

VOLUME V NUMBER 2

FEBRUARY 1987

Atlanta, Georgia

PRESIDENTS CORNER

The March meeting will see a continuation of Jim Hubbard's description and use of designing/customizing your own database in Extended Basic.

In April, Gunter Hirshler has agreed to do a show and tell about the making of the various cables encountered with the TI system. In a related vein Gunter has been an intermediary with Boyd Cone about obtaining parts for cables that Boyd wishes to dispose of. Boyd is concentrating on the Printing business and it appears to be going well for him. Boyd indicated appreciation for those who have supported him in the past. Now he is completely separated from the computer business. As a result of this the club obtained some of the soft/hardware that he had not sold elsewhere or that wasn't sold in the 'clearance' sales of the last Swap Meet. These items are what you have seen for sale at the regular meetings.

I have spoken to an acquaintance in Chicago who has 2 buddies who have bought and received the NEW MYARC computer. If anybody if this area buys it; I hope they will bring it to one of the meetings. Speaking of the GENEVE 9640; Myarc has sent 100 flyers on the computer to the club. The flyers will be at the next meeting and a photocopy is included in this newsletter.

Totally personal but informative: I have been looking for a good deal on a monitor that will work with the TI now and also with the new computer if I ever get it (some year). That means it has to have a composite video input (for 40 cols.) and be RGB input for an 80 column display that is fine and readable. There IS such a beast available but I can't pay \$400. Service Merchandise has one normally at \$287 but till March 22nd it is on sale for \$259. Gunter bought one and uses it on his PC clone.

Thanks to Jim Hubbard, the club answering machine has been fixed. (It got zapped by lightening.) So now there is a line available all the time to either ask your question and pass on any tips you may be aware of.

We normally meet on the 3rd Sunday of the month and have for 4 years. At the February meeting it was voted to have the regular meeting on the 2nd Sunday starting on April 12th. It has been pointed out that Mother's day is the second Sunday in May, so I will ask at the next meeting for a consensus for holding the May meeting on the third Sunday GOT THAT?

The very next meeting is March 15th, after that April 12, and after that (if approved May 17.

FROM THAT POINT ON, COUNT ON THE REGULAR MEETING BEING ON THE SECOND SUNDAY!

For any members interested; there is a ballot at the back of the newsletter. Mail it in or turn it in at the next meeting to vote for the year's officers.

The address label makes it your official ballot. The present officers are the ones listed on the ballot. No one else indicated an interest in the position. Voting will indicate your interest, whether you vote for those current or write in another.

The information in this newsletter came from the work of Bobby Miller and his group of helpers who did the typing in of other newsletters articles. I would name the helpers if I knew who to list. There efforts are greatly appreciated.

Gary Matthews

**** OFFICERS CLUB *** ************* Gary Matthews President 404-233-3096 Jim Hubbard Vice President (1)345-5905 w-482-9421George Sears Treasure 396-4112 Melvin Carter Secretary 997-2617 Marshal Gordon Newsletter Chairman Bobby Miller Newsletter Coordinator 921-8308 Ed Banovatz Library Chairman 872-4088

Jim Fairchild BBS(991-6250) System Operator Charles Dupree BBS(366-1914) System Operator

WARNING!!!!!

(From Grand Rapids Users Newsletter)

Beware of a program which may be floating the country's BBS's called SUPERTRACK. At first appearance, it seems to be a track copier, but in reality is a diskeater. This program was uploaded to my BBS recently and I suspect that it will show up around the country. I was suspicious at first because the program tells you to remove the write-protect tape from your master disk. I could not understand why this would be necessary so I stuck two junk disks in my drives. When the programs starts, BOTH drives come on and the heads chatter like crazy. Whatever was on you disks is now is byte heaven. Your disks are zapped and I doubt that the action the heads are getting is doing them any good also. The noise is quite loud.

I have heard of similar programs for IBM which will zero-out a hard drive, but this is the first one I've seen for the TI. I just wanted to warn everyone so that no valuable programs or data are lost.

CALL NEWSLETTER is the voice of the Atlanta 99/4A Computer Users Group, P.O. Box 190841, Atlanta, GA 30325.

The A9CUG is not affiliated with any commercial company or organization.

CALL NEWSLETTER is published by and for the members of the A9CUG to enhance their knowledge of home computers. It is composed of articles written and/or donated by members of our group and from articles appearing in other home computer users groups around the world. Opinions expressed by the authors do not necessarily represent the officers or members of the A9CUG.

Membership is open to family and individuals who own or are interested in using and programming home computers. Membership includes newsletters as they are printed, access to meetings, and membership privileges. Annual dues are \$15.00.

WHY NOT? THE GRAY-2 COMPUTER

The Gray-2 super-computer has a memory of 256-megawords, for the largest memory available in a computer. It can perform 250 million computations per second. Now, would someone please tell me how to connect it to my TI Peripheral Expansion Box?

EXCERPT FROM WORDPLAY PUNN

SHORT NOTES

From Scott King in the AVIT UG Newsletter(also Tigercub#35)

When you load a program in order to modify it, put a reminder of its file name in the first line.

Example: 1 ! SAVE DSK1.PROG/START

Then when you are ready to save it just LIST line 1,FCTN 8 it, using the space bar, erase the "1 !", and press<ENTER>

BEST BUY OF THE MONTH:

From TRITON: TI EXTENDED BASIC => \$29.95 Call Toll-free 1-800-227-6900

MEGRAM from ATRONIC The only full megabyte (1024k) RAM memory expansion for the 99/4A. Price is "only" \$575.95 (US) MULTIFUNCTION CARDS:

DS/DD Controller with 32k==> \$265.95 (US) RS-232 with 32k==> \$175.95 (US) The above items can be ordered from Ryte

The above items can be ordered from Ryte Data; 210 Mountain Street; Haliburton, Ontario KOM 1SO. For full details, see the add on page 19 of the August 1986 issue of MICROpendium.

QUICKIE FLOPPY MAILER LABELS

100 OPEN #1:"PIO"

110 ESC\$=CHR\$(27)

120 EMP\$=ESC\$&"E" :: NOR\$=ESC\$&"W"&CHR\$(0)::
ENL\$=ESC\$&"W"&CHR\$(1):: UON\$=ESC\$&"
-"&CHR\$(1):: UOF\$=ESC\$&"-"&CHR\$(0)

130 CALL CLEAR :: INPUT "ENTER NUMBER OF LABELS": LAB

140 FOR COUNT=1 TO LAB

150 PRINT #1:EMP\$&ENL\$&UON\$&" PLEASE"::
PRINT #1:" DO NOT BEND"

160 PRINT #1:NOR\$&" FLOPPY DISK ENCLOSED"

170 PRINT #1:ENL\$&UOF\$&" DO NOT XRAY"

180 PRINT #1: ::

190 NEXT COUNT

From Northern Nevada Ninety-Niners Newsletter

The program librarian announced that a new Freeware/Fairware Program called Sideways had been released. This program allows you to print Multiplan files sideways on your printer.

