NewJUG/North Notes

DECEMBER, 1985 VOL. 3, #12

OFFICERS

President: Steve Marino (385-3739) V. P.: Roger Harrison (385-8904)
Treas.,Gps.: Frank Orlando (427-5988) Sec.: Susan Rasmussen (288-0963)
Notes: Ralph Kopperman (914-359-1444) Software: Andy Westner (967-9154)

FUTURE MEETINGS

Ralph Kopperman

Below are the dates of the next six meetings, all to be held at the Dumont High School cafeteria.

December 10	January 14
February 25	March 18
April 15	May 20

MINUTES

Ralph Kopperman

The November meeting came to order at 7:56. There was no vote on the New Milford school district's proposed disposition of its TI system. There was some further discussion of the new Myarc machine (ant there's much more in the newsletter below). Steve announced that there would be a TI-Writer group meeting at his house on Sunday, December 8. It was mentioned that the elections at the December meeting would be by secret ballot. Two additional nominations were made: Steve Marino for VP, and Roger Harrison for Corresponding Secretary (**Notes** editor). The nominations for all offices were then closed. Thus the elections at the December meeting will be as follows:

PRESIDENT: Ralph Kopperman, Steve Marino
VICE PRESIDENT: Roger Harrison, Henry Hein, Steve Marino
TREASURER: Frank Orlando (unopposed)
RECORDING SECRETARY: Ralph Kopperman, Steve Marino, Jr.
CORRESPONDING SECRETARY: Roger Harrison, Henry Hein.

There was a discussion of inter-club corespondence, and the club voted 19-0-1 (abstention) to send a copy of the newsletter to each known club. It was mentioned that the Southwest 99ers (Tucson) had mailed us their library, and we unanimously voted thanks and reciprocation. Steve, Jr. led a discussion of bulletin boards and possible attempts by government and telephone companies to restrict them. The UNION, a group of bulletin boards, is to begin issuing a newsletter. Steve, Jr. also mentioned that our bulletin board is run by software bought from a man on Long Island who promised to upgrade the program, but now rfuses to look at it. We adjourned at 8:43.

For Sale

(complied by Ralph Kopperman)

Roger Harrison (201-385-8904) wishes to sell a BMC color 13" sound monitor still boxed with warantee, for $200.

I'm interested in selling a Foundation 128K RAMdisk for $125.

The Most BASIC

Ralph Kopperman

Attn: Boiler Room
c/o NewJUG/North
Dear Mr./Ms. Room:

This month I suddenly found myself using the mailing list option of TI-Writer (indeed, I never used TI-Writer at all until a year ago, preferring the more-than-adequate text editing system on the Editor/Assembler module). What is the mailing list? It's the option which allows one to type form letters, like this: type (and save as DSK1.LETTER):

December 4, 1985

1
2

Dear *3*,
and on a separate file (DSK1.NAMES), save:
1 Steve Marino(ENTER)
2 Bergenfield, NJ 07621(ENTER)
3 Steve(ENTER)
* (ENTER)
1 Roger Harrison(ENTER)
2 Bergenfield, NJ 07621(ENTER)
3 Roger(ENTER)
* (ENTER)

If you then go to the text forsatter and enter DSK1.LETTER as "input filename", and after entering your printer name, answer Y to "use mailing list?", leave the default (ALL) in response to "which letters?", and give DSK1.NAMES as the mailing list name, you'll get:

December 4, 1985

Steve Marino
Bergenfield, NJ 07621

Dear Steve,
followed by a page feed, and then:

December 4, 1985

Roger Harrison

Bergenfield, NJ 07621

Dear Roger,

The above illustrates but doesn't completely teach the virtues of the mailing list option; more can be found in the TI-Writer manual, 111-115, 163-171. It does, though, show a feature which is both a strength and a weakness of this option. You want the page feed between letters; however, the point of this column is to tell how to get labels from those perhaps lengthy mailing list files without retyping the files (the rather sick suggestion of retyping the file is on p. 171 of the TI-Writer manual).

Over the past three weeks I've used the mailing list option to type about 300 letters, and getting labels proved to be a major problem. The BASIC program described below solves that problem, and means that YOU, TOO, CAN MAKE BIG MONEY FROM YOUR BASEMENT IN YOUR SPARE TIME, SELLING WIDGETS WITH PERSONALIZED LETTERS! Before describing the solution, let me tell you a bit more about the problem. I wanted to take only the ADDRESS lines (1 and 2, but not 3, in the malilling list file above), and put them on 6-line 34-space labels. After putting on just the appropriate lines (and breaking them when necessary not to run off the end of the label), I wanted the program to skip just enough lines to get to the top of the next label. I wanted it to print commas, but I wanted to avoid the LINPUT command, so that the program could be run in BASIC, thus avoiding the need to remove the TI-Writer module. I also wanted it to edit the text before printing it, so that characters which have a special meaning in TI-Writer (&, @, and ^) would not appear as is on the label.

Now to the solution. First, use the RS (replace string) command in the text editor command mode to replace each "," by a "^", and save the altered file under another name, say DSK1.ADDR (this avoids commas in the text, so the INPUT command in BASIC will accept the whole text). Put in your labels, and set your printer to exactly the line on which you want to begin printing. Then RUN the program below. Enter DSK1.ADDR on the prompt Filename or 'TYPE'?, and 12 to the prompt Lines to be printed? Then, just sit back and watch. If, when you're done, you want to type some other labels manually, RUN the program etering TYPE in response to the first prompt, and you can type in whatever you want, one line at a time, using the editing available in BASIC (if you type a line too long to fit on the label, the program will reject it and give you a warning).

```
10 DATA "^&@",34,",","  "
20 READ RS$,L,R$(1),R$(2)
30 INPUT "Filename or 'TYPE'?":F$
40 OPEN #1:"PIO"
50 IF "TYPE"=F$ THEN 390
60 INPUT "Lines to be printed?":Z$
70 OPEN #2:F$
80 I=0
90 INPUT #2:A$
100 IF ASC(A$)>126 THEN 370
105 IF ASC(A$)=42 THEN 310
110 IF POS(Z$,SEG$(A$,1,1),1)=0 THEN 90
120 FOR U=1 TO LEN(RS$)
130 Q=POS(A$,SEG$(RS$,U,1),3)
140 IF Q=0 THEN 160
150 A$=SEG$(A$,1,Q-1)&R$(U)&SEG$(A$,Q+1,99)
155 GOTO 130
160 NEXT U
170 A$=SEG$(A$,3,99)
180 Q=0
190 R=Q
200 Q=POS(A$," ",Q+1)
210 Q=Q+(1-SGN(Q))*LEN(A$)
220 IF (Q-R)*(L-Q)>0 THEN 190
230 IF Q=R THEN 280
240 PRINT #1:SEG$(A$,1,R)
250 I=I+1
260 A$=SEG$(A$,R+1,99)
270 GOTO 180
280 PRINT #1:A$
290 I=I+1
300 GOTO 90
310 FOR J=I TO 5
320 PRINT #1:" "
330 NEXT J
360 GOTO 80
370 CLOSE #2
380 GOTO 460
390 INPUT "Next line or 'STOP'?":A$
400 IF A$="STOP" THEN 460
410 IF LEN(A$)<L THEN 440
420 PRINT "TOO LONG!"
430 GOTO 390
440 PRINT #1:A$
450 GOTO 390
460 CLOSE #1
```

How does it work? Lines 10, 20 set RS$ to "^^&@", L to 34, R$(1) to "," (so that we can later replace ^ by , everywhere), R$(2) to a space, while R$(3) and R$(4) remain 0 spaces (thus ^ can be replaced by a space, while & and @ will disappear under this program). 30-70 allow your prompts and open the files about to be used (note: if your printer isn't called "PIO", replace this in 40 by its name on your system). 80 initializes a variable I which tells how many lines the printer has put on a label, while 90 enters a line which will be processed by the program to decide whether it should be printed on the label, and just what should be printed.

