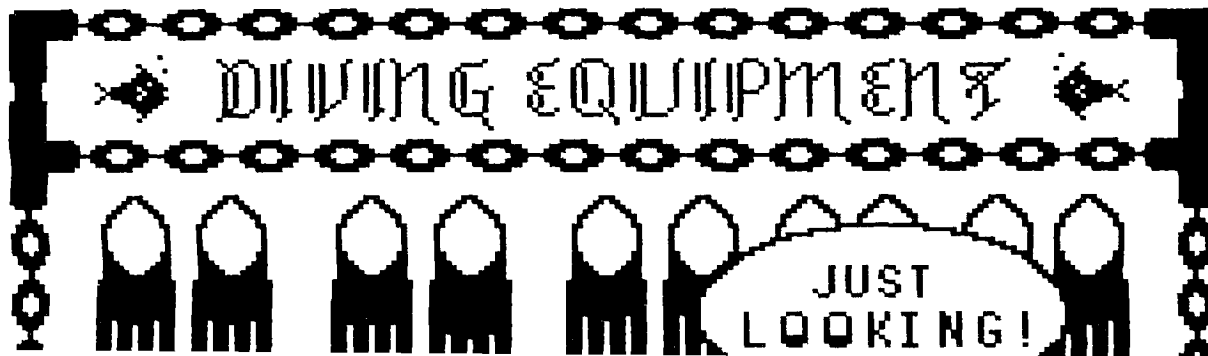




The Ottawa T.I.99/4A Users' Group



VOLUME 8 NUMBER 7.....September 1989



DON'T FORGET THE MEETING -- September 12, 1989

*** ONTARIO, CANADA K1P 5W3 ***

COMING EVENTS

September Meeting: September 12, 1989 Merivale High School
7:30 p.m.

October Meeting: October 3, 1989 Merivale High School
7:30 p.m.

5th Annual TI-FEST April (?) 1990 Merivale High School
Contact Ruth O'Neill
for details or to
volunteer your help.

Newsletter Deadline: September 20, 1989 (or any time before that!)

EDITOR'S NOTES from Ruth O'Neill

Well, this makes 19 down, and one to go. Since I announced in the June issue that I would be stepping down as editor after the October issue was in the mail, I have received exactly no calls from any prospective editor. To the best of my knowledge, no one has contacted the executive, either. Is there anyone out there who is interested, but afraid to step forward? Please call! I was terribly unsure of myself when I decided to ask for this job, but I soon found that I wasn't on my own; there are many people you can turn to for support. They're only too glad to help. I promise to help you out by writing for you, too. And remember, if you are even slightly interested in the job, ask some questions about it! Nobody will force you to take it if you decide against it. This is a volunteer position, after all, and I wouldn't have been doing it this long if I didn't think it was a terrific one. It has been a wonderful experience, and you should give it a try, or at the very least, check it out.

In case anyone is wondering what I'm going to do with my new spare time, I've found a new job in the club. Meet your new co-ordinator for the 1990 Canadian TI-Fest! No date has been set as yet, but be sure that I will keep you posted. Meanwhile, you can reach me these ways:

Mail: Ruth O'Neill
34 McLeod Street
Ottawa, Ontario
Canada
K2P 0Z5

Phone: (613) 234-8050 (That is my home number; please leave a message on the machine during working hours.)

Delphi: REON

Compuserve: 72117,3541

This issue features two members contributing to the newsletter for the first time (at least since I've been here): Jan Czaban to report on his trip to a British TI fair, and Ken Burchill, with a review on HyperCopy. Geneve owners are not the only ones who will be interested in the latter article, by the way; the information about disks should be of interest to many others as well. Oh, yes! Dave Morrison's regular library column isn't in this issue because he was remarkably short on material to write about. Please help him out!

Our Sysop, Peter Arpin, has a message up on the board for those who might wonder why TEXTLINK has been shut down occasionally on Friday evenings recently. Apparently Lloyd Galenzoski has been working on an assembly-language file

manager for the personality card. They have been testing it, and it looks as if it will make managing the BBS much easier for Peter. Thank you, Lloyd!

Although the summer is usually a fairly quiet time in the TI world, there have been a few things happening -- some good, some bad. Examples of good news are Asgard Software's releases of Page Pro (see Lucie's review this issue) and Legends 2: The Sequel, which I thoroughly enjoyed (I've been a Legends fan since it was released, and the sequel was welcome news.)

On the sad side, the TI SIG on the Source is no more. That's because the Source itself is no more, and Source members have been given memberships on CompuServe. Some people on Delphi have been wondering if anyone captured the PARTI thread on the TI in its entirety before it disappeared. If you know of anyone who has, please contact me, and I will pass the word along. And while we're on the subject of bad news, please read the following.

Write to Computer Shopper!

Those of you who read the Computer Shopper's TI Forum, a regular column by Jonathan Zittrain and Barry Traver, might have been surprised to see that it was missing in the September, 1989 issue. While this worried a number of people, Barry Traver was quick to explain that the column has not been cancelled. Due to a change in management at Computer Shopper, column deadlines were changed and the TI Forum just didn't get in in time. There is no cause for alarm. Or is there?

While the new management has said that they have no plans to cancel any of the "Classic Computer" columns at this time, Stan Veit, the former editor of Computer Shopper, suggests that if we want to make sure the TI column continues, we should write now. What if the new editor feels that there are not enough readers to justify continuing the column? Or, more importantly, what if he wonders whether it generates any advertising revenue? He really has no way of knowing the true situation unless we write and tell him about it.

