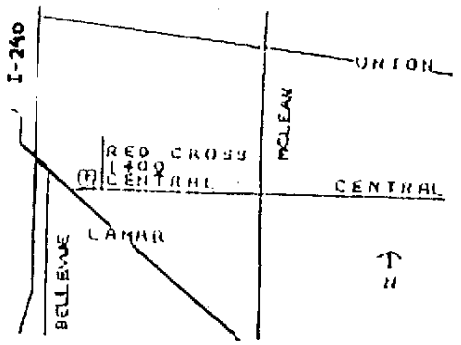
NOTICES

MEETING
7:00 P.M.
Thursday, MARCH 19th
Rec Cross Building
1400 Central Av.

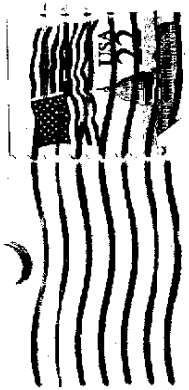
WORKSHOP
9:00 - 12:00
Saturday, MARCH 28th
Al Doss'
4284 Leatherwood
Phone 743-6781



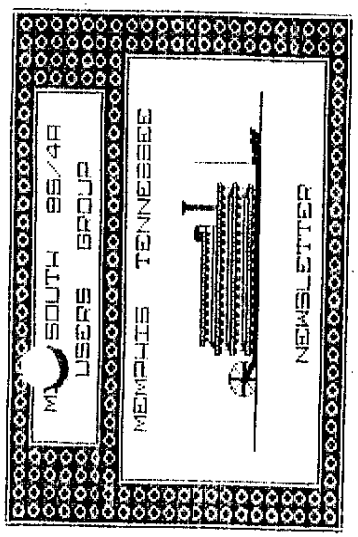
MEMBERSHIP APPLICATION

NAME _____ ☐ $15.00 FAMILY
ADDRESS _____ ☐ $10.00 JUNIOR (under 15)
CITY _____ ST ZIP _____ ☐ $10.00 ASSOCIATE (N/L only)
PHONE (___) - _____ :INTERESTS _____
EQUIPMENT,ETC. _____

Detach and mail with check payable to: Mid-South 99 Users Group, P.O. Box 38522, Germantown, Tn, 38183-0522.



N 2/87$$
FLOYD PEMBERTON
481 SPRUCE
LEWISVILLE, TX 75067

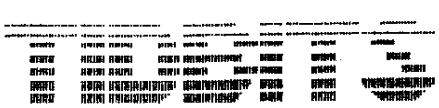


P.O. Box 38522, Germantown, Tn 38183-0522

MARCH INDEX

Equipment Problems.....2
XB Tutorial #6.....4
Status Reprot.....9
Program Bit.....10
Shoppers Corner.....11
Tibbs.....11
Logic.....13
In The News.....18
Modem Tips.....20
Max/RLE Picture.....22

Newsletter for the MID-SOUTH 99 USERS GROUP Vol 5, #3 March 87



EQUIPMENT PROBLEMS

Occasionally I get call by someone having equipment problems wondering where to get their equipment repaired. Texas Instruments still provides full repair or exchange services for all equipment made by Texas Instruments. Just call 1-800-TI-CARES for instructions on sending your equipment to Texas for repairs or exchange. No one in town will repair TI equipment. However, before you send it to Texas make sure you are sending the correct part. Since there is no longer a TI service center here in Memphis to diagnose your problems I would suggest isolating the problem by process of elimination. So let's say you are getting a lot of read/write errors with you disk drive. The problem could be in your disk drive, controller card or maybe even the power supply. The easiest way to find where the problem is is by process of elimination. First try your disk drive in someone elses system. If it has the same problem then your drive is probably the problem. If it works ok then try your controller in someone elses system. If it works then the only thing left is you power supply so then check it with a volt meter and see if you can determine anything... In other words just try out your system piece by piece on someone's elses system to verify what parts are working and the part left is probably the one that is not working. If you need to try out something just give Al Doss, me or one of the officers a call and we will do what we can to help you find the problem. Also one of the reasons for the Saturday workshop is to help anyone who is having equipment, software or problems of any kind. Sometimes I can even isolate your problem over the phone if you give me a call as I have seen most of the most common problems and can usually give you a good idea of what is wrong.