Recall that the ASCII code of a character is the number that the computer understands to represent that character. ASC(A$) gives the ASCII code of the first character of the string. TI-Writer automatically puts a line beginning with an ASCII code above those (32-126) occupied by ordinary text at the end of each file; 100 makes the program end when that line is enocountered. 42 is the code of the symbol *, and 105 makes the program finish a label when * is the first character of a string (as it is in the last string of any set of mailing list entries--see the Roger and Steve entries above). SEG$(B$,m,n) is the strng of length n beginning at position m in B$; POS(B$,C$,n) is the first position occupied by a string C$ in the string which begins at the n'th character of B$, and is 0 if C$ doesn't occur at all, thus 110 makes the program forget A$ and go back to choose another if it doesn't begin with a

number you listed in I$ (step 60), and go on to process it printing if it does. This processing is in two major parts:

Steps 120-160 look for (130) each of the characters of RS$ in A$ (if POS(A$,SEG$(RS$,U,1),3)>0 then the character is in the text after the number and space, otherwise 130 sends you to look for the next character) and if it's found, change it (in 150) to the corresponding R$(U); then look (as a result of 155) for the next occurrence of that character. They thus do a Replace Search in BASIC. Next, 170 strips the number and space from the beginning of A$ (we don't want them to be printed).

Steps 180-270 break A$ up before printing if it's too long for a label. After the variables Q,R are initialized (180,190), Q becomes the position of the next space in A$ (200) and then the length of A$ if there is no next space in A$ (in which Q became 0 in 200, thus the expression in 210 becomes LEN(A$)). As long as Q is bigger than R but less than L (=34) you go on to find the next space (220 sends you back to 190). 230 sends you to print A$ (280), increase the lines-printed counter (290) and get the next A$ if Q was the length of A$, and thus was not increased by 200-210. Otherwise you print the last initial segment of A$ which wasn't too long (240), increase the lines-printed counter, throw away the initial segment of A$ (260) and go back to work on the new, shorter A$.

When you're done printing all the strings which began with numbers you indicated in 60, 310-330 make sure that you skip the right number of lines to get to the top of the next label, and 360 sends you to 80 to start reinitialize the line counter and start the next label.

You can only get to 370,380,430 which close the files and end the program, through 100, while you only get to 390 through 30-50. 390 allows you to enter a string (with no commas). If you enter $TOP, the printer file is closed and the program ends. Otherwise, if A$ is short enough, 410 sends you to print it and return for the next, while if it's too long, you get a warning and go back to try again.

The program doesn't take particularly long to run, but if you never use %, @, or !, you can drop them from RS$, and it will run faster. As written, the program doesn't print commas in the manual entry mode, but that could be corected by typing ~ in place of , and then sending the entered A$ through a loop like 120-160.

NOTES AFIELD

Henry Hein

Andy Westner gave me a partially edited document from TEXNET about some important news. Here it is, fully edited, most of the info on the mystery clone.

NOTES ON THE NEW COMPUTER from the Chicago Faire by J. Peter Hoddie, Boston Computer Society TI User Group.

As everyone is aware, Myarc is planning to introduce a new computer which is rumored to be based on the design of the ill-fated TI 99/8. In fact, Myarc even had a 99/8 to play with before it was cancelled in just two months before TI left the home computer market. The truth about the 99/8 was that is was only incompatible with the 99/4A. Thus when Myarc decided to design a new computer they had to make major changes to the design of the 99/8 and the result of this work is a computer orignally named "Noah" (from the "arc" in Myarc . . .) and now in search of a number for a name. It was widely expected that Myarc would show this computer at the TI Faire in Chicago on November 2. But no dice. They brought along an empty shell of what the machine would look like and a mother board that they claimed was the machine. You may well ask then, why didn't they show it in operation? The answer is simple, although Myarc wouldn't admit it straight out. They blew a chip on the board when they were working on it the day before the show and were unable to replace it in time. But Lou Phillips, president of Myarc still gave a very clear picture of what this new, unnamed machine is all about. First the basic information. It is expected to be released in the first quarter of '86 and sell for $499. The machine has an IBM key board complete with a slash key where the left shift key should be. There are 10 function keys but instead of being mounted on the left of the keyboard as on the IBM keyboard they are mounted across the top of the unit horizontally. There is also a numeric keypad like on the IBM, but instead of an oversized plus (+) key there is a large enter key to facilitate in numeric entry. The cartrige port has been moved to the upper left hand part of the machine above the first few function keys. It will come initially with 256K of CPU memory (expandable to a full 2 megs), 64K of VDP memory, 64K of ROM, a parallel output, and an RS232 I/O port, two internal expansion slots, and a port to hook up a mouse. The mouse Phillips mentioned was the MS (Microsoft) Mouse which brings up the issue of IBM compatibility (more later).

The internal ROM includes 48K of library routines, 8K of GPL interpreter and 8K (seems like a lot to me) of mouse support. When the machine powers up 16K of RAM is used for various internal tasks so that you are left with about 240K of space for your programs. And remember that all the routines, screen and graphics tables are kept in the 64K of VDP memory, so that you really have quite a lot of memory to work with. If you choose to expand the RAM of the system, it will have to be done externally using 3 off board RAM expansion banks. The current Myarc memory cards such as their 128 and 512K cards will work as memory expansion. The machine is built around the TMS9995 microprocessor which is a more advanced version of the TMS9900 inside your TI-99/4A. The 9995 is 2.3 times faster and comparable in speed to a Motorola 68000 that drives Apple's Macintosh. According to Mack McCormick the 9995 can run as fast as 12 mHz but it looks like it will only be running at an incredible 10.7 mHz due to some technical considerations. The 9995 uses 16 bit parallel memory on the main board which allows it to go even faster then the 9900 which was a 16 bit processor doomed to forever run on an 8-bit bus thus working at only half speed (roughly...). The machine will be able to run nearly all programs written for the 99/4A through a bit on the gate array which when set will make the machine look nearly identical to a 99/4A.