The truth is, there are a lot of us who buy Computer Shopper every month just for the TI column. Many of us have other computers as well, so we read other sections of the magazine, too. BUT we wouldn't bother buying it without the TI Forum. We also have friends who have other computers, and pass word on to them of any especially good buys we spot, so we support their advertisers that way. Most importantly, though, because the TI uses so much equipment (floppy disk drives, hard disks, printers, monitors, modems...) that is marketed for other machines, we are interested in many of their advertisers, not just those who might appear in our special section. Most people outside the TI community just aren't aware of that, so we need to tell them.

It isn't enough for one of us to send an informative letter, though. It's too easy to set that aside. We need to make letters pile up on the desks of Computer Shopper staff -- let them know that we are an important market for them, that they NEED us, because we need them, too. The more letters, the better. Write and tell them how important the TI Forum is to you, and how much you missed it in the September issue. Tell them about all the equipment that their advertisers have that we still buy. Write even if you've never read Computer Shopper before, because the TI community needs all the exposure we can get in more general publications like Computer Shopper. Even if you don't read the column, you benefit indirectly by the fact that others read it, and discover that our community is still alive and kicking.

Write today!!!! Get your TI friends to write, too!!!

Letters should be addressed to:

Bob Linstrom, Editor-in-Chief
Computer Shopper
1 Park Avenue
New York, NY 10016

The President's Two Cents' Worth
by Jane Laflamme

Can you believe it? With the coming of this Newsletter, it means that the summer is all but over. Where did it go? I hope you all had a wonderful summer and holidays; are well rested; and ready to plunge right into the computer world again.

There have been a few things happening over the holidays. We had a visit from the Editor of the New Jersey Newsletter, TI-dings. Henri Hein and his wife decided to take a T.I. vacation in Canada. Each town or city they visited, they asked for a contact name in the next place they were planning to go. I was the lucky name for Ottawa. As I was occupied the first night they phoned, Bill Sponchia went down and picked them up and entertained them in his home. He also took them on a short evening tour of the parliament buildings and the city. The next day, I took them on another tour of the city (I hadn't known at that time that Bill had already done so) including once again the parliament buildings! We had a nice long lunch and I returned them to their motel. What a great experience that was for us -- and what a novel idea...

A couple of weeks later Barry, Sharon, and John Traver visited the Laflamme household for a wonderful long weekend that included the changing of the guard, a membership drive at Carlingwood Mall [where several of our members turned up to meet Barry and family], a tour of the Museum of Civilization [including the film The First Emperor] and a lot of getting-to-know-each-other-and-T.I. talk. My family and I really enjoyed their visit and I was quite sad to wave good bye at the end [even though they called my collection of loons, DUCKS -- horrors!].

At the time of writing, we have held two of the three membership drives and both went very well. I would like to thank those members who turned up for them both, and of course the upcoming one at St. Laurent, to thank Marcelle Gibson for calling the malls and arranging, and Ralph Kuhn who organized the actual days there. The time goes by very fast when one has a chance to talk about our computer, plus we did find a few members to boot... not to mention a chance to eat out as well! Thanks again to all.

I feel sorry for those who didn't join us at the executive meeting at Moosehead Lodge. We had very good weather and a good turn out. Close to my heart is the enjoyment of nature. We saw quite a few loon "couples" with chicks on their backs, a beaver, and a blue heron just to name a few, but... you should have seen the size of the fish that you could have caught! We did a lot of canoeing, playing "liars dice" [a tradition now], and as usual, Mal and Mary Taggart fed us TOO well! Oh! Did I forget about the executive meeting? We had that too.

Bill Sponchia has been phoning the membership over the summer to try and raise volunteers for the executive, committees, and find a volunteer for Newsletter Editor. When I asked him how things were going, he told me that we are going to have a lot of volunteers "next year" as they don't have enough time this year.... I am beginning to wonder if we will have a club next year for all those people to volunteer for... We have several key positions open. I definitely will be stepping down as President. With that job and the FEST, I am beginning to flag; also, I'm finding it hard to keep up my enthusiasm after spending seven years actively participating in the club. I am beginning to sound like a broken record, but the club needs YOU -- and needs you THIS year. So if you've had a change of heart, please call Bill back -- now? Before you forget?

There have been a few things happening in the TI world over the summer, but why don't you come to the first meeting and find out about them? The meeting itself will be dedicated to the GENEVE 9640. We will be showing it with a hard disk, a 3 1/2" 1.44 meg floppy drive (these can also be used on a /4a!), plus as many programs and utilities as possible within our time limitations. Please note: The meeting is the SECOND Tuesday of September, September 12th, in the Cafeteria of Merivale High School in Nepean. Oh the whole, the balance of the year will be the first Tuesday, with a couple of exceptions. We will publish the remaining dates in next month's Newsletter. Starting time is 7:30 p.m. See you there...

'Till next month...

A VISIT TO THE UK TI USERS' GROUP ANNUAL MEETING AND FAIR
By Jan Czaban

This event was organized by Stephen Shaw and was held at Romley, a suburb of Manchester June 17. There were various presentations, raffles, demonstrations, and even an annual general meeting which took advantage of the good turn-out. I was in Edinburgh on business but was able to force myself away from my work that Saturday for a three-hour drive along winding roads in HOT weather.

I made Manchester in good time, but was unable to find Romley. British roads prove that the shortest distance between two points is definitely not a straight line. At last I found Romley but no sign of the fair. After several drives along the main road, the TI logo caught my eye, on the main door of the largest and newest building, the community centre, right at the town centre.

There were two large UK-based software distributors present offering new and used copies of virtually all existing TI and second party software at prices ranging between \$5 and \$20. Harrison Software from the US was also present.