Another good idea is to have some backup equipment especially if you use your system for business. Most of the officers own two consoles as when the console goes everything is useless! Modulators seem to be a popular item for going out nowdays. Radio Shack once carried video modulators for $5 but have long sirce sold out (I have checked every Radio Shack in town). If your disk drive goes you I would suggest just purchasing a new one as it is usually cheaper to buy new drive than to get Texas Instruments to repair it unless you can find a place elsewhere that will repair them. Computer Shopper has disk drives priced so low that you can get a used one (MPI) for $98 and that is double sided/double density and maybe a new one for as low as $69. However another item Radio Shack does have is the TI keybcard (keys on the circuit board and these are orignals that TI used) for $3. I replaced my keys with this keyboard as my keys started double striking. Make sure you get the one that says made originally for the TI99/4A and the modulator also says the same. They also have the interral console power supply...

Simple problems can be resolved yourself such as cleaning the contacts (call me if you do not know how) and this helps system lockups.... A widget (cartridge expander) also helps a lot on console wear and helps the contacts from getting as dirty from pulling cartridges in and out. With a widget you can have three cartridges plugged in at once and you use a switch to switch between cartridges. I highly recommend one of these devices if you do not have a Gram Kracker. Also I have found that in some of the old consoles the foam in the edge connector in the module port sometimes comes loose and falls back into the contacts blocking contact between the computer and the cartridge. At the January meeting when I had my console apart I took out the foam in mine to eliminate any problems. The module port is usually the cause of most system lockups and startup problems. I have cleaned mine quite often...

Another problem you might encounter is errors when you initialize a new disk. This shows up as sectors used as the computer locks off sectors it is unable to write to when it verifies your disk during initialization. This is why you should always select to verify your disk when you use DM1000. Note that it is normal for a USED 2 to appear on DM1000 as actually 2 sectors are used for the directory. The disk manager cartridge just does not show you this as it will show 0 used. By the way if you are still using the Disk Manager II cartridge I would suggest getting a copy of DM1000 from our library. DM1000 is so much better than the cartridge it is like the difference between night and day! So if you get errors when you initialize several disks that you know are good you may need to clean your drive. NEVER clean your drive unless you need to as as the saying goes "don't fix something that does not need fixing". Manufacturers of drive cleaning kits recommend cleaning once a month or more but they just want to sell you the kits. Cleaning the drives that often would do more harm than good as the surface of the cleaning disks are abrasive (like sand paper) and will just wear off the head eventually! So if you get these errors and you have an old drive or one that is used a lot then it is time to get a cleaning kit. You can buy one at most any store like Target or any place that has computer equipment. Do NOT buy the dry system always get a liquid system.

One more special note on handling disks. Always treat disks with the utmost care. I have seen people who through them around and lay them everwhere. Many times disk errors, read problems and data loss can be traced to improper handling of the disks. There is only three places your disk should be. One is in the disk sleeve, in the disk drive or in your hand going to or from the drive to or from the sleeve. It is best to store the disks in a upright posisition in some type of case. Keep them away from ANY magnetic field such as the TV, telephone (it has a motor in it for the ringer which has a magnet), speakers, Rick Glisson etc... treat them with respect and you will have fewer problems!

Speaking of respect also respect your computer. Keep a dust cover on at least the console. I keep my environment around my system as clean and dust free as possible...