Thus all your software is still good. Almost. Myarc says 99% compatability. The exceptions they've found are programs that use non-standard methods to scan the keyboard. This is only two programs so far. No big deal. The reason for the problem is that the 99/4A has 48 keys and the new machine has 84 so that a different KSCAN routine obviously had to be used. The programs that don't work use their own KSCAN routine and thus will not work. A few more comments on compatability.

There will probably not be immediate support for speech. The machine can support it but there will be no port for you to plug it into the side of the machine. Myarc is planning to develop something like the Triple Tech card from CorComp to allow you to put the speech synthesizer inside the PE Box.

There is worse news though for those of you with a P-Code card. McCormick said that that card is a technical nightmare and that the increased development time and costs to allow it to work wouldn't be worth it. Besides, he added, P-Code is essentially dead as even its creator has abandoned it.

Now here's the bad news for everyone. You can use your current PEB but you will have to buy a card from Myarc to be able to do it. The reason is that the flex cable and card that connect your console the your PEB doesn't have the intelligence or connectors to allow the new machine to access the expanded memory in the PEB on a 16 bit bus or using the new PAB format (more later). However having to buy this new card isn't all bad. It won't have as bulky a cable as the TI card so you can move the console around freely and it will have a time and date function built in so that you don't need a clock card. It is an added expense however. The communications chip is the same 9901 that is used in the 99/4A running at the same speeds. The graphics chip inside the machine is perhaps the single most impressive component. Myarc is using the 9938, a chip TI developed and then abandonded (like all good things). It has 64 pins and is now being produced by the Japanese (who else?). It is fully compatable with 9918A inside the 99/4A but supports extra modes and features. Where the 9918A has 8 control registers for graphics characteristics, the 9938 has 32 which allows for an incredible amount of flexibility and power.

The 9938 has two text modes. The first is identical to the text mode of the 9918A except that you can choose the foreground and background colors from a set of 512 colors instead of 16.

Text mode two is 80 by 24 or 80 by 26 (which allows for a status line at the bottom like on the IBM) with 6 x 9 characters and a choice of two colors from the same 512. Multicolor mode is still there as well as graphics mode one. Graphics mode two allows definitions of 768 different patterns and a choice of 16 colors from the 512. Graphics mode three is the same as mode 2 except that instead of only being able to have four sprites on a horizontal line at a time you can have up to ten on a horizontal row. Graphics mode four is similar but has 256 x 212 resolution and graphics five can support up to 512 x 424 using interlacing but this mode can only be displayed on an RGB monitor. Graphics mode six has 512 x 212 resolution and 16 colors. Each pixel can have its color individually defined. This mode requires the full 64K of VDP memory for storing the screen. Graphics mode seven has the same resolution but uses a full byte of memory to define the color for each pixel which means that each pixel can be one of 256 colors! This mode requires additional VDP memory to use and Myarc has made provisions for up to 196K of VDP RAM to be put in the console. One of the control bits on the 9938 allows for what Phillips calls "animation tricks." He says that it can do screen swapping which essentially provides for automatic animation controlled by the 9938.

The machine will support the old PAB (Peripheral Access Blocks) format in VDP memory so that, in theory, all peripherals manufactured to TI specifications will work. There is some question as to whether or not the CorComp disk controller will work but Myarc seemed to imply that it would. A new PAB format will also be supported. It will be identical to that developed for the 99/8 and will reside in CPU memory for faster speed. It will also allow for logical record lengths of up to 4096 characters instead of the 255 on the 99/4 and will have a full byte reserved for error codes which means there can be 256 error codes instead of 8 as in the old PAB format. Including support for both the new and old PAB formats is one of the major changes from TI's 99/8.

TI was planning to abandon the old PAB format which would have made your PEB 100% useless. Myarc has made provisions so that you don't have to buy a whole new system. Phillips said that the first two peripherals that would be released would be the new PEB interface (described above) and a new disk controller card that will fit in the internal expansion slot for people who don't have (and don't need to buy) the PEB. This disk controller will support quad density disks which means almost a full megabyte of storage on a single floppy. Phillips said that they already have a version of this controller working and will probably release a version of it for the 99/4A as well. After those two cards are complete Phillips says that the next thing he plans to work on is a card that will allow for IBM compatability. He commented that the reason for choosing the keyboard that they are using was that it could be made into a PC compatable computer easily. He also said that 3.5 inch drives were a definite possibility in the not too distant future.

The computer will come with Extended BASIC built in. But not TI Extended BASIC. Instead it will use an advanced version of Myarc's Extended BASIC II. Phillips said that XB II is very similar to GW Basic from Microsoft and is somewhere between 2 and 4 times faster then TI Extended BASIC. A complete description of XB II, which is now available for use on the 99/4A when using Myarc's 128/512K memory expansion card, will be given elsewhere as it is too long to fit here. The additions to XB II that will be included in the new computer include full mouse support, advanced event driven control keys (which means that you can set your program to automatically branch to a certain line number when a given key is pressed), and support for the new PAB format.

Phillips has promised to release a reference manual for the machine similar to the one released by IBM for the PC. In other words, the machine will have an open architecture and no hidden secrets like TI kept with GPL. This should help enormously in getting new software written and hardware built for the machine by third party companies which can fully utilize the incredible power of Myarc's new machine. Phillips has promised to release the machine and claims that Myarc has sufficient capital to allow it to bring the computer to market. He did however admit that they are expecting a "hard, up-hill battle" for the first year.

When asked about other languages, Phillips said that Pascal would probably not be next but that C would be. His reasoning is that C is what is really in vogue now and it would make new software development easier. Listening to Phillips talk about this new machine made a few things very clear. First, that Myarc really has a machine nearly ready to release. Second, that the machine is state of the art and really something that could compete in the current market. Third,

t Myarc is thinking long term and has big plans. Now ther or not a small engineering company from New Jersey working with a computer developed by TI that lost TI millions, can actually succeed is another question. I think that if anyone can, Myarc will. But there is no way to find out except to wait. A few notes concerning this file: This file was written on November 4, 1985 by J. Peter Hoddie, co-director of the Boston Computer Society TI 99/4A user group. It is based on several pages of notes I took at the TI Faire in Chicago on November 2 during a talk given by Lou Phillips of Myarc. This file is not complete in that I have lots more information on the product and many more editorial comments to make. However in the interest of getting this information to you as quickly as possible I have tried to keep this to a bare minimum (about 55 sectors!). A complete article along with a full description of the faire, the products, people, and talks will be completed in time for the November 20 BCS meeting. It should be well over 10 pages in length. If you want a copy come to the meeting or send $1 to the address below. This file is a rough draft. You may distribute it or publish it in part or in whole as you wish but please include the author's name as well as information as to where the final version can be obtained. Thanks. Boston Comp. Society TI 99/4A User Group One Center Plaza Boston, MA 02108 (617)-353-7369 (author's phone) (reconstructed and uploaded by Barry Boone [76354,1637])