Eight or so tables were set with expanded systems; two featured the Mechatronics 80-column card and a complete Geneve. There were several with 10 Meg drives using the Myarc HFDC and horizon Ram disks were also common. There was no shortage of latest software. I arrived well armed with latest examples of OTIUG software, copies of our newsletters and some personal favorites. Stephen Shaw was easy to locate, an extremely enthusiastic bundle of activity, and quickly introduced key persons from the other UK users groups. Turbo-copy was used for several hours running copies for User group libraries. All our material was "freely" distributed. This seemed an unusual practice, but was quickly taken advantage of. In return, Stephen provided copies of his UK library disks, which will be given to our librarian for further distribution.

The UK groups seem to have a very well organized software system. There was evidence of a good disk-of-the-month and library-loaner disk system. The loaders included with the programs were user friendly and well designed. Also a novel "understanding" regarding re-distribution of copyright material was in practice. It seems the UK groups have TI's and other software companies' permission to copy and distribute material. The following is an excerpt from one of the UK library disks:

"It is not intended to hurt the interests of the few commercial supporters left, and for this reason, as far as it is possible to check, no programs are included which are currently in commercial production in the US, the UK or Canada.";

"Present UK law makes ownership or copying of these disks unlawful, period. However, we have not been able to contact many of the copyright owners and it is not unreasonable to treat copyright of these files as effectively, even if not formally, abandoned. Please be aware however of possible legal liability. You are advised for instance not to place a full page ad in any Sunday newspaper..."

"Any interested copyright owner who may come across this disk is invited to negotiate copying rights, and will on application be supplied with details of all primary supply, together with an offer of royalty therefore;"
"(We have now purchased the rights to LEGALLY copy the following: TI Runner, Midnight Mason, Burger Builder, TI Toad, Micro Pinball. Space Station Pheta. Sort+Spell. Only the User Group is authorized to make copies- consent does not extend to individual members)"...

While these rules seem within reason and almost in tune with what our friendly RCMP spoke about at the last meeting it is nice to see an attempt at a defensible policy on this touchy subject.

There was, unfortunately no new UK-based software. A group representative from the Netherlands was present, however, with a Dutch Demo disk. This group seems to be producing new software. The disk includes high-resolution colour graphics, some games and utilities and will also be given to the library.

The fair attracted in the order of 150 visitors, although actual numbers were not available since the event was not ticketed and people drifted in and out during the day.

A FEW WORDS ON MUSIC-PRO FROM THE AUTHOR
By David Caron

Well it looks like I am back into writing articles again. Actually the only reason I stopped was because of the fact that I eventually ran out of material, but something new has come up. Recently Ruth O'Neill asked me to write an article about the planning and development of Music-Pro (That nice music program you have probably heard of by now.) I quickly realized that this would be a way of releasing all my frustration which had built up in the process of making the program, so I jumped at the chance.

I first thought of the idea of designing a program like Music-Pro several years ago, back when I had only a TI console, TI-Extended Basic and a TI standalone 32K memory expansion box. Before even this time, I had written a music program in TI-BASIC which I called Melody-Memory. However, this program represented notes the same way as they are seen on player piano rolls. It was difficult to lay down a melody, and although it could be saved to cassette, it was impossible to read or edit. Thus the urge was born to write something more "user friendly".

My first step in using real notes off a score sheet was deciding how to display them on the screen. Sprites seemed to be the perfect answer since there were 28 sprites available in Extended Basic, one sprite for every column on the screen. (The 2 extreme right and left columns were not visible on the old fully vacuum tube powered B/W TV set I used). And not only that, each sprite could easily represent any pitch simply by moving it up or down on the staff which would be displayed on the graphics screen. I eagerly raced ahead and defined all the notes (whole, half, quarter, etc..) plus some code which would read the keyboard and place the sprite on the staff where it should go. After debugging, I attempted to stick all 28 possible notes on the screen. Things went great with the first 4 notes, but for some queer reason the fifth note simply would not show up on the screen, unless it was at a different pitch than all of the previous notes (Call magnify was set to 3), and even then only part of the note would show up. This phenomenon left me confused and worried. (I had previously taken the computer apart just to see what was inside.) After several hours, I discovered that the 9918 VDP chip was incapable of displaying more than four sprites on the same line. I found this written right in the TI-extended Basic manual, but then, who would read a book from beginning to end in the form of a dictionary? So much for the easy way. The only option left was to use lots of CALL CHAR's.

Trying to display notes on the graphics screen proved to be much harder since one could not shift a note 4 pixels at a time to change pitch. (The lines in the treble and base staff were each 4 pixels apart). I plunged on ahead and created all the necessary characters needed to construct all the notes and their pitches. This turned out to be over 100 characters. The advantage to using characters was, of course, the fact that they could be displayed at different locations on the screen at the same time, allowing me to have not one but three pairs of treble and base staves. The disadvantage was the fact that there was were no characters left for text or even numbers. This made it difficult to test and keep track of user errors. Tracing was a total disaster, but then I noticed my speech synthesizer and figured it would understand characters sent to it, no matter what they looked like on the screen. This worked, although it was slow.

Each note displayed on the screen was represented by two or three characters. There were 33 pitches available plus 6 possible durations for each note. This meant that there were $33 \times 6 = 198$ possible ways of using the characters. Large two-dimensional arrays were needed to call up the notes from a key press. These arrays consumed large amounts of memory since the contents of the arrays were stored in the program code, and then they were copied into memory for variables during program execution. The call char's also ate considerable memory even when they were used with data and read statements. Then there was the code to read in keypresses, display the note on the screen, and play it back. By the time that inevitable "MEMORY FULL" error came, I discovered I could enter songs, so long as they were not more than 50 notes long!. Yes, I forgot to mention the arrays needed to store the notes, duration, frequency, and volume. (Storing the keypresses would not have worked since the song would not be played any faster than it was put in).