Although the computer was designed to operate in the home environment taking extra care of the disks and the system might

cause you less equipment problems in the future. But most of all PUT A SURGE PROTECTOR ON IT! Although some parts of town have pretty clean power I know a guy who lives in a really old house with terrible wiring and his voltage really varies and his system was blown once from a surge. My printer was blown by one once which sold me on surge protectors as I did not have one at the time. Circuit breakers, fuses and even with your system turned off will NOT stop a power surge. Fuses and circuit breakers just can not react fast enough to stop a surge or if it is for just a few nano-secords it will not react at all. A surge protector will not stop a direct hit by lightning but it does provide invaluable protection. Surge protectors can be purchased in a cube form which can plug into the end of your existing power strip (power strip protectors are also available) for under $10. I even bought one for my phone line as my modem is connected directly to the phone line (direct connect modem) and a surge can come through the phone line the same as the power line. Surge protectors for phone lines cost a little more than power line protectors and are harder to find but Stewart Software (locally) carries them at a resonable price as well as Tenex. Sometimes you can even buy a power strip with a power surge protector and a modem surge protector... I hope this helps you in dealing with and preventing equipment problems...Gary Cox

XB TUTORIAL NO. SIX

By Funlweb Farm
Of Australia

(continued from last month)

For the next few sessions we will continue with topics which are of immediate relevance to console-only users, namely squeezing programs to fit in memory, and extracting maximum speed from XB. My policy so far has been to concentrate on those parts of XB which are not already included in console Basic, not well documented, and not as widely appreciated as they should be. The next few tutorials will be on getting the most into and out of the machine while using XB. On the other hand I can see no point, and have even less interest in writing about, say, SOUND or SPRITEs from the very beginning, as these are fairly well documented in the manuals and the subject of many books and articles. That isn't to say that subtleties in using them won't come up from time to time.

Of late I have been working with Microsoft Basic and Turbo Pascal on CP/M machines with Z-80 processor in science laboratory applications. I must say that the more I see of Microsoft Basic the more I regard it as a cancer that should have been eradicated years ago when computers grew up to have more than 8K of memory. It is only now with their Apple Macintosh version, judging that from reading magazine hype, that they have at last surpassed the level of expressiveness that XB had years ago. TI did a lot of things to screw up this machine, some of historical origin, some quite intentional, but it takes coming from the engineered TI-99/4a file and device handling system to CP/M to make you realize how weak and primitive CP/M is in this area. On the other

hand Turbo Pascal almost makes that Swiss straight-jacket feel comfortable, and even CP/M doesn't seem so bad with Turbo. An excellent product at a realistic price that makes one realize that pirates are only the minor league of brigands in the software business. We can only dream that someone will bring out a native code Pascal anywhere near as good as Turbo for our machine. Shed a tear for TI-99 Basics with their two layers of interpretation (Basic and GPL), on top of working indirectly from the byte oriented VDP memory and GROMs.

Enough raving on for now and on to the real business. Let's now look at how to face up to that 'MEMORY FULL' message. This even comes up when you have memory expansion with a total of 48K of RAM in various guises. Programs always seem to end up needing more memory than is available! I do feel some unease in discussing this topic as many of the things that are done in compacting programs can only be regarded as poor program practice otherwise, as they make the code obscure and difficult to modify or develop further. The other great trade off that must be considered when scrunching programs is speed of execution. Given an equal level of skill in program writing, coding for speed usually results in a longer program than would otherwise be written. Perhaps the easiest example to see is unrolling of a short loop which is repeated a fixed number of times. A FOR-NEXT loop gives compact code but carries a penalty of the loop overhead which could be avoided by writing out the contents of the loop the appropriate number of times. The richer the language the more ways there are to optimize code to do either. Console Basic offers many fewer ways to do this than does XB and is much less fun.

At what stage should you bother trying to make your code compact ? Remember that XB can only OLD or RUN one program at a time, so apart from loading time from cassette, or disk space, there is no reason at all to scrunch a program that runs in the smallest memory it is intended to run on. Most users with disks now have the 32K memory expansion, so this means the bare console. Minimum Basic programs to store in the module's RAM are the only ones you have real incentive to make smaller still. Unless you know from the start that you are going to run short of space because of large arrays of numbers, or a need for maximum string storage room, be expansive -- document your program thoroughly with REMs, use lots of SUBprograms, use obvious explanatory names for variables, avoid reusing variables for unrelated uses ..... and then you run out of room.