NOTES ON THE NEW XBASIC II
from the Chicago Faire

J. Peter Hoddie
Boston Computer Society TI User Group

Myarc has recently released a new version of Extended BASIC which they call Extended BASIC II (XB II). Lou Phillips, president of Myarc, describes this product as a stop gap program until they can get their new computer to market. Which is to say, XB II is essentially the version of BASIC that will be in the new machine with the exception of a few commands (such as mouse support) which are not included in the 99/4A hardware. The biggest advantages if XB II over TI's XB is that it runs between two and four times faster and it can use up to 512K for program storage. XB II will only work with a memory expansion/print spooler/ram disk card from Myarc with at least 128K of memory. The reason XB II is faster is that it is that the entire interpreter is written in assembly language instead of assembly and GPL (TI's slow, interpreted proprietory language). Furthermore, XB II uses CPU memory instead of VDP memory to store strings so that access time to string variables is drastically reduced. XB II is 100% compatable with TI's XB. Myarc uses the assembly loader from the Editor/Assembler cartridge instead of the TI's XB loader so that not only is load time cut way down but assembly programs can be linked which simplifies writing assembly code for XB significantly. The XB II cartridge also includes an empty GROM socket. Phillips said that this socket will allow you to put the GROM from your TI-Writer, Editor/Assembler, or other one GROM ridge into the socket, thus creating, in effect a dual purpose cartridge.

Now to describe some of the new commands in XB II that really make it shine. First off, in XB II you can use 40 column text mode and bit map graphics. Myarc made this possible by moving nearly all the data and tables that TI placed in VDP memory into CPU memory. Thus nearly all of VDP memory is free and can be used for graphics. To support the new graphics modes, Myarc has added a CALL GRAPHS command to set graphics mode, CALL DRAW, CIRCLE, RECT(angle), and FILL commands which Phillips says are similar to GW BASIC from Microsoft. The DCOLOR command will allow you to set the foreground and background colors of the dots being drawn in bit map mode. The graphics routines were written by Mack McCormack who said they were the most difficult routines he ever had to write but he now says they work flawlessly. And Mack is one of the few people who could write these routines for the TI, so if he says they work, they work!

There is a CALL MARGINS commands which allows you to scroll one part of the screen while leaving the rest of the screen intact which will allow the creation of some pretty fancy windowing techniques.

To speed things up more there is a DEFINT commands which lets you create integer variables which run faster and take up less memory. Integers will take up one full word of memory (2 bytes).

Myarc has been around for a long time and worked closely with TI when TI was developing their XB. When TI asked Phillips what he thought of XB he told them (among other things) that he thought it could use a function he called TEKMCHAR. This would allow you to know what key was used to terminate a line of input (i.e. ENTER, down arrow, up arrow, etc.). This would allow a programmer to make the program do different things (such as allow editing of the input field above if input was terminated with an up arrow) depending on how input was terminated. Thus XB II has this function and allows for eight different keys to terminate input.

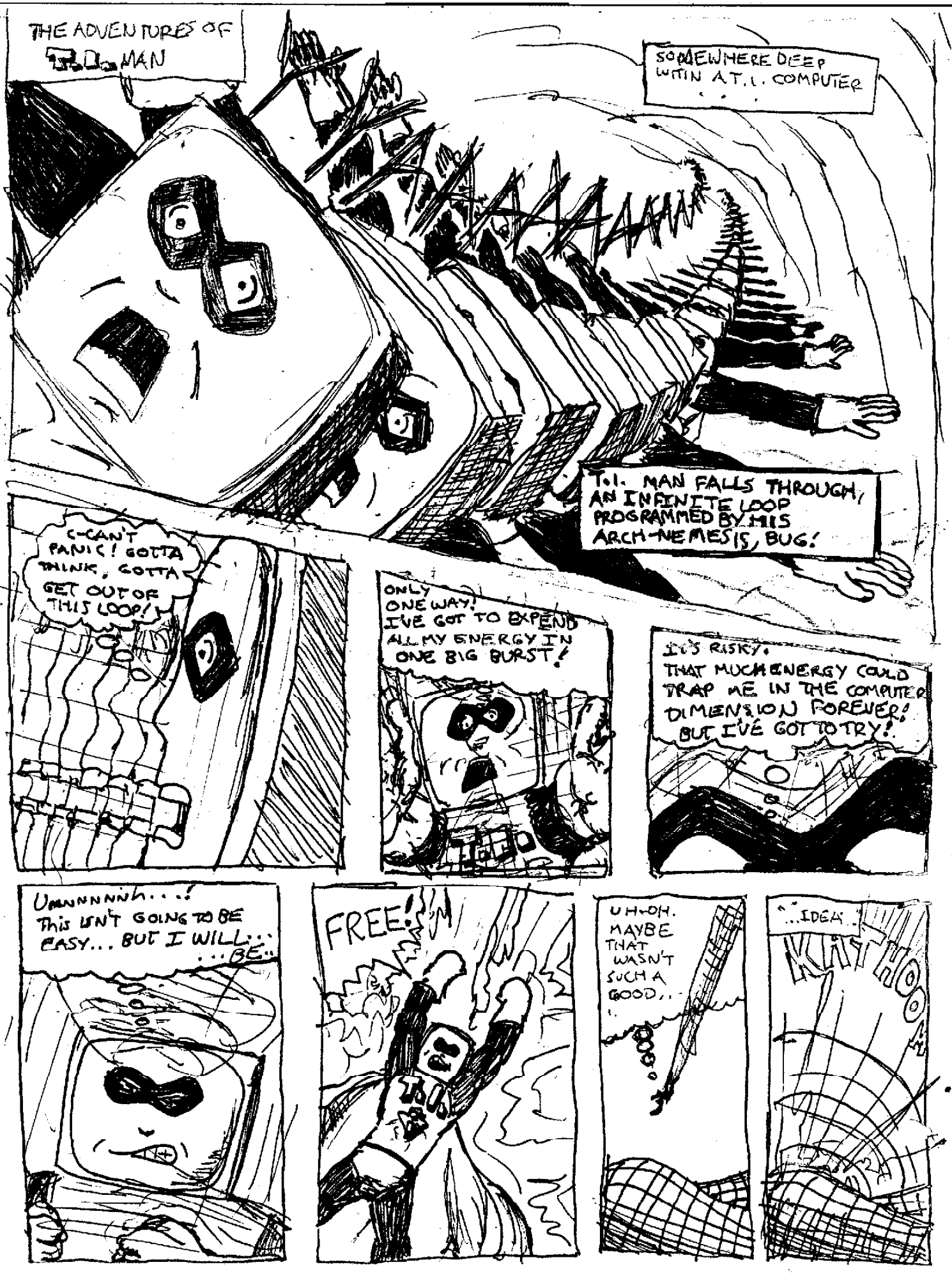
The line editor has also been changed somewhat. Instead of having to hold down the right arrow key to get to the fifth line of a program line to make a change, you can now use the down arrow key which will now just go down one screen line and only go to the next program line after it passes the bottom of the current program line. The same idea applies to the up arrow key.