SHORTbytes (from Cin-Day News by Jim Peterson)

To get the computer to read the CALL KEY input as upper case letters, even if the Alpha Lock is up, Just use Key unit 3:

CALL KEY(3,K,ST)

To get the computer to hold 24 lines of text on the screen without jumping the first line off the top - just put a semi-colon after the 24th line.

You don't have to RESTORE anything with the RESTORE statement. In other words, you Don't have to read a DATA statement before you RESTORE it. You can write your program to optionally or randomly RESTORE any one of your DATA statements and thus to begin reading DATA from any one of the DATA statements.

If you have the Extended Basic Module, why not leave it plugged in and select the Extended Basic Option even when you are programming in Basic. This will allow you to type 5 lines on a line number, which will still run in Basic(Unless you put too many short items in a DATA statement), and the Extended Basic Option will accept input of your program lines much faster, especially when the program gets long. It also accepts changes and deletions much more quickly, and a large number of lines. It will quickly tell how much memory you have left with the SIZE command (but you'll have more in Basic) and will bring your rejected input back to the screen for correction, with fctn 8. It will also run your program, if you stay away from character set 15 and 16, and watch those double colons.

MECHATRONICS GmBH (Revisited)

Recently I had the pleasure of conducting a telephone interview with Franz Wagenbach of T.A.P.E. Ltd.. Franz had just returned from the storm ridden West Germany. My first

question was what was the GmBH after Mechatronics all about. He said it was the equivalent of our Company of Limited designator.

Mechatronics has two facilities one in Ulm and one is Stuttgard. The facility at Ulm is development area while Stuttgard is the production and implementation division. Ulm is responsible for the new 80 column card. This new card is now available from Tenex, Triton and of course T.A.P.E. Ltd.. The features include as the name implies an 80 column display on an analog RGB monitor.

is installed by plugging in card replacement chips in the video processor. Audio must be taken from the monitor jack in the rear of the console. Oh plug incidentally, the card is not for the PEB stand alone with its own а transformer. The monitor is plugged directly into the card. Future plans call for the mouse to plug into the card. Software for the 80 column card is on chips and can be accessed through Extended Basic with a open type statement.

The card gives you 256 colors with expanded graphics including multicolor sprites. You can also preselect color defaults. A special word processor accompanies the card as TI-Writer is not totally compatible. The price is 219.95. Another product just introduced is DOS 80. This chip replacement on your TI Disk Controller allows you to access an 80-96 TPI (Tracks Per Inch) drive, as single density. That will give you 1440 K on a dual-sided drive.

Word is still out on accessing this capability with a Corcomp or Myarc to give you a one megabyte floppy. Please allow Franz three weeks for delivery. The price for this goodie is \$45.00.

The last item is a disk drive for the CC 40. This machine was TI's answer to the lap portable computer. Unfortunately, accessories are really rare. If you have any questions, Franz is always ready to talk to TI enthusiasts. He plans to set up a BBS from possibly 9 P.M. until? He could sure use some advice on this last item, so share your knowledge. Franz can be reached at: T.A.P.E. Ltd., 1439 solano PI., Ontario CA 91764 Phone (714)989-9906.

Drew Molak Northern Nevada 99ers

PRE-SCAN IT!!

One of the annoying things about TI Extended Basic is the long wait you have between typing in RUN and when the program starts. This is because XB scans the program and sets up memory space for variables, arrays, and data. This wait can be considerable if you have a long program with a lot of arrays and variables. The pre-scan commands are a bit obscure and unless you are an extremely disciplined programmer, they are hard to use. Well, Asgard has just released PRE-SCAN IT! to help us out. You won't find the pre-scan commands listed or discussed in the Extended Basic book, they're in the supplement that came with the book and you really don't get much help as to how to use them.

Using the PRE-SCAN IT! program (hereafter referred as PSI),a relatively inexperienced XB programmer (such as myself) can use these commands to not only speed up execution time, but your program may end up shortened and even run faster. First, load the XB program you want to work on and RES it so that the line numbers start at 100, then save it in merge format. Next run the PSI LOAD program which presents you with the of using XB in a 16K system configuration, in a generic configuration, or with 32K. Once you select the version which best suits your system, you are asked for the input and output file names. Then, can customize your pre-scan for whichever version of XB you will be using. There are files for the Mechatronics XB, for Myarc's new XB, and for Craig Miller's revisions.

There is also a provision for any new CALLs which may be added to SB at a future date. Next, you can change up to 5 constants from a numerical value to special punctuation symbols to save memory space. As in changing OPEN #1:DSK1.FILE to OPEN#:DSK1.FILE You also have the option of deleting REM's. The program will then write a MERGE file containing the proper pre-scan syntax: How does it work? Well, the program is written in XB so it is a bit slow. As a test, I ran my BBS program through it. It took an hour, but the BBS is 80+ sectors long with a large number of variables. Before using PSI, it took about 45 seconds for the BBS to start once I had told it to RUN. After using PSI, it took 6 seconds and I got 5 sectors back.

This program really works fantastic. It is written by J Peter Hoddie who is responsible for getting DM1000 and Fast-Term to sit in GRAM1-2 of GK among other things. So, you probably think that an extremely useful program like this costs an arm and a leg? WRONG....Asgard is selling PSI for only \$10!!!! That is one fantastic value. To get your copy, SMAIL Chris at TI9720 or contact an Asgard dealer. This is a MUST HAVE program for everyone.

BACK TO BASICS

Curtis Alan Provance New Hampshire 99'ers Users Group

DEF statements are often overlooked in BASIC programming because they appear to be limited. Not so! On page 11-105 of the User's Reference Guide, second paragraph, you can find the following:

"The DEFine statement specifies the function to be used based on the parameter (if specified), variables, constants, and other built-in functions" (emphasis mine). Furthermore, you may use DEF's within DEF's, as long as nothing ends up referencing itself. This is all well and good, but is it useful? Consider a program which must print strings or numbers. You may want some items centered, such as headings, etc.

100 DEF CENTERED\$(A\$)=SEG\$(BLANK\$,1, WIDTH-LEN(A\$))/2)&A\$

Did you notice that there are three parameters in the DEF? Blank\$ is a string constant (40 spaces) and width is a numeric variable (the width of your screen, printer carriage, whatever you want).