Now first of all a program has to be short enough to load. This is purely a function of program length. Next it has to be able to complete prescan when RUN. For prescan to succeed there must be enough room left over after the prescan for variable pointer and subprogram tables to be set up, and room set aside for numeric values, at 8 bytes per number. String variables are not assigned space until it is actually required, so it is possible for a program to crash later because it can't find enough room for strings. The well known hiccupping of long Basic programs occurs while Basic scratches around to reclaim string space when it has run out of new space. XB does it too, but it is a lot faster at 'garbage collection'. Now let's look at how to squeeze programs in, starting with things that affect the program length only.

The most obvious thing to do is to remove REMs from your program. I would suggest that this be left till later in the development process as you put them there in the first place to help. At the least keep some for now. If you have been following earlier Tutorial advice to use lots of clearly named subprograms then you don't need many REMs. For the same reasons you should not abbreviate subprogram names beyond recognition at this stage. Basic as an interpreted language, where the source code is also the run-time code, has this problem that commentary and explanation are not eliminated by a compiler or assembler and compete for memory space with the executing program. One way round the problem is to restore REMs to a file copy after intensive development is over, even if it does make it too long to RUN. The REMS can always be removed later.

Now it's time to look at what makes an XB program as long as it is. To get started let's look at two very short programs to clear the screen.

```
100 CALL CLEAR
```

Before entering this clean up the machine with NEW and SIZE it. Then enter this program and SIZE it again. The difference will be the length of the program 13928-13914 = 14 . I will mostly quote SIZEs on the basis of a console only machine for simplicity, but there are some interesting differences. With memory expension XB lists high memory and stack separately, and ignores low memory altogether. XB stores the program and numeric variables in high memory (24K), while the stack - 12K of VDP memory - contains variable descriptions, subprogram details, PABs, and the string storage space. This ALL has to fit in 14K of VDP RAM with XB/console only. Console Basic doesn't use memory expansion for Basic at all. Now try a second program which does almost the same thing

```
100 DISPLAY ERASE ALL
```

and SIZE it. Only 9 bytes now ! Although the LIST of this second program is longer, the computer thinks it is shorter. Consult your XB manual p40 where you will find all three words DISPLAY, ERASE, ALL are listed as reserved words, as is CALL but not CLEAR. Reserved words are treated differently -- when you enter the line they are recognised and "tokenized" as one byte symbols with ASCII values >127. 'CLEAR' takes 7 bytes, one the token for a string without quotes, one for a length byte, and 5 for the string itself. Why use tokens ? For one thing it shortens the program length, and also makes it easier for the interpreter to recognize them when the program is running. XB's range of tokens is very limited and built-in subprograms are the way XB gets around this.

Now you don't have to take my word for this. If you have an expanded system you can write programs using CALL PEEK to explore stored programs, or better still use the E/A DEBUG (reassembled uncompressed object code so the XB loader can handle it) for a quicker look. With console XB you can at best get an indirect insight by entering <CTRL+various keys> in a REM statement and LISTing that. Be careful, you can crash the computer in ways wondrous to behold that way Someone forgot to tell the computer

not to try to turn token values back into reserved words when LISTing REMs. Ever notice when writing file specifications that keywords that do extra duty elsewhere LIST with the extra space, but the others do not. EASY-BUG in Minimem also allows you to look directly into VDP RAM or cartridge RAM to see Basic programs their internal state.