A FASTER FLOPPY FOR THE GENEVE

by Ken Burchill

So, you say it takes so long for GPL to load that you are able to read five or six chapters of War and Peace while you wait? If you are like me, then you have the need for speed. Unfortunately, there was nothing we could do about it until Mike Dodd came out with his HyperCopy. This program modifies the skew and interlace of a floppy disk to increase its speed. Before I describe how to use it, an explanation of the terms interlace and skew will help you understand how it works.

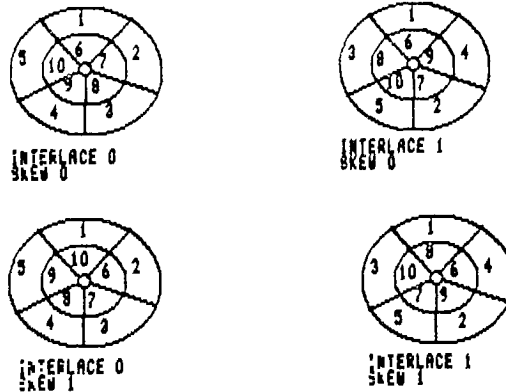
A floppy disk is organized into sectors of 256 bytes. When you load a program or file, the computer reads a sector, then stores it into a memory location, then reads the next sector, and stores that into a memory location, then reads the next sector, etc. In addition, the disk must keep spinning at a constant rate. In the time it takes the computer to store the sector read in, several sectors may have gone by. Hence, the computer has to wait for the desired sector to come back around. The ideal situation is to have the next sector at the position the disk head is at after it stores the last sector into memory. This can be achieved by putting the sectors out of sequence. For instance, assume there are only 5 sectors on a disk. If there is an interlace of zero, then the sectors would be in the following order: {1,2,3,4,5}. This would only be the ideal situation if the computer was fast enough to store the sector almost instantly, or so slow it took the time for five sectors to go by to store the sector. However, if the time it takes to load a sector into memory is the same as the time for the disk to spin past one sector, then an interlace of 1 is desirable {1,4,2,5,3}. For instance, assume a file to be read is on sectors 1 and 2. The computer would read the sector, load it into memory, and by the time it is ready to read in the next sector, sector 4 would have gone past and sector 2 would be right where the disk head is. Hence, interlace is the number of sectors each sector is out of sequence.

The idea of skew is similar. Skew is the number of sectors each track is out of sequence. A floppy disk is divided into concentric circles called tracks. Each track contains a certain number of sectors. If we had a disk with two tracks, with 5 sectors per track, and an interlace of zero and a skew of zero, the tracks would look like this: Track 1 = {1,2,3,4,5} Track 2 = {6,7,8,9,10}. This would be the ideal situation if your computer could load sectors into memory almost instantly and your drive's head could change tracks instantly. If we have an interlace of zero and a skew of 1, then the tracks look like this: Track 1 = {1,2,3,4,5} Track 2 = {10,6,7,8,9}. This would be the ideal situation if your computer could load sectors into memory almost instantly and it took your disk head the time for one sector to go past to change tracks. For instance, if a file were on sectors 5 and 6, the computer would load sector 5, change tracks, and by the time it is ready to load in another sector, sector 10 would have passed and sector 6 would be right where the disk head is. By combining the optimal interlace and skew, the time it takes to load in files can be decreased to the optimal level.

If you have trouble understanding the above, do not worry, because you do not have to know how Hypercopy works to benefit from it. Before you load the program, you should create a batch file in which no memory is allocated for a RAM disk or a print spooler, MDOS is not put in TIMODE, and execute this batch file every time you wish to run Hypercopy. The program loads from MDOS by typing in HC-TI (if you have a TI disk controller), HC-CC (if you have a Corcomp disk controller), or HC-MY (if you have a Myarc disk controller). You are then greeted by the title screen (press ENTER) then the menu with seven options. The first is 1) Copy disk to disk, which allows you to make a copy of a disk to another disk, which can have a different interlace and skew, using two disk drives. The second option is 2) Copy disk to disk(s). This option will load a disk into memory, then allow you to make copies. This is useful if you wish to make several copies of a disk, or have only one disk drive. This is also why you must have no memory allocated for a RAM disk or spooler, since Hypercopy tries to use this memory. The third option is 3) Format disk, which formats a disk with the interlace and skew that you specify. The fourth option is 4) Read interlace and skew, which gives you the interlace and skew of a disk. The fifth option is 5) Catalog disk which gives a directory listing of a disk. The sixth is 6) Configure defaults, which allows you to set the default interlace and skew for each of your disk drives. The optimal interlace and skew for your system can only be found through trial and error by creating a large

file with different interlaces and skews, and seeing how long it takes to load this file in. However, I found that the defaults that are already in the system worked best with my TI controller (there are also system defaults for Myarc and Corcomp controllers). The seventh option is 7) Return to MDOS, which allows you to exit Hypercopy.

Overall, I found Hypercopy an excellent program that sped up my floppy disk accesses considerably. One thing to keep in mind is that it can not alter the interlace and skew on virtual disk drives (hard drives, RAM disks, etc.). An option I would have liked to see is the ability to access the different commands from the MDOS command line. For example, if you could type in HC-FORMAT A: to format the disk in drive A: using the optimal interlace and skew rather than the disk controller's defaults; it would make formatting a disk a lot easier. However, these are minor shortcomings. Hypercopy is a valuable asset to any Geneve owner who makes heavy use of their floppy disk.



EXEC - A Super New Utility for the 9640
by James Dow

EXEC allows a 9640 user to run most e/a option 5 programs from the MDOS prompt without TIMODE turned on. This latest offering from Barry Boone helps bridge the gap between the 9640 MDOS environment and the GPL environment. Version 1.0 of EXEC is quite small -- ONLY 21 sectors. If you are using the ramdisk built into MDOS, don't worry, because MDOS is compatible with it.

The following is a partial list of programs that work with v1.0 of EXEC. Note that version numbers of the software tested are included (when known) as a reference only.