You may also right justify text:

110 DEF RIGHT\$(A\$)=SEG\$(BLANK\$,1, WIDTH-LEN(A\$))&A\$

Numbers may also be manipulated. This DEF rounds a number to the number of DIGITS:

120 DEF ROUND(A)=INT(A'DIGITS+.5)
/10'DIGITS

As mentioned previously, you may also use DEF's inside other DEF's:

130 DEF DOLLAR\$(A\$)="\$"&RIGHT\$(A\$)
Print codes may also be added easily with
DEF's. These control codes are relatively
common:

140 DEF BOLD\$(A\$)=CHR\$(27)&"E" &CHR\$(20)&"F"

150 DEF BIG\$(A\$)=CHR\$(14)&A\$&CHR\$(20)

If you need help in a special application or want to share some interesting DEF's with other club members, please drop me a note.

FUNNELWEB FARM EXTENDED BASIC TUTORIAL PART 1

Extended Basic Tutorials from FUNNELWEB FARM

1. INTRODUCTION

In this series of notes on TI Extended Basic for the TI-99/4A we will concentrate on those features which have not received due attention in User-group newsletters or commercial magazines. In fact most of the programs published in these sources make little use of that most powerful feature of XB, the user defined sub program, or of some other features of XB. Worse still is to find commercially available game programs which are object lessons in how to write tangled and obscure code. The trigger for this set of tutorial notes was a totally erroneous comment in the TI.S.H.U.G Newsdigest in June 1983. Some of the books I have seen on TI Basic don't even treat that simpler language correctly, and I don't know of any systematic attempts to explore the workings of XB. The best helper is TI's Extended Basic Tutorial tape or disk. The programs in this collection are unprotected and so open for inspection and it's worth looking at their listings to see an example of how sub programs can give an easily understood overall structure to a program.

Well, what are we going to talk about then? Intentions at the moment are to look at: (1) User-defined sub programs (2) Prescan switch commands (3) Coding for faster running (4) Bugs in Extended Basic (5) Crunching program length (6) XB and the Peripheral Box (7) Linking in Assembler routines

Initially the discussion will be restricted to things which can be done with the console and XB only. Actually, for most game programming the presence of the memory expansion doesn't speed up XB all that much as speed still seems to be limited by the built-in sub programs (CALL COINC, etc) which are executed from GROM through the GPL interpreter. The real virtue expansion system for game programming, apart from allowing longer programs, is that GPL be shoved aside for machine code can routines in the speed critical parts of the game, which are usually only a very small part of the code for a game. Even so, careful attention to XB programming can often provide the necessary speed.

As an example, the speed of the puck in TEX-BOUNCE is a factor of 10 faster in the finally released version than it was in the first pass at coding the game.

Other topics will depend mainly on suggestions from the people following this tutorial series. Otherwise it will be whatever catches our fancy here at Funnelweb Farm.

II. Sub programs in OVERVIEW

Every dialect of Basic, TI Extended Basic being no exception, allows the use of subroutines. Each of these is a section of code with the end marked by a RETURN statement, which is entered by a GOSUB statement elsewhere in the program. When RETURN is reached control passes back to the statement following the GOSUB. Look at the code segments.

290 300 GOSUB 2000 310 2000 CALL KEY(Q,X,Y):: IF Y=1 THEN RETURN ELSE 2000

This simple example waits for and returns the ASCII code for a fresh keystroke, and might be called from a number of places in the program. Very useful, but there are problems. If the line number of the subroutine is changed, other than by RESequencing of the whole program (and many dialects of Basic for microcomputers aren't even that helpful) then the GOSUBs will go astray. Another trouble, which you usually find when you resume work on a program after a lapse of time, is that the statement GOSUB 2000 doesn't carry the slightest clue as to what is at 2000 unless you go and look there or use REM statements. Even more confusingly usually change 2000 will RESequencing, hiding even that aid to memory. There is an even more subtle problem -- you don't really care what the variable "Y" in the subroutine was called as it was only a passing detail in the subroutine. However, if "Y" is used as a variable anywhere else in the program its value will be affected.

The internal workings of the subroutine are not separated from the rest of the program, but XB does provide four ways of isolating parts of a program.

- (1) Built-in sub programs
- (2) DEF of functions
- (3) CALL LINK to machine code routines
- (4) User defined BASIC sub programs

The first of these, built-in sub programs, are already well known from console Basic. The important thing is that they have recognizable names in CALL statements, and that information passes to and from the sub programs through a well defined list of parameters and return variables. No obscure Peeks and Pokes are needed. The price paid for the power and expressiveness of TI Basic and XB is the slowness of the GROM/GPL implementation.

DEF function is a primitive form of user defined sub program found in almost all BASICs. Often its use is restricted to a special set of variable names, FNA, FNB,... but TI Basic allows complete freedom in naming DEFed functions (as long as they don't clash with variable names). The "dummy" variable "X" is used as in a mathematical function, not as an array index

100 DEF CUBE(X)=X#X#X

doesn't clash with or affect a variable of the same name "X" elsewhere in the program. "CUBE" can't then be a variable whose value is assigned any other way, but "X" may be. Though DEF does help program clarity it executes very slowly in TI Basic, and more slowly than user defined sub program CALLs in XB.

CALL LINK to machine code routines goes under various names in other dialects of basic if it is provided (eg USR() in some). It is only available in XB when the memory expansion is attached, as the TI/994A console has only 256 bytes of CPU RAM for the TMS9900 lurking in there. We will take up this topic later.

You should have your TI Extended Basic Manual handy and look through the section on programs. The discussion given is essentially correct but far too brief, and leaves too many things unsaid. From experiment and experience I have found that things work just the way one would reasonably expect them to do (this is not always so in other parts of XB). The main thing is to get into the right frame of mind your expectations. This process is

helped by figuring out, in general terms at least, just how the computer does what it does. Unfortunately most TI99/4A manuals avoid explanations in depth presumably in the spirit of "Home Computing". TI's approach can fall short of the mark, so we are now going to try to do what TI chickened out of.

The user defined sub program feature of XB allows you to write your own sub programs in Basic which may be CALLed up from the main program by name in the same way that the built-in ones are. Unlike the routines accessed by GOSUBs the internal workings of a sub program do not affect the main program except as allowed by the parameter list attached to the sub program CALL. Unlike the built-in sub programs which pass information in only one direction, either in or out for each parameter in the list, a user sub program may use any one variable in the list to pass information in either direction. These sub programs provide the programming known as "procedures" in other computer languages, for instance Pascal, Logo, FORTRAN. Lack of proper "procedures" has always been the major limitation of BASIC as a computer language. TI XB is one of BASICs that does provide this the facility. Not all BASICs, even those of very recent vintage are so civilized. For example the magazine Australian Personal Computer in an older issue (Mar 84) carried a review of the IBM PCjr computer just released in the The Cartridge Basic for this machine apparently does not support procedures.

