THE HUGgers
HOOSIER USERS GROUP
People Helping People

April 1988 THE HUGgers NEWLETTER Volume 6, Number 3

MONTHLY MEETING APRIL 17,1988
2:00 PM
ST.ANN'S SCHOOL

## USING TI-ARTIST INSTANCES IN X-BASIC
BY ROY CARLSON

For those of you who have purchased TI-ARTIST, that excellent graphics drawing program from Inscebot, you may noticed a subtle allusion in the back of the manual to the possibility of incorporating instance files into your x-basic programs. Without going into any detail, the manual suggests that since the instance file is saved in a DV80 format, you should be able to read it and perhaps incorporate it into an x-basic printout or display. Well...tain't quite so straight forward.

Ok, background! I started out trying to learn how to do bit-mapped graphics on my printer, whose manual was of course written in ms basic, instead of TI, so we had to interpret that, then fight through the veil of poor explanation. What we finally gleaned was reasonably simple. At any point on the printer line, you can begin bit-mapped graphics by sending "escape,k,n1,n2", then all the graphics in decimal #'s one at a time. Gee, that sounds simple!! No really, n1 and n2 are actually just two #'s that tell the printer how many informations to recognize as a b-m code. N1 is determined by dividing the total # of codes you will send by 256, then using the remainder as n1, and n2 is determined as the integer part of the division.

Example. If you are planning on sending 312 codes of information (a pix about 39 character wide), divide 312 by 256. You get 1 with 56 remainder. Voila, n1=56...n2=1. Then, all you have to do is convert these #'s into TI-basic command form. Example...Print#1:Chr$(27); "K"; Chr$(56); Chr$(1). Now the printer will take the next 312 #'s it sees, and convert each into a one-wire wide by 8 wire high print code. These codes are in the form of an accumulated decimal aquivalent on the eight print wires. (see drawing). The wire at the top is at the 27 value, the bottom at 20. Add the values of each wire together from top to bottom, and you get a number between 0 and 255, (0 being no wires will print, and 255, all wires do!)

Ok, simple, right? Just draw out your picture on graph paper coloring in all the squares appropriately, then spend 2 or 3 hours converting that to numbers for each vertical line of each character square, and then a few more hours to type them all into data statements, and it works. Almost! You must remember to open your printer with a .crlf so that it doesn't do abritrary carriage returns and line feeds in the middle of your picture, and of course, you must send carriage returns at the appropriate times, and you'll probably have to experiment with the line spacing, (on my printer, around 10

(continued on back)

## OFFICER'S CORNER

There are some changes a comin' gang! HUGgers are looking at some new phases of our existence as a club.

St.Ann is considering closing the school. How this will affect us is not known yet, but we should have more information by the coming meeting.

We need to consider whether or not there is enough interest in dual computer families that we need to set up a S.I.G. for the members who use other computers too, especially PC's. At the same time we will have to be considering just what we are willing to do as individuals to help support the group and recruit new members. These are things we have discussed for many months and now is the time to put our ideas and plans into action.

This is a blossoming time for the TI 99/4A there are now ramdisks up to 1 Meg.and you can have a 20 Megabyte hard drive for less than a single floppy drive used to cost!! There are more and more applications programs coming out, ones that can be used for real world tasks, not just to "play" with the computer.

We will have some of these applications being demonstrated at the next meeting as part of the program. The ones we have avaliable through the library are: GENEALOGY a set of TI-WRITER templates and STUDENT ORGANIZER both by Jan Knapp of St.Louis 99'ers. We also have a FANTASTIC new Monopoly game by Steve Karasek. It is fast and works superbly! Steve is a professional programmer who has several releases that will be avaliable to look at this month's meeting.

So, Ihope to see all of you at this meeting, things are happening; and don't forget the elections coming up.