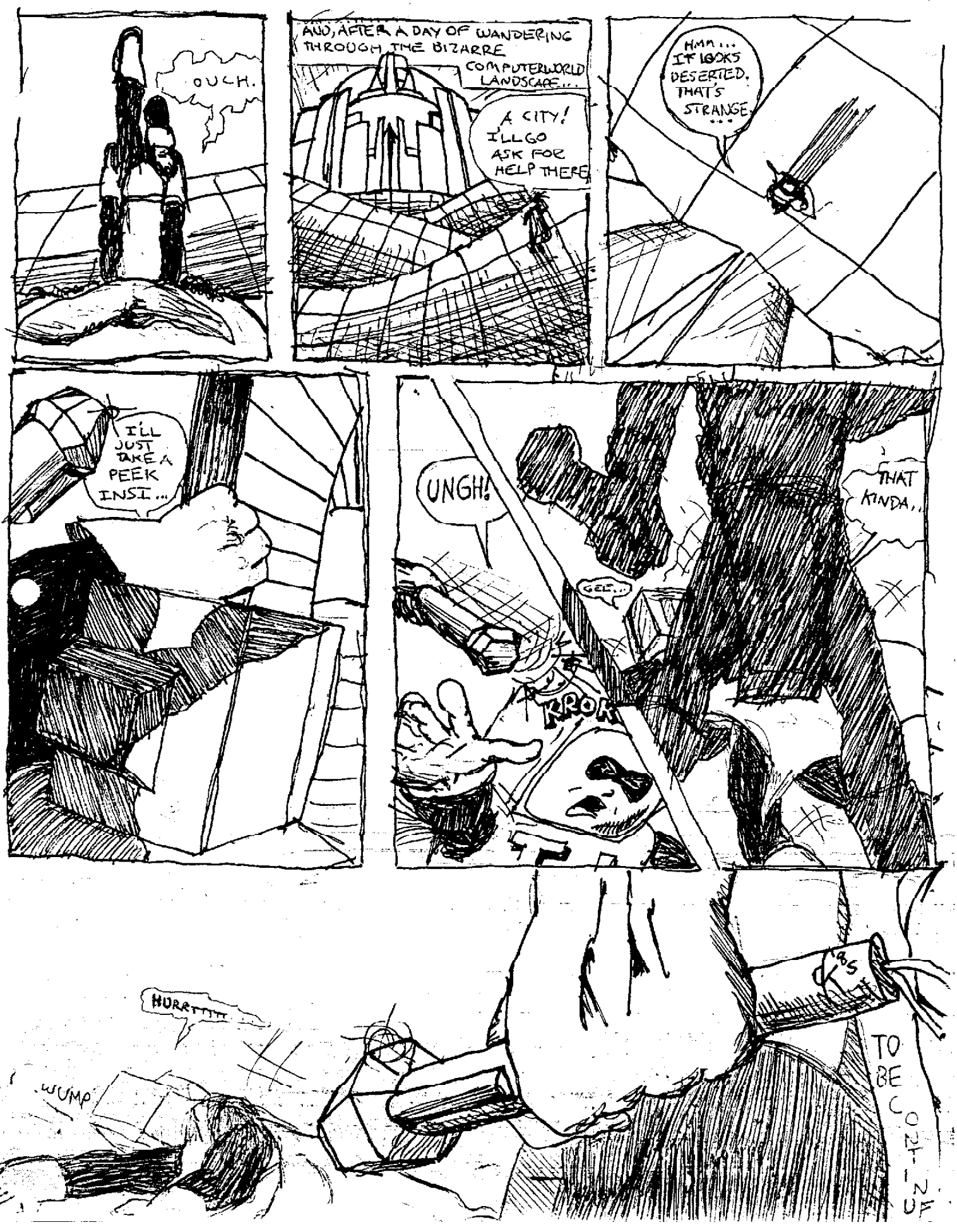
XB II uses the same tokens as XB so that they are fully compatible. The only difference is that XB II must obviously use some of the tokens that were left unused so that it could incorporate the new functions.

XB II will also let you run TI BASIC programs as character sets 15 and 16 are available for use due to some moving around of things in VDP memory. This may mess up some programs that directly POKE or PEEK to VDP memory to control sprites but otherwise should cause no problems.

Phillips said that there will probably not be a compiler for XB II for the 99/4A but that there probably will be one for the new computer which will use an extension of XB II.

XB II is now available along with a 128K expansion card from Myarc for around $250.


THE ADVENTURES OF T.I. MAN
SOMEWHERE DEEP WITHIN A.T.I. COMPUTER . . .
T.I. MAN FALLS THROUGH, AN INFINITE LOOP PROGRAMMED BY HIS ARCH-NEMESIS, BUG!
C-CAN'T PANIC! GOTTA THINK, GOTTA GET OUT OF THIS LOOP!
ONLY ONE WAY! I'VE GOT TO EXPEND ALL MY ENERGY IN ONE BIG BURST!
IT'S RISKY. THAT MUCH ENERGY COULD TRAP ME IN THE COMPUTER DIMENSION FOREVER! BUT I'VE GOT TO TRY!
UMMNNNNH...! THIS ISN'T GOING TO BE EASY... BUT I WILL... ...BE...
FREE!
UH-OH. MAYBE THAT WASN'T SUCH A GOOD... IDEA...
KA-THOOM


OUCH.
AND, AFTER A DAY OF WANDERING THROUGH THE BIZARRE COMPUTERWORLD LANDSCAPE...
A CITY! I'LL GO ASK FOR HELP THERE.
HMM... IT LOOKS DESERTED. THAT'S STRANGE...
I'LL JUST TAKE A PEEK INSI...
UNGH!
GEE...
THAT KINDA..
KROK
HURRTTTT
WUMP
TO BE CONTINUED

8

This file was written by J. Peter Hoddie of the Boston Computer Society TI-99/4A User Group based on a discussion given by Lou Phillips on November 2, 1985 at the TI Faire in Chicago. This article is a rough draft but may be reprinted or used in any other way you wish as long as you include the author's name and information about where more show information can be obtained. For over 10 pages of information and comments on the show come to the November 20 BCS TIUG meeting or send $1 to:

Boston Computer Society
TI User Group
One Center Plaza
Boston, MA 02108
(617) 353-7369 (author's phone)

N E W S B Y T E S
by Henry Hein

REVIEW: DISKO, a program that can do more than it was advertised to do? Get this! Recently I purchased a set of disks, SS,SD, and two had bad sectors on them when I initialized them as DS disks. With the DISKO I was not able to locate the bad sectors because the program was meant to deal with SS disks. I tried this out of curiosity. I made flippies out of these disks and tried to initialize the flip side of each. The bad sectors showed up on the flip side and then I was able to locate the bad sectors, type in the required 'e' symbols where they were supposed to be, gave the 'rewrite sector' command. Now that the required symbols were added I checked with the Disk Manager to see if I had 359 sectors. I did say 359. The first sector is used for the diskname only and the disk manager cataloged 358 Sectors Available. After reinitializing as a DS disk the number 718 'sectors available' showed up. I did it! It worked! No matter how often I tried in the 'test mode' of the Disk Manager it didn't work. It did work with DISKO!