Perhaps IBM doesn't really want or expect anyone to program their own machine seriously in Basic. You will find that with true sub programs available, that you can't even conceive any more of how one could bear writing substantial programs without them (even within the 14 Kbyte limit of the unexpanded TI-99/4A let alone on a machine with more memory).

The details of how procedures or sub programs work vary from one language to another. The common feature is that the variables within a procedure are localized within that procedure. How they communicate with the rest of the program, and what happens to them when the sub program has run its course varies from language to language. XB goes its own well defined way, but is not at all flexible in how it does it.

Now let's look at how Extended Basic handles sub programs. The RUNning of any XB program goes in two steps. The first is the prescan, that interval of time after you type RUN and press ENTER, and before anything happens. During this time the XB interpreter scans through the program, checking a few things for correctness that it couldn't possibly check as the lines were entered one by one, such as there being a NEXT for each FOR. The TI BASICs do only the most rudimentary syntax checking as each line is entered, and leave detailed checking until each line is executed. This is not the best way to do things but we are stuck with it and it does have one use. At the same time XB extracts the names of all variables, sets aside space for them, and sets up the procedure by which it associates variable names with storage locations during the running of a program.

Just how XB does this is not immediately clear, but it must involve a search through the variable names every time one is encountered to trade off speed for economy of storage.

also recognizes which built-in sub programs are actually CALLed. How can it tell the difference between a sub program name and a variable name? That's easy since built-in sub program names are always preceded by CALL. This is why sub program names are not reserved words and can also be used as variable names. This process means the slow search through the GROM that library tables is only done at pre-scan, and Basic then has its own list for each program of where to go in GROM for the GPL routine without having to conduct the GROM search every time it encounters a sub program name while executing a program. In Command Mode the computer has no way provided to find user defined sub program names in an XB program in memory even in BREAK status. XB also establishes the process for looking up the DATA and IMAGE statements in the program.

Well then, what does XB do with user sub programs? First of all XB locates the sub program names that aren't built into the language. It can do this by finding each name after a CALL or SUB statement, and then looking it up in the GROM library index of built-in sub program names. You can run a quick check on this process by entering the one line program 100 CALL NOTHING

TI Basic will go out of its tiny 26K brain and halt execution with a BAD NAME IN 100 error message, while XB, being somewhat smarter, will try to execute line 100, but halts with a SUBPROGRAM NOT FOUND IN 100 message.

The XB manual insists that all sub program code comes at the end of the program, with nothing but sub programs after the first SUB statement (apart for REMarks which are anyway). ignored XB then scans establishes new variable storage areas, starting with the variable names in the SUB xxx(parameter list), for each sub program from SUB to SUBEND, as if it were a separate program. It seems that XB keeps only a single master list for sub program names no matter where found, and consulted whenever the interpreter encounters a CALL during program execution. Any DATA statements are also thrown into the common data pool.

Try the following little program to convince yourself.

100 DATA 1

110 READ X :: PRINT X :: READ X :: PRINT X

120 SUB NOTHING

130 DATA 2

140 SUBEND

When you run this program it makes no difference that the second data item is apparently located in a sub program. Images behave likewise. On the other hand DEFed functions, if you care to use them, are strictly confined to the particular part of the program in which they are defined, be it main or sub. During the pre-scan DEFed names are kept within the allocation process separately for each subprogram or the main program.

Once again try a little programming experiment to illustrate the point.

100 DEF X=1 :: PRINT X;Y :: CALL SP(Y)

:: PRINT X;Y

110 SUB SP(Z) :: DEF X=2 :: Z=X :: DEF Y=3

120 SUBEND

This point is not explicitly made in the XB manual and has been the subject of misleading or incorrect comment in magazines and newsletters. A little reflection on how XB handles the details will usually clear up difficulties.

TI BASICs assign nominal values to all variables mentioned in the program as part of the prescan, zero for numeric and null for strings, unlike some languages (some Basics even) which will issue an error message if an unassigned variable is presumed upon. This means that XB can't work like TI LOGO which has a rule that if it an undefined variable within a procedure it checks the chain of CALLing procedures until it finds a value. However, Pascal which erases all information left within a procedure when it is finished with it, XB retains from CALL to CALL values of variables entirely contained in the sub program. The values of variables transferred into the sub program through the SUB parameter list will of course take on their newly passed values each time the sub program is CALLed.

A little program will show the difference.

100 FOR I=1 TO 9 :: CALL SBPR(0):: NEXT I 110 SUB SBPR(A):: A=A+1::B=B+1::PRINT A;B 120 SUBEND

The first variable printed is reset to 0 each time SBPR is called, while the second, B, is incremented from its previous value each time. Array variables are stored as a whole in one place in a program, within the main program or sub program in which the DIMension statement for the array occurs. XB doesn't tolerate attempts to re-dimension arrays, so information on arrays can only be passed down the chain of sub programs in one direction. Any attempt by a XB sub program itself, CALL either directly indirectly from any sub program CALLed from the first, no matter how many times removed. result in an error. Recursive procedures, an essential part of TI LOGO, are NOT possible with XB sub programs, since CALLing a sub program does not set up a new private library of values.

All of this discussion of the behavior of TI Extended Basic comes from programming experience with Version 110 of XB on a TI99/4A with 1981 title screen. Earlier Versions and consoles are not common in Australia, but TI generally seems to take a lot of trouble to keep new versions of programs compatible with the old. On the other hand TI has also been very reticent about the details of how XB works. The Editor/Assembler manual has very little to

say about it, less by far even than it tells about console Basic. I am not presently aware of any discussion of the syntax of the Graphics Programming Language (GPL), let alone of the source code for the GPL interpreter which resides in the condole ROM of every 99/4A.

Another simple programming experiment will demonstrate what we mean by saying that XB sets up a separate Basic program for each sub program. RUN the following

100 X=1 :: CALL SBPR :: BREAK

110 SUB SBPR :: X=2 :: BREAK :: SUBEND

When the program BREAKs examine the value of variable of X by entering the command PRINT X, and then CONtinue to the next program BREAK, which this time will be in the main program, where you can once again examine variable values.

We will now summarize the properties of XB sub programs as procedures in complete XB programs, leaving the details of joining up the various procedures to the next section. (a) XB treats each sub program as a separate program, building a distinct table of named (REFed) and DEFed variables for each.(b) All DATA statements are treated as being in a common pool equally accessible from all sub programs or the main program as are also IMAGE statements, CHARacters, SPRITES, COLORs, and file specifications.(c) All other information is passed from the CALLing main or sub program by the parameter lists in CALL and SUB statements. XB doesn't provide for declaration of common variables available on a global basis to all sub programs as can be done in some languages. (d) Variable values confined within a sub program are static, and preserved for the next time the sub program is CALLed. Some languages such as Pascal delete all traces of a procedure after it has been used.