- Archiver v3.02 and v3.03
- MDM 5 v1.28
- Firstbase v1.0
- Preditor
- Telco v2.3
- RAG Assembler v7
- RAG Linker v3
- Pterm
- DM1000 v3.5

The following require TIMODE to be turned on to function with EXEC:

- My-word
- DSKU

The above lists are not complete, nor do I specify programs that do not work, since I suspect that most programmers will upgrade software or provide patches to make their programs run under this new extremely useful piece of software.

As usual, Barry is releasing this program as fairware. So if you do use EXEC or any OTHER program of Barry's, please send in your donation and help keep him in the TI community.

FROGGY



Last June, Froggy was surrounded by menacing dinosaurs including the fearsome Triceratops.

What happened next? Well, she dived...

Sightseeing for some weeks, she visited a store that



Triceratops



sold only Diving equipment ("No, Thanks!") and, after 3 months, our heroine finally came back to the surface of the earth.

But she encountered an Indian! Good? Bad? Who knows? And what about the PAGE-PRO Penguin? Did she meet him?



But what is PAGE-PRO anyway? Please stay tuned for more adventures and fun...

PAGE PRO 99

ASGARD Software released this fabulous program last Spring. I bought a copy at our TI-Fest, and immediately used it; unlike "The Printer's Apprentice", reviewed last year but put aside because it is too slow and cumbersome, PAGE PRO 99 is fun, easy to use, and very fast.

It is a WYSIWYG program: you type the text on the screen, and you can have up to 28 pictures on one page; you can have a big and a small font. Some are included with PP99, and you can convert much more pics and fonts from the TI-Artist format (I made a French font in no time at all).

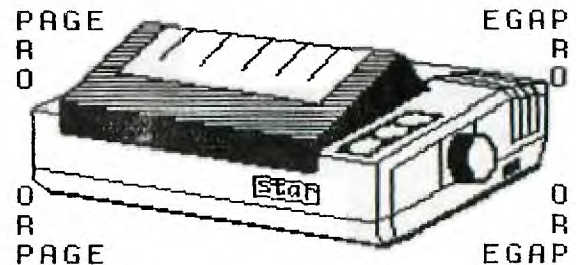


The PPR099 Editor uses some of TI-Writer commands; but, alas, no word wrap, tabs, or reformat, but you can also load a WRTR text. Justification, as here, is done by hand.

INS/DEL affect text only but typing over a pict. will erase part of it. To move a picture, you must delete it, then reload it. PP99 does not keep the pics in memory, only their location on the page. A toggle key allows you to see them or not: screen switching is faster with pics off.



The Line function is used mainly to draw frames, like the two titles at left. You can draw lines or type in four directions!



Printing can be done in draft mode (pics "off") or as camera-ready copy; it is fast, and a ram-disk is an added asset!

A Look At Assembler Language -- Entrance and Exit
by R. A. Green

This month we look at program entrance and exit code. The standard E/A Option 5 program (called memory image) is loaded then invoked (called) via a BL instruction with the GPL workspace. The standard procedure is to save the return address, the workspace pointer and the GROM address then on program exit to restore these and also set the COND bit of the status byte to zero. The following code will accomplish this.

```

*
* Entrance Code
*
* Called by BL linkage.
* Saves: Return address in R11,
*         Workspace Pointer,
*         Interrupt Mask,
*         GROM Address.
*
        BLWP  $+4           Save WSP and STATUS
        DATA ENTWSP-18,$+2 Note short workspace
        MOV  @22(R13),R14  Save Caller's R11
* Note: Our workspace now looks as if we were
*       called via BLWP.
        MOVB *R9,R12       Read GROM address
        SWPB R12
        MOVB *R9,R12
        SWPB R12
        DEC  R12
        B    @BEGIN       Go to start of pgm
*
* Exit Code
*
* Called by B from anywhere in the program
* Restores all values and zeros STATUS
*
EXIT   LWPI  ENTWSP-18     Our short workspace
        MOVB R12,*R10     Restore GROM Address
        SWPB R12
        MOVB R12,*R10
        MOVB R11,@>837C   Zero STATUS
        RTWP             Return to OS
ENTWSP EVEN             Short Workspace
        DATA >9800,>9C02 R9, R10 (GROM stuff)
        DATA 0          R11 (a constant)
        BSS 2           R12, Saved GROM address
        BSS 2           R13, Old WSP
        BSS 2           R14, Old R11
        BSS 2           R15, Old status
*
* Real start of main program
*
BEGIN  LWPI  ....
        ....
        B    @EXIT       Exit main program

```

This Entrance/Exit code will only be executed once per program so it should be optimized for space rather than time. No cost analysis of the code has been done, but I claim it to be the minimum required. Can you prove me wrong?

This code is useful in almost every program you are likely to write, but it hardly seems worthwhile making it into a macro since it will be used only once per program. It is, however, a good candidate for one of the other Assembler Programmer tools, the COPY statement. Your main program then might look as shown below.

```

MYPGM  DEF  MYPGM          Define pgm name
        EVEN             Program entry point
        COPY 'DSK*.BEGINEXIT' Standard code
BEGIN  LWPI  MYWSP        Real start of pgm
        ....
        B    @EXIT       Terminate pgm

```

The TI Writer Option 3 loader uses the same conventions as E/A Option 5 except that TI Writer disables the screen before invoking the program.

Next time we will look at string searching.

"Disks travel in packs."

FAST EXTENDED BASIC

LUCIE DORRIS

Another year ahead... and not too many fresh ideas! Instead of spending the summer thinking about this column, I sat in the sun, but I did manage to write a few new programs. When you can produce nothing more original than a SPRITE TESTER, you need help! Send me your suggestions! To make myself forgiven, I will give you my ROOTS, a short but very useful program.