Carl Clark
acting President

## ONE MORE TURBO

by Jim Ellis

I bought the "Desktalk II" from USA-FLEX for $158, it had been $175. I would not be surprised if it comes down even lower as the state of the art advanes, as it undoubtly will. It comes with Quicklink software for an IBM or compatible. I thought that my Avatex 1200 was nice with it being only 90% Hayes compatible, but this new modem takes it all. It sure is nice to have all of those features, such as automatic baud rate. With the Avatex, you had to push a switch to change from 300 to 1200 or vice versa. The "Desktalk II" is fully Hayes compatible, according to the manual. It has non-volatile memory in which you can store one phone number plus other parameters that you desire and they will be there when you need them. Any changes you make but don't save will be lost and the unit will power up with its original defaults. The unit comes with a black plastic case and a very hefty power transformer. It performs at 300, 600, 1200, 2400 baud rates. It is compatible with Bell 103 and 212, plus CCITT V.22 and V.22bis standards. It is upgradeable to 4800 baud for a fee, when available. The 2400 baud rate should come in handy to keep phone bills down when calling those out of town BBS. There are quite a few that support 2400 baud already. I'm not so sure about the 4800 being a plus, as most of the boards jump from 2400 to 9600 and forget that anything is in between. Probably the best thing is how easy it gets to be using a smart modem, no switches to throw, push, flip, etc., just type and go. There are many things about the way the modem responds that can be changed by entering various commands from the keyboard. Since there aren't any buttons, switches or the like, it can be located somewhere out of the way. Too bad it won't go inside the 'P' box. I received it within 48 hrs after placing the order by WATS line and putting it on my MasterCard. That was UPS brown not blue. Later.....

## THE COMPLETE LIST OF CALL LOADS

The following list comes from the Pittsburg Users Group and my own collection. All CALL LOAD'S require X-Basic, 32 K, and must be preceeded with a CALL INIT. They are typed in as follows:
CALL LOAD(address,Values)

| Address | Value(s) | Action Produced. |
|---|---|---|
| 8192 | P | If P isn't 70 or 121 then CALL INIT. |
| 8194 | | First free address in low memory. |
| 8196 | | Last free address in low memory. |
| -28672 | P | If P=0 (no speech.) If P=96 or P=255 (Speech attached) |
| -31900 | N | N=Step. Creates sound. |
| -31572 | 0 to 255 | Vary keyboard response. |
| -31740 | P,Q | Sound Generator P and Q = frequency? |
| -31744 | 0 to 15 | Continuation of last sound.(0-soft, 15-loud) |
| -31748 | 0 to 255 | Change cursor flashing rate. |
| -31788 | 160 | Blank out Screen (press any key to activate) |
| | 192 | No Autosprite or sound. |
| | 224 | Normal operation |
| | 225 | Magnified Sprites. |
| | 226 | Double sized sprites. |
| | 227 | Magnified, Double size sprites. |
| | 232 | Multicolored Mode (48 by 64 Squares) |
| -31794 | P | Timer for CALL SOUND (Counts from 255 to 0) |
| -31804 | X,Y | Return to title screen via CALL PEEK(2,X,Y) prior to the CALL LOAD. |
| | 0 to 255 | Change cursor flashing rate. |
| -31806 | 0 | Normal operation. |
| | 16 | Disable QUIT key (FTCN 4) |
| | 32 | Disable sound. |
| | 48 | Disable sound and QUIT. |
| | 64 | Disable automatic sprite motion. |
| | 80 | Disable sprites and QUIT. |
| | 96 | Disable sprites and sound. |
| | 128 | Disable sprites, sound, and QUIT. |
| -31808 | P,Q | Double random numbers (0 to 255) need RANDOMIZE. |
| -31860 | 4 | X-BASIC to BASIC (via NEW) |
| | 8 | Auto run of DSK2.LOAD. |
| -31866 | P,Q | End of CPU address (P6 + Q). |
| -31868 | 0 | No RUN or LIST after FTCN 4. |
| | 0,0 | Turns off Memory Expantion. |
| | 255,231 | Turns on Memory Expantion. |
| -31873 | 3 to 30 | Screen Columns start with PRINT |
| -31877 | P | P=32 If Sprite coincidence, P=64 If 5 Sprites on a line. |
| -31878 | P | Highest number sprite in motion. |
| -31879 | P | Timer for VDP interrupts every 1/60 sec. (0 to 255) |
| -31880 | P | Random number (0 to 99) need RANDOMIZE. |
| -31884 | 0 to 5 | Change Keyboard mode. Like CALL KEY |
| -31888 | 63,255 | Disable all Disk Drives. Use NEW to free memory. |
| | 55,215 | Enable all Disk Drives. Use NEW. |
| -31931 | 0 | Unprotect X-BASIC programs. |
| | 2 | Sets ON WARNING NEXT. |
| | 4 | Sets ON WARNING STOP. |
| | 14 | Sets UNTRACE. |
| | 15 | Sets UNTRACE and NUM. |
| | 16 | Sets TRACE. |
| | 64 | Sets ON BREAK NEXT. |
| | 128 | Protects X-BASIC programs. |
| -31952 | P | If 55 then Memory Exp. is on. |
| | 255,231,255,231 | Erases program in memory (NEW) |
| -31961 | 51 | Returns to Title Screen. (BYE) |
| | 255 | Restarts X-BASIC. |
| -31962 | 51 | Returns to Title Screen. (BYE) |
| | 255 | Restarts X-BASIC. |
| -31974 | P,Q | End of VDP stack address (P6+Q) |
| -32112 | 8 | Searches disk for ???????? |
| -32114 | 2 | Random Garbage. |
| | 13 | Screen goes wild. |
| | 119 | Produes Lines. |
| -32116 | 2 | Random Characters on Screen. |
| | 4 | X-BASIC to console BASIC. |
| -32187 | 0 | Unprotect X-BASIC programs. |
| | 2 | Set ON WARNING NEXT. |
| | 4 | Set ON WARNING STOP. |
| | 9 | Set 0 line number. |
| | 14 | Set UNTRACE. |
| | 15 | Set UNTRACE and NUM. |
| | 16 | Set TRACE. |
| | 64 | Set ON BREAK NEXT. |
| | 128 | Protect X-BASIC programs. |
| -32188 | 1 | Change Color and get SYNTAX ERROR. |
| | 127 | Change Color and get BREAKPOINT. |
| -32577 | 1 | "Mushy" keybord. |
| | 128 | Disables keybord. |
| -32699 | | Same as for -32187. |