Another discovery! The DISKO was meant to handle only single-sided disks. I found a way to get at the sectors on the back side of the disk. To do that, boot it up, choose the starting and ending sectors of the program you want, write them on a pencil pad, and if the sector ends in a hex number higher than 167 you know it's on the back side of the disk. Reboot DISKO and type in the ending sector number.

You can only call it backwards sector by sector for altering or modifying printer commands (which are easily recognizable in ASCII for your specific printer, for example) and a few other text editing tasks in Personal Record Keeping files, TI Writer, Multiplan, and other data base files. BASIC and XBASIC programs which contain the default command of alien printers can be easily recognized and changed. Even in ASSEMBLER programs can be altered for this purpose, among others. Suppose you were given a BASIC or XB program which had a PIO instead of RS232.BA=4800 or otherwise. You could search the program quickly. As I wrote above, if the hex number on the double sided disk is higher than 167 you can only search using the 'back one sector' key. Below hex 167 you can use both 'back' and 'forward' key commands to search. It does a better and quicker job than DISKFIXER. DISKO will allow you to see in ASCII sector by sector and allows you to modify each sector, and if you know what you're doing you can do many other programming tricks without rewriting programming commands in ASSEMBLY codes or other languages. This is very tricky and only for the pros, however. Meanwhile it's a good trick to perform for some of the elementary stuff I mentioned earlier. There sure was something not bargained for in that DISKO program. You can look for word clues in adventure type games, too. Perhaps even getting insight on how assembly programs work.

Now for the BIG experiment! For those who have the double density cards is it possible to access the 1436 sectors the same way? I don't yet have the card but for those of you who do figure your hex# for 1436 is 59C. Go to it! Remember that it only reads forward to sector hex 167 and backward ONLY down to sector hex 167. It must be noted that the DISKFIXER program can do much of the same except display the sectors in ASCII. DISKFIXER can print to printer, however, what is displayed on the screen. I don't see much value in that, however, since we know very little about the symbols used for control codes unless we check each hex # with their control code uses. Not worth the problem unless you're really into it.

SWAP NOTE: I'd like to swap any game cartridge I have for the SPEECH EDITOR cartridge. Phone (201)-385-9057.

ITEM: For those who have the SPEAKSPELL program and can't get it to run it's because you need the SPEECH EDITOR cartridge. You can't access the program SPEAK1 (All programs and files are protected) unless you modify sector 00 with DISKO by erasing the "P" and "rewrite to disk" keys. Reboot the program SPEAK1 in Basic mode and edit each "CALL SAY" statement with PRINT #3. SAVE the program naming it SPEAK2 and try running it. The program will run only the "SECRET WORD" subprogram with the TEII cartridge in place. I'm still trying to get it to run the other subprograms but have not met with much success. It was a challenge for me to get as far as I did. Calling the TI hot line I was told that it's impossible to do. Really? Still learning!..and with the help of some of you techies out there we may still get the whole thing to run without the SPEECH EDITOR cartridge.

ITEM: MYARC blew a chip before the Chicago Faire. Production updated to end of 1st quarter of 86. An IBM Compatible TI 99'er could be very well worth waiting for!!! Their reputation is well established in their TI 99/4a hardware support items. I'm sure that they can iron out any wrinkles, and we know they wouldn't sell us clinkers.

ITEM: Andy Westner is our club's direct contact with MYARC and asked me to announce that he can get us MYARC equipment for the 99/4a at up to 70% off. Here are some items:

RS232 Card for $70.00
DD/DS Disk Controller Card for $135.00
128K Card $145.00
512K Card $245.00

MYARC's new XB II is now available. It is only compatible with the 128Kor 512k cards. It could be purchased with the 128 K card for about $250. Ask Andy for details.

Do you remember the original price of TI's RS232 card? The SS/SD controller? The others never existed and are definitely worth the price! Roger, Andy, Steve Sr. give them all high ratings.

Question & Answer

?Steve Marino, Jr.?

Hurrah! I finally got a few questions! First, about the screen dump... A screen dump is a file, or small program that will save a screen onto disk, hard copy, or some other means. To demonstrate how a screen dump works, here are 2 small programs that will dump a screen to disk, and save it as a TI Writer file...

To save a screen:

```
100 REM--SCREEN DUMP, SAVE
110 REM--ADD AT THE END OF A FULL SCREEN, AND IT WILL CONVERT THAT SCREEN INTO A TI WRITER FILE.
130 REM--CHOOSE A FILE NAME
140 OPEN #1:"DSK1.[filename]",DISPLAY,VARIABLE 80,INPUT
150 FOR ROW=1 TO 24
160 FOR COLUMN=1 TO 32
170 CALL GCHAR(ROW,COLUMN,X)
180 PRINT #1:X
190 NEXT COLUMN
200 NEXT ROW
210 CLOSE #1
```

To reproduce the saved screen:

```
100 REM--REPRODUCE SAVED SCREEN
110  OPEN #1:"DSK1.[filename used above]",DISPLAY,VARIABLE 80,OUTPUT
120 FOR ROW=1 TO 24
130 FOR COLUMN=1 TO 32
140 INPUT #1:X
150 CALL VCHAR(ROW,COLUMN,X)
160 NEXT COLUMN
170 NEXT ROW
180 CLOSE #1
190 REM--DELAY TO LET USER TO SEE IT...
200 FOR I=1 TO 1000
210 NEXT I
220 END
```

(Editor's note: it's my guess that the above will give some trouble with commas and spaces. To avoid, try running with LINPUT in 140--but note that I haven't, so this is no guarantee. R. K.)

Now, an explaination: In the save program, at line 140 it opens the file under a DISPLAY VARIABLE 80 format (a TI Writer file), INPUT which means, Input to the file. At lines 150 and 160 it takes the screen size in X-BASIC. In line 170, CALL GCHAR is actually CALL GET CHARACTER, in other words, it's taking the characters on the screen and holding them as ROW,COLUMN,X where X is the data. At line 180, it prints X to disk. 190 and 200 end the loops, and line 210 closes the file.

In the next program, it does the opposite. It INPUTS to the file, takes X, and prints it to the screen. I hope that solved your problem...

-_-_->Logon's Run<-_-_-

by Steve Marino Jr.

Some excellent news about the BBS. We are averaging 15+ calls a day, and that isn't bad... Space is not a problem at the moment, but it will soon get there. Some new features are added. Our uploads and downloads are back! Hurrah!

And yes, 'The Union' BBS list...

The Main Base
The Circus.....(201)-592-0456
The Museum.....(201)-943-4379
The Library.....(201)-652-8831
The Falcon's Nest.....(201)-338-9104
The Dugout.....(201)-573-1213
The Outer Limits.....(201)-568-4281
Mystic Caverns.....(201)-791-9183
Guard Tower Nine.....(201)-797-2595
Guardpost 64.....(201)-458-8056
Dragon's Lair.....(201)-929-8161
Bloom County.....(201)-569-3111

KEYBOARD REPLACEMENT
Having keyboard problems?