In TI Basics, unlike those which store programs as ASCII files the line number is always stored as a 2 byte integer, and it makes no difference to program length to use line #1 or line #10000. Try various line numbers in one of the examples above. If you are peeking around in the program, don't expect to find the line number at the start of its program line. It is in a separate table below the program, and each 4 byte entry has the line number followed by the location of the line itself. The line # table is sorted into order, but new or edited lines are always added to the lower address end of the program block. The program lines themselves are preceded by a length byte and terminated by a null (>00) byte. I won't go into it here but you can use this general information to interpret the various time delays when you edit a line or enter a new line.

From this you can see that there is a 6 byte overhead associated with every new line number. Now enter the program lines above as lines #100 and #200 and SIZE. Next combine them as a single line

```
100 CALL CLEAR :: DISPLAY ERASE ALL
```

SIZE again. There is a saving of 5 bytes. The reserved word "::" has cost 1 byte, but 6 bytes have been saved by having one line fewer. Now if you scrunch a 500 line console Basic style of program into 200 XB multi-statement lines you have gained 1500 bytes. Of course you can't do this to every line because line numbers, as well as being line editor markers, are also where GOTOs and GOSUBs go, so you will usually end up with a few short lines you can't condense. FOR-NEXT loops work perfectly well within or across multi-statement lines. The use of prescan switch commands is costly because you end up with !@P+ and SUBEND on separate lines at the end of each subprogram so treated. Still, it's usually worth doing even though a long program may have several hundred bytes tied up in prescan switching. In desperation at the end you can always remove prescan switches starting with the shortest subprograms.

How much room does a variable take up ? Take a simple numeric variable. There are 8 bytes for the radix-100 floating point form that both TI Basics use for all numbers (they even do 1+1 to 14 significant figures every time - another reason they are slow). Next the interpreter has to be able find where this value is stored so there's 2 bytes for a pointer to the value, and 2 more point to the name associated with this value. Further it has to record the nature of the variable, whether it is numeric or string, simple or array, DEFed or normal. Also in a Basic language which allows long variable names a length record is also likely, though not absolutely necessary. All told there is a practical minimum of 14 bytes of overhead for every simple numeric variable.

As I have noted in other connections in this series, TI in its self-defeating secretive way, never explicitly specified the details. TI Basic is most likely highly consistant in this from model to model, because any console can be called on to work with separate E/A or Minimem Basic support utilities such as NUMREF. On the other hand each XB module contains its own set of support utilities, and only has to be internally self consistent. There is information in TI's published data (XB, E/A, Technical manuals), giving details of VDP stack entries built by the E/A CALL LINK with some hints as to changes for the XB version. So to use XB LINKs at this level of detail you have to work by implication. Now it is done from time to time, but TI does not seem to have guaranteed explicitly to programmers that such procedures would work with all XB modules, or that the LINK stack entries are similar to internal table entries. Most likely they do and are. Only TI knows for sure Then again TI lost big while Apple and IBM make lots of money being more open about their machines, though Apple seems to be developing more secretive ways as it gets older and more arrogant.

Time for some little program experiments again. Enter the miniscule program

```
100 A=0
```

Before you do anything else work out how many bytes this uses. The answer is 11. In accepting the line the editor has already figured 'A' for a variable (because it starts with an allowable character) and not a reserved word and it is represented exactly as it occurs, no token involved. On the other hand it doesn't give a care that '0' is meant to be a number and treats it as an unquoted string. If it isn't an honest number, say 2N, it will only find out later when it RUNs and tries to convert it to a floating point number.

SIZE the program, then RUN it and SIZE again. XB does not reset everything until you have made an editing change, as you know from debugging efforts after BREAKing (Fctn-4) program execution. At this stage you get more information from an expanded system, which will show 8 bytes of memory used and 9 bytes of stack. Now repeat the process with a longer variable name. The length is reflected both in the original program length and in the stack used. The stack usage is 2 bytes plus the variable's name length more than the minimum we figured out before. Most likely the 2 bytes are for a linked list structure to help table searching, and there is a symbol table entry of the variable name. Now turn off your expansion system and be like everybody else with console only, and repeat the above. Now you will find the increase over the program length is always the 14 bytes we figured earlier no matter how long the variable name is. Now try

```
100 A23456789012345,A23456789012345=0
```