| | | |
|---|---|---|
| -32700 | 0 | Clears Screen for an instant. |
| -32729 | 0 | Runs DSK1.LOAD |
| -32730 | 32 | Reset to Title Screen. |
| -32961 | 51 | Reset to Title Screen. |
| | 149 | Sets ON BREAK GOTO, Locks system. |

THESE REQUIRE EDITOR/ASSEMBLER OR MINIMEMORY:

| | | |
|---|---|---|
| 784 | P | Change background Color. |
| -24574 | 8 | Expands MiniMem to 24K. |
| -30945 | 0 | White Edges. |
| -32272 | 0," | *,-31095,0 Text mode. |
| -32766 | 0 | Bit map mode. |
| -32768 | 0 | Graphics (Normal mode.) |
| -32280 | 0 | Multi-Color Mode. |
| -32352 | 167 | Blanks Screen untill key is pressed. |

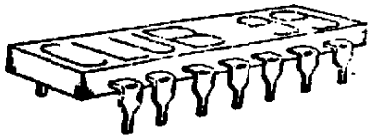
The following are CALL PEEK's: (use same format for CALL LOAD).

| | | |
|---|---|---|
| -28672 | N | IF N=96 or N=127 Speech is attached. |
| -31888 | H,L | IF H & L are 59 and 227, 57 and 221, or 55 and 215 Disk Drives are attached. |
| -31952 | N | IF N=55 NO 32K expantion. |
| 8198 | H,L | IF H=178 and L=85 Then CALL INIT has been executed. |
| -31806 | N | IF N=16 then QUIT has been disabled. |
| -31952 | A,B | Is the pointer to starting address on Line Number Table. |
| -31950 | A,B | Is pointer to the ending address of the Line Number Tables. |
| -31954 | A,B | Is the current line being referenced in the table. |

The author is not responsible for any incorrect numbers or identifications. Some of them have been tried and work well, others have not been tried. (Basicly because I don't know how to use them.) I hope someone can use this list and if anyone knows of any more, please tell me.