SPRITE TESTER allows you to test all the sprite attributes before you actually write your program line: definition, color, starting position (dotrow and dotcolumn), motion (velocity), and magnification. The variables are kept on the screen all the time, and change before your eyes as you play with them. If you don't want to enter a sprite definition, there is a pre-defined sprite. This program could also be useful for people who just want to learn more about sprites.

```
100 ! SPRITE TESTER / L.Dorais, Ottawa UG / Aug. 89
110 !
120 CALL CLEAR :: CALL SCREEN(4) :: CALL COLOR(14,3,4)
130 PDS=RPTS("993C7EFFFF7E3C99",4) :: NS=RPTS("1234567890",3)&"12"
140 A=1 :: B=192 :: C=256 :: D=127 :: GOTO 160 :: AS$ CV,DC,DR,J,K,
    MA,MOT,RV,S,SC,ST,X :: CALL KEY :: CALL CHAR :: CALL HCHAR
150 CALL SPRITE :: CALL DELSPRITE :: CALL MAGNIFY :: CALL LOCATE ::
    CALL MOTION :: !@P-
160 DISPLAY AT(3,9)ERASE ALL:"SPRITE TESTER": : : "SPRITE: <1> pre-
    defined": : " <2> user-defined"
170 GOSUB 530 :: MA=1 :: IF K=49 THEN CALL CHAR(136,PDS) :: GOTO 220
180 DISPLAY AT(16,1):"SPRITE DEFINITION:"
190 FOR X=18 TO 21 :: DISPLAY AT(X,2):">":TAB(19):"<" :: ACCEPT
    AT(X,3)VALIDATE("0123456789ABCDEF")SIZE(-16):AS$
200 IF X>18 AND AS<>" " THEN MA=3
210 CALL CHAR(118+X,AS) :: NEXT X
220 DISPLAY AT(1,1)ERASE ALL:"In the SPRITE TEST screen," : "PRESS:" :
    ARROWS/WRZC to POSITION it" : "then" : " FCTN+ARROWS for MOTION."
230 DISPLAY AT(12,1):"any time:" : " 1-8 to change STEPPING" : " K to
    cycle COLOR" : " M to cycle MAGNIFICATION" : " Q to EXIT"
240 DISPLAY AT(24,3):"press ANY KEY when ready"
250 CALL KEY(0,K,S) :: IF S=0 THEN 250 ELSE CALL CLEAR :: J=8 ::
    SC=2 :: DR=73 :: DC=113 :: RV,CV=0
260 FOR X=1 TO 32 :: K=ASC(SEGS(NS,X,1)) :: CALL HCHAR(1,X,K)
270 IF X>1 AND X<25 THEN CALL HCHAR(X,1,K) :: CALL HCHAR(X,2,140,31)
280 NEXT X :: CALL MAGNIFY(MA) :: CALL SPRITE(#1,136,SC,DR,DC)
290 DISPLAY AT(24,1):"(s#,c#,co,dro,dco,rvel,cvel)" :: CALL
    HCHAR(24,32,109)
300 IF RV=0 AND CV=0 THEN MOT=0 :: CALL CHAR(140,"FF818181818181FF")
    :: GOTO 400
310 CALL KEY(0,K,S) :: IF S=0 THEN 310 ELSE IF K>7 AND K<12 THEN 410
320 IF K>48 AND K<57 THEN J=K-48 :: GOTO 310
330 K=POS("EXSDWRZCKMQ",CHR$(K),1) :: IF K=0 THEN 310 ELSE IF K>8
    THEN 460 ELSE IF MOT THEN 310
340 IF K/2=INT(K/2) THEN K=K-1 :: ST=J ELSE ST=-J
350 ON (K+1)/2 GOTO 360,370,380,390 ! position sprite
360 CALL SP(DR,ST,A,B) :: GOTO 400
370 CALL SP(DC,ST,A,C) :: GOTO 400
380 CALL SP(DR,-J,A,B) :: CALL SP(DC,ST,A,C) :: GOTO 400
390 CALL SP(DR,J,A,B) :: CALL SP(DC,ST,A,C)
400 CALL LOCATE(#1,DR,DC) :: GOSUB 520 :: GOTO 310
410 MOT=1 :: CALL CHAR(140," ") :: IF K=9 OR K=10 THEN ST=J ELSE ST=-J
420 ON INT(K/2)-3 GOTO 430,440 ! move sprite
430 CALL SP(CV,ST,-D,D) :: GOTO 450
```

```

440 CALL SP(RV,ST,-D,D)
450 CALL MOTION(#1,RV,CV) :: GOSUB 520 :: GOTO 300
460 ON K-8 GOTO 470,490,510 ! color,magnify,quit
470 SC=SC+1 :: IF SC=17 THEN SC=2
480 CALL COLOR(#1,SC) :: GOSUB 520 :: GOTO 310
490 MA=MA+1 :: IF MA=5 THEN MA=1
500 CALL MAGNIFY(MA):: GOSUB 520 :: GOTO 310
510 DISPLAY AT(12,2)BEEP:"Press <1> RERUN <2> QUIT" :: gosub 530 ::
IF K=50 THEN END ELSE CALL DELSPRITE(#1) :: GOTO 160
520 DISPLAY AT(23,1):USING "(#n, ch,###,###,###,###,###,###)": "#",SC,DR,DC,
RV,CV :: CALL HCHAR(23,32,MA+48) :: RETURN
530 CALL KEY(0,K,S) :: IF S=0 OR K<48 OR K>50 THEN 530 ELSE RETURN
540 !E+
550 SUB SP(V,S,Y,Z) :: V=V+S :: IF V<Y OR V>Z THEN V=V-S
560 SUBEND

```

We start by defining PD\$ as our pre-defined character, repeated four times to make a sprite; NS\$ is a 32-char. string used in the test screen. In line 140, we initialize the limit checking values for sprite movement: I hope to have made the movement routines a bit faster by using variables. The remainder of the line is the beginning of the pre-scan.