Still 14 bytes RUNtime overhead ! Change the second A to a B to make a distinct variable name, 15 bytes long. Only another 14 bytes overhead ! So how come ? Maybe it's doing without the full word list link and squeezing things up a bit, but what about the symbol table ? The only consistent conclusion is that it doesn't

have one as such, but points to the first location of the variable name in the program as located by the prescan. Read the Tutorial on prescans again. XB always searches in VDP RAM for variable names even if the program itself and the numeric values pointed to are stored in expansion memory.

If you wanted to make a faster interpreted Basic, you would, in the prescan, replace all variable names by some token plus a storage pointer to eliminate table searches. Which is just what TI claim in their Software Development Handbook to have done with the Basic for their 990 minicomputers. Unfortunately they failed to make an honest machine of the 99/4.

(to be continued next month)

STATUS REPORT

If you missed the February meeting you really missed a good one! Although many regular attending members were missing for various reasons attendance was still heavy with a croud of about 50 people and excitement was high as about 8 or so new prospective members came to their first meeting following a very generous mention of our group by the Commercial Appeal (local paper) the Wednesday before. This was our first mention by the paper in the regular section (the part everyone reads) following a letter I wrote to them asking for assistance in informing people that our group is here to help TI99/4A owners with support, information and technical assistance. We even had a visitor from the Dallas II ers group who was in town on business. We are hoping that he will be able to attend our meetings often so as to trade information and programs between our group and theirs.

I would like to apologize for the technical problems that we had as we were short of some cables and a disk froze in it's jacket during the Multiplan Tax demo. This is the first time I have ever seen a disk freeze in a jacket like that!

As for a status report of the group we seem to be doing well. Membership is down from 120 members about a year ago to about 95 members now with our newsletter going out to over 150 people and user groups total. Most of the members that have dropped off are the ones who had only a passing interest in the TI99/4A and never participated in any group activities etc... Our treasurer reports about 1,000 dollars in our treasury. Most of the money in the group goes toward production of the newsletters ($40-$50) and for the phone bill and maintaince of the club TIBBS ($30). Then miscelaneous expenses include library disks and other small items that are necessary.

I would like to thank all the new people for comming to the last eting and I hope that you decide to join us. The purpose of our roup is as I mentioned is to provide support and assistance to TI99/4A owners. We have a library full of recent Public Domain and FREEWARE software available to all members free of charge (except for $1 if you want to buy our disks). We also operate a 24 hour electronic bulletin board which is accessible by a telephone modem and any brand computer system where you may commuricate with other club members and receive programs by

transfering them from our system to yours using a modem. At our Thursday meetings we have our library available, newsletters from other groups available for reading, a for sale table where people bring items they have to sell and we carry on demonstrations of equipment, software and various tutorials. Our Saturday workshop is for people who want help with a specific problem, want help c running a program, need a diagnosis of equipment problems or jus want to sit around and talk about the TI... Then every Thursday except our meeting night TI99 programming classes are held by Dick Vanderberg.

Before I end I would like to thank Gerald for his kind comments in his article last month about me. I to all that I do not only because I am dedicated to the TI99/4A but because this group was partially responsible for the knowledge of computers that I have and I now like sharing it with others... I would like to point out though that many others in the group work behind the scenes to keep the group running. These dedicated people produce the news letters, run the BBS's, write programs, handle group business etc... Let's not forget these people who continue month after month giving their time and effort to make our group the best one possible!

I would like to end by wishing Jonathan Leslie a fairwell as he moves to Kansas City Missouri. Jonathan has been a really enthusastic and helpful member and he will be greatly missed! However, I might add that this move does not get him off the hook for writing articles for the newsletter! ... Gary Cox - Secretar

PROGRAM BIT

6:00pm Doors Open

7:00pm - 7:15pm General Discussion.