Have fun, Harry!

Edgar Lecuyer




## TIP

From Eugene 99/4A

If FCTN 4 is too much of a reach, try this (Hold down at the same time) FCTN-Space Bar-J. It Works!

--Frank Biller

--Librarian

## TI-WRITER TRICKS WITH CLASS!!

by Edward Stamm BLUE GRASS 99 COMPUTER SOCIETY, INC.

copied from 9/87 OZARK 99'ER NEWS pg 5

Some of you may use TI-Writer enough to be extremely proficient at it, while others may know just enough to get by. By necessity, I've had to be counted among the former. With my trusty Gemini 10X, I've written dozens of papers as well as a Master's Thesis with TI-Writer. Along the way, I've picked up a few nifty tricks that make TI-Writer perform like the software written for Big Blue. I thought that I might share a few of these tricks with you. All of you will know some of what follows, and some of you may even know everything, and still others may prefer to do things differently. But some of these tips may benifit enough readers to make this article worthwhile. Many of the hints which follow will work only with Gemini or compatible printers, so be forwarned.

### IN THE BEGINING

When writing a paper, I never worry about formatting commands--at first. I set the tabs at 2 and 38, so that all text can be viewed within the monitor screen. I then proceed to type to my heart's content, knowing that the formating commands will be added later. When text is typed this way, you will run out of memory in about 630 lines or so. Don't type that much. Stop around 400 lines, save it, and start a new file. This is convenient for two reasons. First, it takes forever to save and load a 600 line file (about 30 seconds per 100 lines). While it really doesn't matter (technically) if a file is broken up, this may prevent comprehension difficulties when you are composing the text.

### "Mass" files

After typing the text, and saving it (calling it, say, DSK1.XXX), I call up my "MASS" file (I literally call it "DSK1.MASS" for lack of a better name), which I always keep nearby. As many of you know, one file (when printed) can call up another using the "Include File" (.IF DSK1.FILENAME) formatting

command. To ease things a bit, I put nearly all my formatting commands in one file, with the last line being ".IF DSK1.XXX". The formatting commands I use (but you may prefer others) are:

```
.LM 9
.RM 69
.PL 58
.LS 2
.FI
.HE ^^^^^^^^^^^^^^^^^^^^^^^^^ % (OR.FO)
.IN +5
.IF DSK1.XXX
.IF DSK1.YYY
```

This will give double-spaced text, about 25 lines long, nifty margins, with page numbers, no less. Don't forget to load the file "DSK1.MASS"; not "DSK1.XXX" when formatting.

## Embedded Commands

You are easily able to embed two printer commands, besides the familiar & (underscoring) and @(overstriking) commands (others can be embedded, but with more difficulty; these will be discussed later). These two commands are emphasized and double-strike. The "Emphasized" commands is "ESC E". To embed "ESC E", type the following: "CNT U", "FCTN R", "CNT U" (these three keystroke combinations will henceforth be called "ESC") "E". You will see on the screen a funny character and an 'E". Don't put a space bar between these characters. These will not be printed, and unlike "&" and "@" they will work even if you print the file (PF) while you are in the Edit mode. Everything following these symbols will be emphasized. To cancel the command, type "ESC" "F". Important Note! If you do not intend to "fill" your text (i.e., include the formatting command ".FI"), or if you are printing in the Edit mode, then watch out. Embedded commands exist in the eyes of the monitor, but they do not exist in the eyes of the printer. Consequently, all printed text will be moved two places to the left (at least for the line of text in which the embedded command appears). Embedded double-strike is "ESC" "G", and is cancelled by "ESC" "H". This command is different from "@" in that "ESC" "G" continues to double-strike until commanded to stop. The same warning concerning moving text applies here, too.