The program proper starts in line 160. You must choose between a pre-defined sprite (CHAR 136=PD\$ in line 170) or your own; in this case, you enter four lines of hex codes (for four char.); if you need only one character (magnification 1 or 2), enter nothing on lines 2-4; lines 200-210 will adjust the default magnification and define char. 136-139 accordingly.

Lines 220-230 display the instructions; STEPPING means the number of pixels the sprite will jump if you position or move it; the starting default J is set to 8 in line 250, for a full character jump. For fine tuning, press any digit key from 1 to 8. I used "K" for Color because "C" is used to position the sprite. Other variables initialized here are sprite color, starting dotrow and dotcolumn, and velocities. After you press any key, the test screen is built, with the 24 rows and 32 columns numbered at top and left (the sprite's dotrow and dotcolumn pixel values are eight times bigger); the CALL SPRITE statement and the magnification factor are displayed at the bottom, with a line of explanation (290). Because we refresh that statement each time we change something, we use the subroutine in line 520 to display an IMAGE; but since the character "#" tells Tex to expect a value, how do we display that character for itself? Well, we send it as a constant string value, along with the other variables.

Now you can start to play with your sprite! You can find its starting position or set it in motion, but you cannot do both at the same time. The flag MOT is set to "0" when velocity values CV and RV are "0" (line 300); otherwise, as soon as you press an ARROW+FCTN key, MOT is set to "1" and the position keys (ESDXWRZC) will not respond any more (line 330: IF MOT THEN 310; remember, IF MOT means IF MOT<>0). If you want to modify the starting position after you started motion, just play with CV[EL] and RV[EL] until they are both "0": line 300 will then reset MOT to "0" and re-locate your sprite at its starting position indicated by DR and DC. To indicate that you are back into "positioning", the little squares reappear in the background. The other functions, color (K), magnification (M), stepping (1-8) and Exit (Q) can be used any time, whether positioning or moving the sprite.

The digit key press is dealt with in line 320: stepping default J is reset to the key value minus 48. Changing color is done in lines 470-480; it is a cycle, coming back to black (2) at the end. Magnification (lines 480-490) is also a cycle. Upon exiting with Q, you are asked if you want to RERUN or QUIT.

I tried to simplify my positioning and moving routines one step further than in BARBIE (last April): I still use a user-defined SUB SP, to check for limits Y and Z passed as parameters A,B,C,D (initialized in line 130). But by carefully designing the key-pressed position string (line 330), I grouped them by two and found that the only difference for DR or DC is in the stepping: positive or negative, depending on K being odd or even. By setting ST to j or -J accordingly (line 340), I got rid of half the moving statements! J is still used when we move the sprite diagonally (lines 380-390). Those four lines all lead to line 400, where the sprite is re-located and the new values are shown on the screen.

As soon as you press a FCTN+ARROW, i.e. a key value of 8-11, you branch to line 410; flag MOT is set to "1" and the little squares (char. 140) disappear; then again, ST becomes J or -J depending on the key pressed (the odd-even bit did not work here...). Lower limit for velocity is really -128, but I set it to -127 (-D) for simplicity. Again, our sub SP does the job of stepping and checking. The sprite motion is modified in line 450, and the new values are displayed by sub 520. Have fun!

And now ROOTS, which answers a need I had for years: Tex can give the square root of any number with SQR(x), but I sometimes needed the cubic root, and found no function! I got the answer in a Math program on TV Ontario: you cube many carefully chosen numbers until the result matches your input number. This is called the "Averaging" process because at each pass the program decides if your answer lies before or after the average value of HI and LO. And why not add more roots? My program works with roots 1 to 9!

```

100 ! ** ROOTS ** L.Dorais/Ottawa UG/July 1989
110 !
120 ON WARNING NEXT :: CALL CLEAR :: PR$="PIO"
130 L$=RPT$(" ",28):: E$=RPT$(" ",168):: S$=RPT$(" ",8)
140 CALL CHAR(120,"000000000002050F",121,"1F10202040408080",
122,"018182C2C4646830",123,"0B0101",125,"FF")
150 DISPLAY AT(5,9):"x3y"&FPTS{"",10):S$&"{z" :: GOSUB 280
160 ACCEPT AT(6,12)VALIDATE(NUMERIC)BEEP:N :: IF R>2 THEN 180
170 IF R=1 THEN AV=N :: GOTO 220 ELSE AV=SQR(N):: GOTO 220
180 LO=0 :: HI=SQR(N)
190 AV=(LO+HI)/2 :: T=AV^R
200 IF OAV=AV THEN 220 ELSE OAV=AV :: DISPLAY AT(12,8):AV
210 IF N<T THEN HI=AV :: GOTO 190 ELSE IF N>T THEN LO=AV :: GOTO 190
220 AN=AV :: DISPLAY AT(12,8)BEEP:"=";AN
230 DISPLAY AT(22,1):L$:" [A]nother [C]hange root [P]rint [Q]uit"
240 CALL KEY(0,K,S):: IF S=0 THEN 240 ELSE K=POS("AC PQ",CHR$(K),1)
250 IF K=0 THEN 240 ELSE ON K GOTO 260,260,270,290
260 DISPLAY AT(7,12):E$:E$:E$ :: IF K=2 THEN GOSUB 280 :: GOTO 160 ELSE 160
270 OPEN #1:PR$ :: PRINT #1:S$&" ":S$&" "&STR$(R)&"/" ::
PRINT #1:S$&"\ / " :N:TAB(26):"=";AN:"" :: CLOSE #1 :: GOTO 240
280 ACCEPT AT(5,10)VALIDATE("123456789")SIZE(-1)BEEP:R :: RETURN
290 END

```

ON WARNING NEXT in line 120 tells Tex to repeat the statement if by mistake you enter a letter (or nothing) where it expects a number. E\$ is a 6-line eraser, and S\$ just some spaces. I cut the character definition to single characters, but of course you could CALL CHAR(120,"0000....101"), i.e. up to 123, as one long string.