(e) XB sub programs may not CALL themselves directly or indirectly in a closed chain. Subject to this restriction a sub program may be CALLed from any other sub program.

(f) The MERGE command available in XB with a disk system (32K memory expansion optional) allows a library of XB sub programs to be stored on disk and incorporated as needed in other programs.

THE END FOR NOW.

TIVRITER OVERLAY OVERVIEW by Tom Kennedy

How many of you have a typewriter, please raise your hand. Keep your hand up if your typewriter has interchangeable text. How about automatic bold and underline? Or some amount of memory storage (for letter heads, etc.)? How about an erase key? Those of you left have probably got a pretty expensive piece of machinery, but TI-WRITER has ten times the functions, or features of the best typewriters. With TI-WRITER, your only limitation is your own creativity.

To start off with, what will you need to operate your Word Processor? You must have the 99/4A console (TI-WRITER won't work with the 99/4), a TV or monitor, the cartridge and disk package, the disk system, memory expansion, the RS232 interface, and a printer. In other words, the whole works. The printer is something you definitely want to be careful in choosing because all of your work will be in vain if you can't print out exactly what you type in, and with an attractive appearance. First, let's look at the command line. That's the line at the top of the screen when you're in the command mode. There are seven commands shown and sixteen sub-commands that are options of the main seven. The commands are selected by typing only the letters that are capitalized in the word. For instance: "F" for Files, "SH" for SearcH, or "LF" for Load File. That's an interesting point: you can access any of the sub-commands from the main command menu. In other words, to ShowDirectory (which is a disk catalog) you would enter the command mode, (FCTN 9), and either type "F" for files, and "SD" for ShowDirectory, or just type "SD" immediately. This feature saves a lot of time and keystrokes. which text is created.

Next is Tabs. When you hit "T", the top part of your text is shown with a scale across the top showing the current tabs and margins. Changes are made by simply typing over existing entries with the appropriate symbol (L.R.T. or I).

Save, Delete, Print, Purge, or ShowDirectory. "PF" for print file is not what you'll get when you print out through the text formatter; it just prints a "hard copy" of the whole file, Just as you see it on the screen. It doesn't print with any of the modifications made by the format commands (more on those later). "PF" is useful for making a fast copy of a long letter, or whatever, in order to check for errors without having to scroll back and forth or up and down. Purge simply erases the file from memory to prepare for a new entry. It is similar to the "NEW" command in BASIC.

Next is "L" for Lines. This allows you to work wit whole lines or groups of lines by moving them to somewhere else in the text, copying to somewhere else and leaving the original intact, to delete groups of lines, or to quickly move the cursor to some line in the text with the ShowLines option.

Search (or "SS") gives you the option of either the FindString routine or the ReplaceString routine. FindString will move the cursor to the first and/or each successive use of the word string you give. ReplaceString searches the text for a given string and replaces all or one occurrence with the new string. This is great for correcting a repetitive spelling error.

RecoverEdit is a failsafe repair in case the text buffer was purged in either the File or Quit command. It will pull back everything but the first line and restore the file. I guess the loss of the first line is the penalty paid for accidentally erasing a file, which can't be done very easily.

Finally, Quit, as the name implies, blows it all apart and leaves you with the reasing a file, which can't be done very easily.

Finally, Quit, as the name implies, blows it all apart and leaves you with the reasing a file, which can't be done keystrok<u>e</u>s. first command is Edit. This simply enters you into the text-edit mode in your fingers are at. The problem though, is that it can be very confusing trying to remember the fifty different key combinations that activate the thirty functions. A better method is to just pick which keys you're going to use for what function and ignore the rest. What I do is use the number line keys for anything shown on the overlay strip and just memorize the few functions hidden down in the keyboard. Let's start by going down the overlay strip, left to right as shown on the next page.

The last four key functions to mention are the cursor arrows: UP, DOWN, LEFT, & RIGHT. These stay the same as in console BASIC. Now, if you're still following along you may be quite confused with this onslaught of information. The point is, you can't learn all of this in one sitting, but after using TI-WRITER for a while you start to pick things up as you need them. Rest assured, you do spend the majority of your time typing. The purpose of most of the functions I've mentioned are to manipulate the text which is already in the file. I have simply tried to cover all of this in order to bring something to your attention that you might have missed, or to peak your interest in the capability of the TI-WRITER software.

To review, in the command mode we can choose between Edit, Tabs, Files, Lines, SearcH, RecoverEdit, or Quit. As sub-commands of those seven, we can choose Load File, Save File, Print File, Delete File, Purge, ShowDirectory, Move Lines, Copy Lines, Delete Lines, Showlines, FindString, ReplaceString, or Exit.

```
* Then it draws all subsequent words up through the paragraph until it

* encounters a Carriage Return.

* FCTN 2 * In Word Wrap mode (solid cursor), 32 blank characters are inserted

* (CTRL G)* after the cursor. The bulk of the text is pushed down the line. After

* insertion of new text, hit Reformat. Any remaining spaces are removed.

* In the Fixed mode (hollow cursor), this operates the same as BASIC.

* CTRL 3 * This allows you to choose which of the five color combinations of

* text/screen you prefer. The default, for no good reason, is white on

* dark blue. This is hard on the eyes. I prefer to turn down the color

* on my monitor and use either black on green or black on light blue.

* (CTRL N)* of the line.

* CRTL 4 * This advances the cursor to the beginning of the following remains and the cursor is on.
  Ins Char * FCTN 2 *
  Screen
Color
 Del Line * FCTN 3 *
*(CTRL N)*
*(CTRL N)* of the line.

Next *CRTL 4* This advances the cursor to the beginning of the following paragraph
Paragraph*(CTRL J)* and puts the first line at the top of the page.

Roll Down* FCTN 4* This is a "vertical block scroll", meaning the next 24 lines of text

* of text are shown. Scans quickly down the text to get to some point.

Dupe Line* CTRL 5 * Creates a duplicate below of the line the cursor is on. The Move/Copy

* function can do the same, but this key makes it faster and easier to

* create repetitive lines such as a double row of '*'s under a title.

Next * FCTN 5 * A "horizontal block scroll". It jumps across to

* display the next block of 40 characters, in increments of

* 20. For example, the screen starts out on column one to

* forty, then twenty to sixty, then forty to eighty.

Last * CTRL 6 * The opposite of "Next Paragraph"
Last * CTRL 6 *
Paragraph*(CTRL H)*
Roll up * FCTN 6 *
                                                                           The opposite of "Roll Down"
*(CTRL B)*

Word Tab * CTRL 7 * This moves the cursor down the line to the first letter of each word.

*(CTRL W)*

Tab * FCTN 7 * Just like on a typewriter, this moves the cursor to next setting,
                                                                           Just like on a typewriter, this moves the cursor to next setting, defined using the Tab function on the command line. Places Carriage Return at end of current line, then skips down to next line. If you have preset an auto-indent, (by using an "I" in Tabs) then it also indents over to the proper column. Inserts a blank line above the line the cursor is on.
                                    *(CTRL I)*
                                          CTRL 8
 New
 Paragraph*
 Ins Line * FCTN 8 *
                                   *(CTRL 0)*
* CTRL 9 *
New Page * CTRL 0 * Inserts a blank line with a Np and Cr symbol at the beginning.

* This causes the printer to feed to the next page.

Command/ * FCTN 9 * This is how to exit from the edit mode to get to the command line.

Escape *(CTRL C)* It is also used to cancel a command already in progress.

Word Wrap*CTRL 0 * Switches from the "Word Wrap" mode to the "Fixed" mode. In Word Wrap,

* upon reaching the end of the line the cursor jumps to the nest line.

* If you're in the middle of a word at the end of the line, the word you

* were on moves down too. This allows you to just type continuously

* without looking up to see when to hit enter. In the fixed

* mode, when you reach the end of the line your letters just

* pile on top of each other and you hit enter to move to the

* next line.

Line * FCTN 0 * This removes or displays the four-digit line numbers at the left side
                                                                          This removes or displays the four-digit line numbers at the left side of the screen. The numbers are used for reference when manipulating blocks or lines of text, just like when editing a BASIC program, line numbers are needed to refer to where changes will be made. Quit is the same as in console BASIC. Use Quit option of the Command line to safely exit TI-WRITER. The same as Tab except it backs up one setting.
                                         FCTN 0
 Line
 Numbers
 Quit
                                          FCTN = *
 Back Tab *
                                          CTRL T *
 Beginning* CTRL V *
                                                                           Moves the cursor to the beginning of the line you're on.
of Line *
Del.End *
of Line*
```