## Transliterate Statements

For some documents, I add a few 'Transliterate" commands (.TL x:y,z). A transliterate statement enables the printer to print out y,z whenever it encounters x, without actually printing x. As an important example, think about "underlining". NO, not underscoring, but underlining. For those of us with dot-matrix printers, this is a big difference. Underscoring merely puts a line under each letter, while underlining puts a continuous line under whole word(s). To do this, include these two formatting lines:

```
.TL 60:27,45,1
.TL 62:27,45,0
```

With this, whenever "<" is encountered, the printer begins to underline, and will continue to do so until it encounters ">". Therefore, it these commands are used, <these words> will be underlined, while &these &words will be underscored. ANOTHER IMPORTANT NOTE: If your left margin is other than "0", and you are underlining a lot of continuous text, you will have annoying underlines is the left margins. To avoid this, there are two remedies. One is to print out your document first, and then add the ">" at the appropriate places to cancel the underline and "<" at the begining of the next word to begin it again, and so on. The second remedy is to place your tabs at 9 and 69 when you are typing your document (using those annoying windows) and placing the "<" and ">" at the appropriate places in the text (and make sure that you do not 'fill' your document). I find this to be the easiest method. Because of the hassels, underlining is most ideal for using with words or short phrases only.

Maybe you want to italicize certain words in your text. Put these .TL commands in:

```
.TL 123:27,52
.TL 125:27,53
```

With these commands, everthing following a { will be italicized and everything following } won't be. This is great if you want to italicize just one word in the text for stress.

Perhaps you need subscript for footnotes:

```
.TL 94:27,83,0
.TL 92:27,84
```

When "~" is encountered, everything following it is in subscript. The command is stopped when "&" is encountered. If your text is not being double-struck, then after "&", you should add "ESC" "H", since the superscript command autoatically begins double-strike. To resume emphasized print, you must add "ESC""E", since superscript automatically cancels it.

lines per inch), and of course it is advisable to do it with one directional printing just in case your printer (like mine), perversely likes to indent a line every once in a while. Ok, by this time, I had understood what the printer manual tried to convey about the decimal #'s, and looking at a DVBO file saved by an instance in TI-Artist. I recognized the format. Each line after the first had exactly B decimal #'s under 256, which looked just like the ones I needed for bi-mapping. Ah-Ha! All I need to do now, I thought, was write a small program that would read the file and send it to the printer. After all, in the first line of the instance file, there were two numbers, explained as the width and height of the instance. (2,3 for instance translated into 2 characters wide, 3 characters high.) All that would be needed was to read in 2*8 codes, print them, do a cr. lf. get 2*8 more, etc. Relatively simple! Off we went, did the program, fired the printer, and... what a mess. Didn't look like it at all. First impression, bad program of course. But, when I put my own numbers in from a graph, it worked. Hmmm... Well, it turns out that the numbers stored in the instance file are derived by accumulating horizontal rows instead of vertical. These #'s are useless to the printer. How to convert? I put them into a B*B array, read them back out the opposite way, and guess what? It worked. Enclosed is the program listing that I wrote, so you can look it over and see what I did. It's slow, it's inefficient, but it's a good tutorial in array logic, and printer control. Hope you enjoy it. Now, if we could incorporate this into a text processor, we could have our own "news room" style program. Only major problem seems to be that to get a picture to fit together vertically without venetian blind effect, the line spacing has to be moved, and when you do that, your text gets scrunched together. (Maybe that's why "newsroom" is semi-unreadable, it converts text to graphics). Anyway, anyone wants to try? May the 4's be with you, ROY.

For anyone interested, the illustration and the program listing will be avaliable,free, at the meeting this month.

**HOOSIER USERS GROUP**
P.O. Box 2222
Indianapolis, IN 46206-2222

Forwarding and Address
Correction Requested

INDIANAPOLIS, IN
-PM
13 APR
1988

Earth
Domestic USA

**TIME DATED**
April 17, 1988
**MATERIAL**

[redacted]

Dan Eicher
4410 Cardinal Drive
Indianapolis, IN 46237