Line 150 displays a nice root symbol on the screen and gosub 280 to ask for a Root value, the default being "3". The ACCEPT routine in the sub should not accept "0", therefore we validate a string instead of DIGIT. The ACCEPT N statement in line 160 validates NUMERIC because it has to include "E+" if you wish to use scientific notation; even if you don't, Tex will for very big numbers (see below). Next we deal with root values of 1 and 2, which don't need the averaging process. Any number powered to "1" is always itself, therefore its one root is also itself; and of course the XB guru gave us the SQR(x) function to find square roots.

The Averaging process starts in line 180: we arbitrarily set LO to zero, and HI to the SQR value of our number N; why not? It will always be higher than our final answer. Next, AV becomes the average of HI and LO, and Tex finds what the answer T would be if AV were raised to the R power. In line 200, it is told to stop when the current AV is equal to the previous (OAV) one, otherwise it could go on forever! If AV and OAV are still different, we go on: if our number N is smaller than the trial answer T, the root we are seeking is between LO and the average value AV, therefore AV becomes the new HI; otherwise, if N is higher than T, its root is between AV and HI, so AV becomes LO, and we return to line 190. Harder to explain than to do: Tex takes only a few seconds, and displays all its intermediate AV values in line 200. The final ANSwer is displayed in line 220, preceded by a "=" and announced by a beep. You can then print the result if you wish (with a nice root symbol), or change the default root, and/or do another, or quit.

How accurate is it? Well, up to a few decimals, and I think only the really big numbers are a bit off mark, because Tex transforms them into scientific notation. For example, for the cubic root of 57899987899, I got 3868.65045; Tex printed "5.79E+10" as my number; so I re-entered 5.79E+10 as my N value, and got 3868.650719. If you stick to smaller values, you will be closer to the truth!

If you enjoy this column, and would like to get the previous issues, I have decided to "publish" the two previous years, and to give you the programs as a bonus! Some have been updated since they appeared here, so even if you slaved away typing them, you will still get something out of the disks. The year 1987-1988 is ready, and will be featured as the September D.O.M in our library. The year 1988-89 should be ready soon.

And now apologies: I added a few goodies to my SORT program (last May), including J. Clulow's assembly sort (yes, it will be on the 88-89 disk), and found that you cannot save data files of 255 bytes to a physical disk, although my ramdisk never hiccuped! To be on the safe side, change VARIABLE 255 to VARIABLE 254 in line 560. Also, the accept-device routine did not check correctly if you just press ENTER; modify as:

```
550 ACCEPT AT(24,10)SIZE(-12)BEEP:DEV$ :: DEV$="DSK"&DEV$ :: IF
    DEV$="DSK" THEN DEV$=DEV$&"1." :: GOTO 190
```

HOTLINE NUMBERS

The executive has expressed a desire to assist all members should you have some problems or questions, want to do some library swapping or borrow a book. This will be the place to look. Listed here are the members of the executive, committee heads, and others in the group willing to help in their specialized areas. Of course, if you wish to be placed on the list, just give me a call. I know there is a lot of expertise within our Group, so I hope to add to this list. Please respect normal hours unless you specifically know that someone doesn't mind a call at 3am, or use the BBS to leave a message at 738-0617, 24 hours a day, 7 days a week.

- JANE LAFLAMME.....PRESIDENT.....(H) 837-1719 or (W) 745-2225
- AL PALMER.....VICE PRESIDENT.....594-9216
- MARCELLE GIBSON.....SECRETARY.....233-2384
- BILL SPONCHIA.....TREASURER.....523-0878
- MICHAEL TAYLOR.....PAST PRESIDENT.....831-0143
- PETER ARPIN.....SYSOP.....523-0017
- RUTH O'NEILL.....NEWSLETTER EDITOR.....234-8050
- TONY HOPKINS.....ADVERTISING.....746-4463
- DAVE MORRISON.....LIBRARY CHAIRMAN.....737-4889
- JACK McALLISTER.....CASSETTE LIBRARY.....225-6989
- HENRI MONAT.....ARCHIVES.....824-0941
- LUCIE DORAIS.....MEMBERSHIPS.....232-0393
- BOB BOONE.....HARDWARE/SOFTWARE.....(705) 476-9391
- ART GREEN.....ASSEMBLY HELP.....837-1955
- DICK PICHE.....TECH.....521-8667
- CLUB BBS.....SET MODEM TO 8N1.....738-0617

2307

P.O. BOX 2144, STATION D, OTTAWA
*** ONTARIO, CANADA K1P 5W3 ***

EDMONTON 99er USER'S GROUP
P.O. BOX 11283
EDMONTON, ALBERTA
T5J 3L1



FL 54 G TI L

SALE

*k Road, L
IER, ONT
1J 9H6
-2225 Fax:

IRIGLEY DATA
ANELAFLAMME
Serve 7604
TEXLINK) (6)

ie follow

MPUTEF

T and JS

Support your Canadian dealers:

COMPUTER DOWNLOAD UNLIMITED

8 Talon St.
North Bay, Ontario
P1A 1N5
(705) 476-9391
Data (TEXLINK) 476-3043
CompuServe 73657,3617
DELPHI I.D. "CDU"

DATA*PORT

2846 Gottingen St.
Halifax, N.S.
B3K 3E1
(902) 454-0232
Data (TEXLINK) 455-2076