TIVRITER FORMATTER OVERVIEW

by Tom Kennedy

Now I want to cover the Text Formatter, which prints out the document. Most importantly, the special symbols, called Format Commands, that the formatter uses to alter the print-out of the document, which are installed in the Text Editor.

In other words, you put these commands into the text when you write it and as the formatter comes across them it changes the text accordingly but doesn't actually print the symbols.

There are six groups of formatter commands that are all applied in a similar manner. All commands must be in caps and must be on a line that starts with a period.

The use of these commands in your text is what separates the word processor from a typewriter. They allow you to get the most out of your printer.

So, now you've written your document, and inserted all the format commands, now how do you print it out? First, save the document and exit the Text Editor. At the title menu, select Text formatter, (make sure the program disk is in the drive) and the screen will blank with the prompt "ENTER INPUT FILENAME". Enter the name of the file you just saved, (ex. DSK1.MYFILE) and hit enter.

Next, the prompt "ENTER PRINT DEVICENAME" appears after the file is loaded. If you use a serial printer, the device name would be RS232.BA=xxx with xxx being the baud rate. If you're using a parallel printer, the device name is PIO. Also, you must add either .CR or .LF to the end of the device name. This tells TI-Writer whether your printer will handle the carriage return or the line feed. Check your printer manual and the TI-Writer manual in detail to find out which you use.

The next prompt is "USE MAILING LIST". If you aren't printing "form letters" just hit enter to accept the default of N (NO).

Next is "WHAT PAGE(S)? <ALL>. If you want to print the whole document, accept the default for all pages. Otherwise, you can print any of the pages or groups of pages.

The prompt "NUMBER OF COPIES: 1" tells how many copies of each page are to be printed.

The last prompt is "PAUSE AT END OF PAGE? N". The main purpose of this function is if you are using separate sheets of paper it will stop and wait for you to align the next sheet.

Now, about the Mailing List Option. Let's say you've written a form letter to send out to various individuals, maybe a resume'. You write the letter like normal, but when you come to a name or address or something that will change with each letter, you put in its place a variable in the form of *n*, where n is a number to identify the order. So instead of starting off with: "Dear Mr. Smith" you would have "Dear Mr. 1*" and so on. when you're all through with your letter, save it and purge the memory. Now you must create what is called a Value File, which is your mailing list where TI-Writer will draw the variables from. A value file consists of a list values to be inserted into the letter, listed one to a line, preceded by the number of the variable and ending with a carriage return symbol. Groups of values must be separated by a line with just an asterisk and a carriage return. For example:

- 1 John Smith
- **2 123 STREET**
- 3 Seattle, WA
 - *
- 1 Jane Doe
- 2 456 STREET
- 3 Seattle, WA

At the top of your letter you insert the .ML f command where f equals the filename of your value file. After selecting the mailing list option the computer will use this command to fill in the variables. If there is no .ML command in the letter then when you are prompted for "MAILING LIST NAME:" you supply the filename. This allows you to call on a number of files for different groups.

```
Page 12
                             A9CUG CALL NEWSLETTER
Text Dimension commands, as the name implies, move or shape the words in the
document (margins, linespacing, right justify, etc.)
       : FILL
                  : PUTS AS MANY WORDS ON A LINE AS WILL FIT.
 .NF
       : NO FILL : CANCELS FILL.
 . AD
       : ADJUST
                  : ALIGNS THE TEXT TO THE LEFT AND RIGHT MARGINS. (RT. JUSTIFY)
 .NA
       : NO ADJUST: CANCELS ADJUST.
 .LM n : LF MARGIN: SETS LEFT MARGIN TO "n".
 .RM n : RT MARGIN: SETS RIGHT MARGIN TO "n".
 .IN n : INDENT
                  : CREATES AN AUTO-INDENT FROM LEFT MARGIN.
 .LS n : LINE SP : SETS LINE SPACING TO "n" LINES.
 .PL n : PG LENGTH: DEFINES NUMBER OF LINES TO A PAGE.
       : BEGIN PG : DEFINES FIRST LINE OF NEW PAGE.
Internal Format commands control the spacing of characters on a line.
 .SP n : SPACE
                  : SIMILAR TO THE TAB FUNCTION.
 .CE n : CENTER
                  : CENTERS NEXT "n" LINES BETVEEN MARGINS.
Highlighting commands control functions such as underline or bold and allow you to
redefine characters to use them to send CTRL codes to the printer.
       : REQUIRED : JOINS WORDS TOGETHER WHEN REQUIRED TO PREVENT SPLITTING IN
                         REFORMATING, UNDERLINE, ETC.
           SPACE
       : UNDERLINE: (UNDERSCORE) UNDERLINES ALL TEXT FOLLOWING UNTIL NEXT PACE.
 &
       : BOLD
                  : (OVERSTRIKE) RETYPES FOLLOWING TEXT FOUR TIMES.
 .TL xx: TRANS-
                  : ALLOWS REASSIGNMENT OF ONE CHARACTER TO REPRESENT A NUMBER.
       : LITERATE : OF CHARACTER VALUES TO SEND CODES TO THE PRINTER.
 .CO t : COMMENT : SIMILAR TO REM IN BASIC -- ALLOWS NOTES THAT DONT PRINT.
Page identification commands print notes in the upper or lower corner of each
page, either headers or footers.
 .HE t : HEADER
                 : PRINTS TEXT (t) AND PAGE NUMBER AT TOP OF EACH PAGE.
```

