(J,K)+50)
930 NEXT K
940 PRINT #2:N$(J)&CHR$(255)
&K$
950 K$=""
960 NEXT J
970 CLOSE #2
980 GOTO 140
990 CALL CLEAR
1000 INPUT "STUDENT'S NAME?
":Q$
1010 FOR J=1 TO X
1020 IF N$(J)=Q$ THEN 1060
1030 NEXT J
1040 PRINT ;;:"NAME NOT FOUN
D":
1050 GOTO 140
1060 INPUT "CORRECT WHICH TE
ST? (0 TO QUIT) "I:C
1070 IF C=0 THEN 1110
1080 PRINT ;;:N$(J);"'S TEST
#";STR$(T(J,C));
1090 INPUT "CORRECT TO? "I:T
(J,C)
1100 GOTO 1060
1110 GOTO 140

```

```

1120 CALL CLEAR
1130 PRINT "OUTPUT TO:"(1)S
CREEN?;"(2)PRINTER?"
1140 CALL KEY(I,K,S)
1150 IF (S=0)+(K<49)+(K>50)T
HEN 1140
1160 IF K=49 THEN 1200
1170 INPUT "PRINTER DESIGNAT
ION? "I$
1180 OPEN #2:P$
1190 F0=2
1200 PRINT #F0:S$
1210 FOR J=1 TO X
1220 PRINT #F0:N$(J);" AVERA
GE ";
1230 FOR K=1 TO HN
1240 TT=TT+(J,K)
1250 NEXT K
1260 AV=TT/HN
1270 TAV=TAV+AV
1280 PRINT #F0:AV
1290 TT=0
1300 NEXT J
1310 PRINT #F0:"CLASS AVERAG
E ";TAV/X
1320 TAV=0
1330 IF F0=0 THEN 1360
1340 F0=0
1350 CLOSE #2
1360 GOTO 140
1370 PRINT ;;:"STUDENT'S NAM
ES - "I$type END when finish
ed":
1380 X=X+1
1390 M$="NAME #";STR$(X)&" "
1400 INPUT M$:N$(X)
1410 IF N$(X)<>"END" THEN 13
80
1420 X=X-1
1430 RETURN
1440 FOR J=1 TO X
1450 M$=N$(J)&"'S GRADE? "
1460 INPUT M$:T(J,N)
1470 NEXT J
1480 IF N$HN THEN 240
1490 HN=0
1500 RETURN
1510 END

```

The reason that 50 is added to the value in line 920, before saving, and subtracted again in line 750 after loading, is because of a quirk of the computer that I don't recall seeing in print anywhere. Did you know that INPUT will read a string beginning with ASCII 0, 2, 4, 7, 10, 12, 14, 18,

20, 26, 27, 31, 32, or 44 as a null string (a blank), and will drop these characters at the end of a string? And ASCII 32 will be dropped at the beginning or end of a string. And ASCII 0 within a string, or ASCII 34 anywhere, will crash, while ASCII 44 within a string will lose the rest of the string. I should have known what ASCII 0, 32 (the space), 34 (quotes) and 44 (comma) would do, but why the others?

LINPUT will accept anything, of course, but I wanted to keep this in BASIC for the teachers who are struggling along without the XBasic module or disk drive.

Chick De Marti published in LA 99ers TOPICS the surprising discovery that PRINT USING and DISPLAY USING can read the IMAGE format from a variable, array or string!

Which led me to some fooling around -

```

100 !PRINT USING DENO by Jim
Peterson, based on a discov
ery by Chick De Marti
110 CALL CLEAR :: RANDOMIZE
:: CALL SCREEN(5):: FOR S=2
TO 14 :: CALL COLOR(S,S,S)::
NEXT S
120 N=INT(13#RND+1):: C$=CHR
$(8#N+32-(N=4)*11)
130 FOR J=N TO 12 :: A$=RPT$(
" ",J)&"#&RPT$(" ",26-J*2)
&"# " :: PRINT USING A$:C$,C$
:: NEXT J
140 FOR J=12 TO N STEP -1 ::
A$=RPT$(" ",J)&"#&RPT$("
",26-J*2)&"# " :: PRINT USING
A$:C$,C$ :: NEXT J :: GOTO 1
20

```

Here is one last Tigercub challenge. What is the longest possible one-liner? And what is the longest possible one-liner that actually does something?

MEMORY FULL

Jim Peterson