Replacing a TI99/4A
Keyboard
by
Herman Geschwind
(TI8429)

Replacing a keyboard on a 99/4A is really a very simple job that requires no special skills beyond the use of common sense and ordinary prudence, nor are any special tools or soldering required.

A good source for a replacement keyboard assembly is Radio Shack. Evidently TI unloaded their surplus stock of keyboards to Radio Shack (and several other electronic parts houses). The Radio Shack part number is 277-1017. Be warned, however, since the keyboard was selling for $2.95 or less, sales were brisk and the keyboard might no longer be in stock. It seems that people cannot resist a bargain and I know of TI'ers that bought three or four, just in case. If your keyboard needs replacement, check with your friends or a local users group and chances are good that you might find one, if Radio Shack no longer has a stock. By the way of tools, all that will be required is a medium size (1/8") Phillips-head screw driver and perhaps a flashlight.

First off, inspect your Radio Shack purchase and make sure that the keyboard layout is the same as your 99/4A keyboard. If there are any extra keys or if the keycaps are labelled differently, STOP. Under the same part number Radio Shack also sold non-99/4A keyboards. Adapting a non-99/4A keyboard for use with a TI home computer requires special skills which are beyond the ken of the average layman, if it can be done at all.

TI had subcontracted for keyboards from various sources in Japan and Korea. Thus the shape and texture of the keycaps might be different. Don't let that put you off, the main thing is that the number of keys and the layout are the same as your original.

If your keyboard passed this test, unpack it and test all

keys. Look for keys that might be binding or feel "sticky". Try to get a feel for the key action. Once you are satisfied that your replacement passed this test, go on. Disconnect all cables from the console (power, video, PE box, etc.). Your console should be cool. If you had just used it, allow some time for it to cool down and for whatever residual electrical charges there might be to dissipate. Observe normal precautions about static electricity!

On a clean working surface turn the console over. There will be seven Phillips screws to undo. Four at the the narrow end of the console, three are at the other end and recessed. After undoing the screws, the bottom shell should come off. If it does not, recheck and make sure that you have removed all seven screws.

Once the bottom shell has been removed, three components will be visible: A printed circuit board, approximately 4" square, which houses the power supply; the keyboard assembly, which runs from the power supply board to the edge of the console; and a larger assembly which runs parallel to keyboard and power supply all the way across the console, the motherboard.

The power supply is held in place by two Phillips screws along the edge closest to the keyboard assembly. Remove these two screws and gently move the power supply board an inch or so to the side. Don't force anything since there are wires attached to the power supply board.

Next, locate and undo the four Phillips head screws that secure the keyboard assembly. Gently lift the keyboard assembly an inch or so. You might have to lift the edge of motherboard just a little to allow the keyboard assembly to clear.

At this point you will notice that the keyboard is attached to the motherboard by a ribbon cable. Locate the connector at the motherboard side. Use a flashlight, if necessary. GENTLY pry the connector loose by pressing down with a screwdriver. Do this in several small steps along the length of the connector. The idea is to remove the connector without bending any of the pins on the motherboard.

Once the connector has unsnapped, remove the old keyboard assembly. Take time to take a good look at the row of gold pins that were uncovered when the connector came off. Make sure that all pins are straight and evenly spaced. If not, very gently try to straighten whatever pins were bent. If this is necessary, proceed with caution and use a minimum amount of force!

Insert the new keyboard assembly without forcing it in place. Line up the connector of your new keyboard with the pins. GENTLY start connecting the pins with the connector. Visually doublecheck that all pins mated with the connector. If not, pry the connector loose and try again. Being too hasty at this step could result in a broken pin, so do be careful! Once you are sure that everything is lined up properly, firmly but gently press the connector down on the pins.

Relax now, the most ticklish part of the job is behind you! Next, observe that there is approximately one to two inches of extra ribbon cable. This extra length needs to be folded up and into the space between motherboard and console housing as you simultaneous seat the keyboard assembly in place. To seat the assembly you also need to lift the motherboard edge just a bit for the edge of the keyboard assembly to slip under it, simultaneously try to get the extra length of ribbon cable to fold as described. If this sounds like a job for three hands, you are right and a helper at this point of the installation does make things easier.

Now, line up the screw holes and secure the keyboard assembly with one screw. Lift the console a little bit (remember, everything should still be upside down) and test the row of keys with the numbers on it. There should be full travel for all keys, particularly the keys numbered 4 through 8. If any of these keys appear to bind or feel different from the 1 or Ø key, then the ribbon cable is not folded properly. Undo the one screw and recheck the ribbon cable. Remember, any extra length of the ribbon cable should not touch the keyboard assembly but be tucked into the empty space above the motherboard.

Once the keys check out, replace all four screws on the keyboard assembly.

Relocate the power board, make sure that the ON/OFF switch connects properly. If necessary, lift the board just a little and observe the switch action. Once the switch works properly, secure the power board with its two screws.

Replace the console bottom cover. Make sure that it lines up properly with the top before replacing the screws. Seat all seven screws and then tighten by working over cross.

Your replacement keyboard is now installed and should be working properly. Don't throw your old keyboard away just yet. If it is only a few keys that refuse to work, take it to your friendly radio or tv repairman. Quite often the judicious application of contact cleaner and a good general cleaning can restore a balky keyboard to pristine health.

FROM TI NEWS

THE GEN-II PRODUCT DESCRIPTION

By Perry Metzger

Captured off ++David's Place++

by Steve Marino Jr.

The modern BBS is essentially unchanged from the first ones of several years back. We still have BBS's which act like the big mainframe computers of days gone by, with the user logging in from a remote site, using his computer as a dumb terminal (with maybe some file transfer intelligence) while his system really has resources as great as those as the BBS it is calling. This arrangement forces people to use cumbersome systems, with user impediments instead of interfaces. I speak of forcing users to work with simple line-oriented editors when their system has the power to handle full-screen editors, the display of menus at low speed when the users system can make them pop up at high speed, etc.

This is all because we have allowed modem speed to become a bottleneck to the speed of the individual computers. Full screen editing isn't practical at speeds less than 4800 baud. And at 4800 baud, the line would be mostly free, wasting all that bandwidth except when refreshing the screen. And this would do no good for users of affordable modems. But note that even at 300 baud, the bandwidth of the modem (its maximum data transfer rate) is rarely used completely. What if we could use

it fully somehow, even to the point of using the idle time while reading messages and eliminating the idle time while editing messages? What if we gave the user the full screen editor he wants and let the systems do data transfer while he is editing?

We also note that, with a few exceptions, BBS's are islands of computation. They don't talk to each other, and mail can't be sent from one to another. This, too, is a waste. Systems such as the FIDOnet system of interconnected BBS's, and the USEnet network of interconnected UN*X machines, both utilize the technique of off-peak message transfer, where the systems, late at night, call each other to exchange mail and, on USEnet, national discussions. This system is ideal for BBS's since it can take advantage of late times, such as at 4:00 in the morning, when no one is logged on (or at least not too many people would try to log on) to exchange information. It is also cheap, and, if done by relaying messages through small local leaps and gateways formed by such systems as PC-Pursuit and CompuServe, which potentially offer the ability to transfer large amounts of information at very low cost, can make mail nearly free on a national basis. Although national discussions might be a bit expensive, they would evolve in only a few years when higher speed modems and cheap hard disks make them practical.