- .FO t : FOOTER : PRINTS TEXT (t) AND PAGE NUMBER AT BOTTOM OF EACH PAGE.
- .PA : PAGE # : RESETS PAGE NUMBER IN .HE AND .FO

File management commands

- .IF f : INCLUDE : MERGES A FILE TO PRINT A DOCUMENT TOO LARGE FOR ONE FILE. FILE
- Mail Merge option commands are used to supply values to the variables in a letter that has been set up for the mail merge option
 - :MAIL LIST: IDENTIFIES VALUE FILE (f) FOR MAIL LIST.
 - :VARIABLE : INSERTED IN TEXT AS VARIABLE FOR ASSIGNMENT FROM VALUE FILE.
 - .DP n:t:DISPLAY : PROMPTS YOU USING TEXT "t" TO ASSIGN TO VARIABLE (*n*).
 - PROMPT:

Another way to insert values is to use the Define Prompt command. With this command you do not insert a .ML comand calling a value file and instead you insert lines containing the format: .DP n:t - where n is the number of the variable and t is the prompt text. Now, when you come to the prompt "USE MAILING LIST?" you select "N" for NO and as the document is printed when a variable is encountered the printing stops and the text you chose appears on the screen asking you for the appropriate value. If you don't include a ".DP n:t" command in your text, the computer responds with "ENTER DATA FOR VARIABLE *n*" and it can get confusing trying to remember which item you're on. This method is handy for letters which you only want to print one copy at different times to different people.

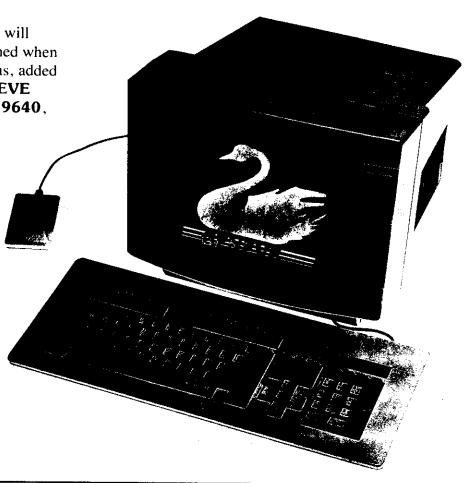
Let me tell you, this is why I bought a computer. I'm sure we all went through that period of time before buying a computer when we would ask: "what am I going to use a computer for, anyway?". Well I decided there were two things I wanted to do: 1) Store files of data (recipes, albums, etc.) and 2) Use my computer as a typewriter. I didn't know about TI-WRITER when I bought the 99/4A, but now I know that I made the best choice possible. I hope you will all find TI-WRITER as easy to use and as powerful as I have.

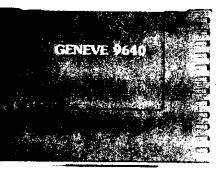
The GENEVE 9640 by MYARC, Inc.

It was a long time coming, but we think you will agree, the wait was worth it. MYARC listened when you told us what you wanted, took your ideas, added many of our own, and engineered the GENEVE 9640. Take a close look at The GENEVE 9640, and see, if you don't agree.

The **GENEVE 9640** has composite video output, to connect to a TV or computer monitor, like the T199/4A, as well as both high resolution (512 x 212 pixel's), and very high resolution (512 x 424 pixel's) RGB video output. In addition, up to 256 colors can be displayed on the screen at one time.

A PC style keyboard with input buffer is included with the **GENEVE 9640**. The RGB High Resolution Monitor with Sound, Monitor Cable, Mouse, and Enhanced Keyboard are optional.





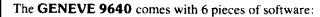




The **GENEVE 9640** comes in a protective shell ready to plug into the Tl PEB. Con-

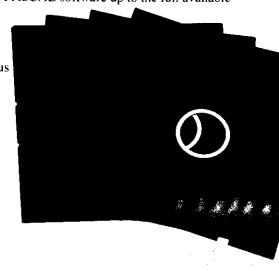
nections are provided for Monitor with audio, Joy Stick, Mouse, and Keyboard.

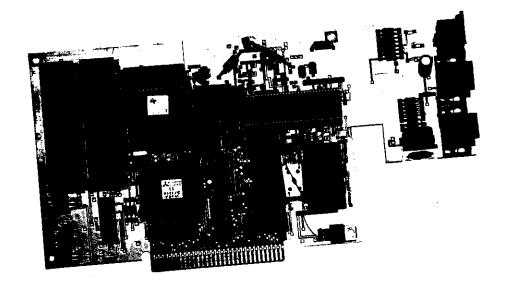
An outstanding 350 page manual with sections covering the introduction to and setup of the GENEVE 9640, using M-DOS, and Advanced Basic.



- 1. Cartridge Saver which allows the saving of most cartridge software to disk, then loading and running the saved software from disk.
- 2. Advanced Basic with new commands like draw and fill, the increased operating speed you always wanted, plus 32, 40, and 80 column display, and yes, it is compatible with T1 BASIC and EXTENDED BASIC.
- 3. The newest version 4.21 PASCAL run time to allow the loading and running of standard PASCAL software up to the full available memory.
- 4. T1 Word Processor upgraded to 80 columns, increased operating speed, plus Edit and Format reside in memory at the same time.
- 5. Microsoft MULTI-PLAN upgraded to 80 columns, increased memory, and increased operating speed.
- 6. The MYARC DISK OPER-ATING SYSTEM.







There are so many features included in the **GENEVE 9640** we can not list them all in this brochure but three more you will want to know about are: built in real time clock with battery backup, 640K RAM, 12MHz clock speed and the **GENEVE 9640** is compatible with the vast majority of T199/4A software.