But aside from a few hundred FIDO boards, most systems are 100% standalone, and none take advantage of modern communications technology to exploit the currently limited modem bandwidth to its fullest.

The Generation 2 Project, GenII (Pronounced "Jen Two") will change all this. We propose to create a communications program and host software to upgrade BBS technology to the fullest extent possible. We will create a second-generation system that exploits 100% of the modem bandwidth, and networks too, with provisions for local multi-board discussions and national ones when modem bandwidth and larger disk spaces become routine. And now, quick descriptions of what these two new features will do to the BBS as we know it.

We believe that the bandwidth on your modem is underexploited. While you are sitting, typing in your message even at 300 baud on a conventional BBS, you are only hitting a few characters per second, while the modem can handle about 40 a second. In addition, one often pauses messages to read them, wasting the bandwidth totally during that period. And while you are reading, effectively no information gets sent to the remote BBS, wasting that half of the bandwidth.

By moving the BBS (subsequently called the host) and your (remote) system toward looser coupling by moving most functions onto the remote system with the host effectively acting as a file depot, AND using a multiple, multi-channel bidirectional protocol, we can exploit all that wasted bandwidth. The idea is best understood with a simple example. We will use the future David's Place BBS to demonstrate.

John Doe instructs his computer to call David's Place. A carrier is established, and the two systems exchange packets to start up their protocol transmission. John sees a password and username request (no need to establish linefeed needs or parity, its done for you) and enters the information, which is dispatched over to the remote. He is now logged in. Since John has a 1200 baud modem, the system uses an effective 300 baud worth, or 1/4 of his bandwidth, to transmit the login "atmosphere" message giving system news. Meanwhile, the other 900 baud worth of his modem is busy sending over system menus (if his system hasn't stored them -- if it has, only changes are transmitted), his mail, and, if there is still time, any new magazines or messages on boards that he hasn't yet seen. But let's assume that he stops reading the system message (he's seen it a million times) just after his mail has almost, but not quite, finished coming over. It is a torrid loveletter from his wife, Jane Doe, who is on a business trip in Silicon Valley and happens to have an account on a BBS there in the network. Since John can't read at more than 300 or 400 baud, and most of the message was sent over already, the system sends the end of the screen at 300 baud while it continues to start on feeding over new messages on the boards and magazines. John gets 3/4 of a screen of mail popping up instantly, and the end of the letter creeps on the screen at 300 baud. Since he hasn't gotten that far and can't read faster, he doesn't mind. Meanwhile, all of the new magazine on Nuclear Weapons In Urban Renewal has gotten to his machine, and is waiting either on disk or in memory for him to see when he finishes with the letter. He does finish reading the letter, and decides to send an equally torrid reply to Jane. He tells the menu, in conventional current David's Place style (the menus are uploaded and customized for each board) to Autoreply. A full-screen editor with lots of nice features pops up, with Jane's letter in the top window and his reply that he is typing in the bottom (note: mult-window editing, as the pioneering EMACS editor at MIT showed, can be done on plain-jane displays). While he is typing, he is using no bandwidth for normal reading, and the BBS sends, at a full ripping 1200 baud, messages, messages, and more messages, which are all spooled on his computer. Finished with the letter, he tells the system to save it. It is now transmitted at close to 1200 baud on the remote-to-host direction of the full-duplex modem. He then sees the old David's Place main menu (now including a few downloads as they don't interfere with using the rest of the board), selects downloads, and asks for a new dictionary file for his spelling checker. It would take 15 minutes to download normally, but he isn't going to be spending them waiting. And while he was looking over the list of downloads and the menu, which came up instantly, messages were still coming down the pipeline. He selects magazines, and lazily reads a hysterical one on proposals to level the South Bronx with atomics while the last new messages and the dictionary file share the bandwidth, about 600 baud apiece. Note that, had it been a 300 baud modem, he would have still gotten an advantage out of having messages come up the line while he was paused in another message (the system message, for instance) and while editing with the lovely full screen editor. His loveletter to his wife is now just about finished being transmitted to the host. Forgot about it, didn't you? When he is done reading the magazine, he starts reading the debate board, where yet another message about Ronald Reagan, his favorite person in the world next to his dirty-minded wife, is being ragged on again. He starts editing a new message (while the last of the new messages from the board come up, and the system switches full bandwidth to downloading) when the sysop breaks in. Nondestructively. A window pops up, and, depending on the computer he is using, he might be able to continue typing in his reply while the sysop is typing. The fact that the sysop

is typing to him takes up very little bandwidth, and thus doesn't slow down the download appreciably. Our friend finishes his reply, and it starts going down (at 1200 baud) while the upload is occurring and while he is chatting with the sysop. The sysop leaves, and John starts reading the computer's board.

I think you get the idea. The system improves user interface and throughput of information dramatically by separating the two systems and letting the remote use its computing power while storing information coming up the line. And, thanks to interrupt-driven I/O, available even on lowly C64's, it doesn't require multitasking and is easily done with conventional hardware. All of the hardware is conventional. The only difference is the software. And this same software will flexibly support the national GenII Mail Net, and the local GenII Discussion Nets. Plus, we will be able to get gateways to USEnet so you can send mail to your friends and associates there via the gateways. Nice, isnt it? And all of it is 100% feasable.

We plan to write the system in C for portability to nearly all computers. We will thus be able to have it running on most major brands of computers, with the exception of some sytems like old Atari's and Vic 20's and the like. It also can't run directly on the few systems that don't support C like the Color Computer from Radio Shack. But an enthusiast could still write support packages for such systems as the Color Computer. The Vic-20's of the world will never be able to run it, and neither will dumb terminals. But we will have a limited "dumb term" mode in our BBS software to support those few machines (it will be an overlay on BBS's with little memory).

Sound interesting? Want to contribute ideas or help in writing this beast? Well, we want your help. Contact our group on David's Place (201-666-7696), where we are having a discussion on this subject (send E-Mail to us there as well). We can't really accomodate people too far away, alas, as I can't afford to spend too much time on CompuServe and I don't think any big service like that is about to donate time to us so we can communicate. Announcements of meetings for this group are also to be found on David's Place for those who live reasonably near Hillsdale, NJ. If you do live near Hillsdale, by all means call up and find out when the (next) meeting will be. We accept all suggestions and help, and plan to distribute the system afterwards as "User Supported Software" with proceeds going toward maintenance of the system.

Contact PERRY METZGER @[David's Place: 201-666-7696]

Edited & concept co-development by Chris Rowland TWINFINITY . @[David's Place]

Please respond to keep contact.

Dallas TI Home Computer
c/o Richard Roberts
1221 Mosswood
Irving, TX 75061

Ralph Kopperman
Connie Kopperman
49 Cedar Street
Tappan, NY 10983