The companies listed below are the MYARC dealers who are stocking and have available for immediate shipment the GENEVE 9640. For more information or to place an order contact:

AMERICAN COMMUNICATIONS 428 Jean Wells Drive Goose Creek SC 29445 (803) 797-5033

CITRONIC TECHNOLOGIES 981 Townley Avenue Union, NJ 07083 (201) 686-5619

COMPU MASTER 112 Fairmead Road Louisville, KY 40207 (502) 896-1935

COMPUTERS & CRAFTS 1418 Biscayne Little Choot, WI 54140 (414) 788-6656 after 6 PM CST

DHEIN'S TRUE VALUE 7 West Airline Hwy. Waterloo, 1A 50703 (319) 236-3861

DISK ONLY SOFTWARE P. O. Box 4170 Rockville, MD 20850 (301) 369-1339 EDUCOMP 6516 O'Henry North Bridgeville, OH 44039 (216) 327-6579

GREAT LAKES SOFTWARE 804 East Grand River Avenue Howell. M1 48843 (517) 546-0566

HUNTER ELECTRONICS 4N 370 Pine Grove Bensenville, IL 50106 (312) 766-9503

L & M SYSTEMS 2330 East J-8 #173 Lancaster, CA 93535 (805) 948-1587

LAFLAMME & WRIGLEY WHOLESALE 5480 Canotek Rd., Unit 16 Ottawa, ON, Canada K1J 9B1 (613) 745-2225

LYBROOK INDUSTRIES P. O. Box 1014 Wolfforth, TX 79382 (806) 866-9892 NOVA COMPUTERWARE 52 Airport Road Edmonton, AB, Canada T5G OW7 (403) 452-0372

PALACE SIMULATIONS 13 Hill Street Morristown, NJ 07960 (201) 292-1306

PLAZA COMPUTERS 424 Plaza Drive Birmingham, AL 35235 (205) 836-7608

QUEEN ANNE SHOPPE 6-1/2 Boston Street, Room No. 4 Seattle, WA 98109 (206) 283-0953

ROBINETTE & ASSOCIATES 2850 Delk Road, Apt. 39A Marietta, GA 30067 (404) 953-4584

SEEBER COMPUTERS 10668 Kaufman Road, NE Silverton, OR 97381 (503) 873-3140 SPECIALIST IN 821 Main Street Hopkins, MN 55343 (612) 938-3161

SUPERSOFT 34 California Drive Rochester, NY 14616 (716) 663-0397

TENEX COMPUTER EXPRESS P. O. Box 6578 South Bend, IN 46660 1-800-348-2778 (In Indiana 1-800-225-6838)

TEXAMENTS 53 Center Street Patchogue, NY 11772 (516) 475-6463

VIDEO ELECTRONICS MARKETPLACE 458 Pleasant Street Brockton. MA 02401 (617) 559-8935

VIDEOVISION 236 Upton St., P. O. Box 183 Grafton, MA 01519 (617) 839-4134

TI/Reg. TM Texas Instruments Inc.

MULTIPLAN/Reg. TM Microsoft Corp.

MYARC, Inc. P. O. Box 140 Basking Ridge, New Jersey 07920 Phone (201) 766-1700

NO ONE IS COMING AND IT IS 3 A.M.

It is a basic tenet of the unwritten social contract by which we live that all citizens have equal rights unless they do something to forfeit them. It follows that all are entitled to have their fellow citizens respect those rights.

Just as our system accepts that people are innocent until proven guilty, we informally accept that they are worthy of respect until they lose it. Not only does respect call upon us to take heed of the interests of others, it also decrees that we refrain from interfering with them.

To respect people generally is to avoid degrading or doing them injury when they have done nothing to us.

Any list would have to include attainment, integrity, fortitude, wisdom, skill and courage. Above all, we respect something called 'character' which brings together all of these traits. The spread of skepticism has reached the point where it has become part of our culture. Books debunking beliefs regularly reach best seller lists. 'Now It Can Be Told' accounts of famous events reveal dishonor, pettiness and incompetence among revered historical figures.

WHY I STOP AT THE STOP Out of this has come a near reversal of the theory that a person is innocent until proven guilty - at least in the informal court of public opinion. Anyone who does anything is called to the dock of the mass they do something media, where a thoughtless slip in phrasing can sentence a reputation to death. This may be unfair, but it is a reality.

In a democracy the people are ultimately responsible for the state of affairs. Anybody who complains there is no respect anymore might as well be talking into a mirror. If, as some say, children don't respect their parents, if students don't respect their teachers, if there is disrespect for the law, we should all ask ourselves where it all begins.

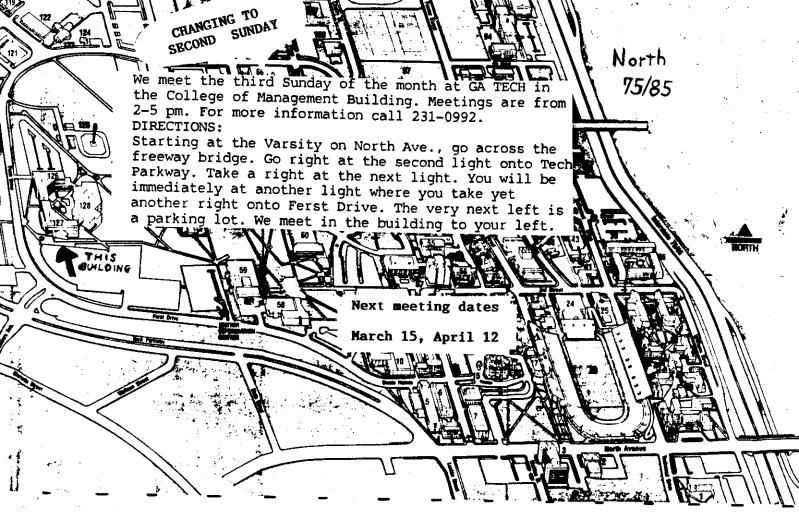
It remains for society to build a new and more solid base of self-respect. This will only be by placing a higher premium on individual character. We cannot demand character from others until we first demand it of ourselves; for each of our lives is a thread in the fabric of mutual respect of which our society is composed.

And that is why I stop at the stop sign at 3 A.M. when no one is coming, and why I pay for programs when I could get them free.

(From: SFTI Newsletter - Attributed to: THE ROYAL BANK LETTER)

BALLOT FOR OFFICERS

PRESIDENT	Gary Matthews	11
	Write in	11
VICE PRESIDENT	Jim Hubbard	
	Write in	1
SECRETARY	Melvin Carter	
	Write in	
TREASURER	George Sears	
	Write in	



ATLANTA 99/4A COMPUTER USERS GROUP POST OFFICE BOX 19841 ATLANTA, GEORGIA 30325

BULK RATE
U.S.POSTAGE
P A I D
ATLANTA, GA
PERMIT No. 492

UG EX IN Miami County Area 99/4A H.C.U.G. P.O. Box 1194 Peru, IN 4697Ø

DUES ARE DUE THIS MONTH
DUES HERE DUE LAST MONTH
THIS IS YOUR LAST NEHSLETTER