7:15pm - 7:35pm Video taped presentation made by Texas Instruments on how computer chips are made. The tape goes through the different processes involved in developing and producing a chip. The tape is non-technical and is quite interesting what all must be done to make a small computer chip!

7:35pm - 8:05pm Demonstration of a program called Fuzzy decision maker. This is a program which helps you make decisions based on mathmatical principals of decision making. Programs just requires XB. Demo by Ralph Wilson.

8:05pm - 8:15pm Demonstration of the new YAK-MAN Talks menu driven speech program by Trinity Systems. Runs in TI BASIC with TE2 cassette recorder and speech synthesizer. Demo by Gary Cox

8:15pm - 8:50pm Demonstration of the new Triton Turbo XT IBM/TI99 compatible system engineered by MG (Millers Graphics). There has been a lot of controversy on what the machine will do, how it works with TI99/4A and why you need it instead of just going out and buying a IBM clone. We hope to determine the answers to those questions and more... The system is on order and is expected to

arrive before meeting time (we hope!). Demo by Mec Swope.

8:50pm - ..... General play period.

11:00pm Doors Close

As a closing note, the Geneve did not arrive last month and the person who has it on order has moved out of town. Hopfully we will be able to obtain one soon from some source for a demonstration. This will be a super meeting so don't miss it...Gary Cox - Program Chairman

SHOPPERS CORNER

Since several members have had their video modulators go out recently, the club has decided to purchase some extras, which we will be selling for $5 a piece. Radio Shack once carried them but have long since sold out. With the exception of a place that occasionally has them in the Computer Shopper they usually cost from $19-$30 from TI. A limited quanity will be available but if enough people are interested I might be able get some more.

For sale, TI Professional Computer (partially IBM compatible but not 99/4A compatible) with 2 DS/DD drives, green screen monitor, corner type computer desk. Asking $1300. For more information all Mrs. Norma Hunt at 901-353-5350 M-F after 4pm and all weekend.

Looking for a great modem? Try the Avatex 1200HC! Available from Lyco computers at 1-800-233-8760 in the $110 price range. It operates at both 300 and 1200 with auto dial and auto answer capabilities and is fully Hayes Compatible. I doubt you could find a better buy. (if they do not have a cable for the TI99/4A you may have to switch pins 2 and 3 but someone in the club can help you with that). Several people in the club have one and like it very much.

Games "N" Gadgets in the Mall of Memphis still carries TI compatible software as the only place in town who carries anything specifically for the TI99/4A.
Check out the For Sale/Trade table at each meeting for bargins...

TIBBS

For those of you that do not have a modem, I would like to encourage you in that direction. I will first start by describing what you will need, if what you now have is only console, tape, or bare bones system.

First of all, some type of expansion is needed. Most importantly is having the 3 major expansion components. These consist of a memory expansion, a disk system, and an RS232 interface. There now are several ways to add these to your console. The most

popular is going with an original TI-expansion box, or boat anchor, as it is often called. You will most likely have to get this one used as TI no longer makes them. The other options are the mini expansion system which contains all three of these components. The only drawback, if any, is that with a mini expansion system is that it cannot be expanded. This can be overcome since most expansion cards are sold as stand-alones, at a slightly higher price tag. These stand-alones would plug into the side of the mini box. Myarc and CorComp both sell these mini systems. I have seen a recent ad for the CorComp system for about $380. Another alternative is a IBM styled expansion box sold by Ryte Data of Canada. It is called the 99AT expansion. This box will house up to 4 disk drives and will hold 5 standard TI expansion cards. This box sells for $155. Although you would still need the 3 cards also, the ad says it is compatible with all current TI products.

If you already have these 3 major expansion options then, you are well on your way to getting online. All that you need now is a modem and a terminal software. Both of these are a little bit easier on the budget than a expansion route. First of all, there are several terminal software packages that are 'try before you buy" available for the asking. All that is required is that you compensate the author if you like and use the program. This fee is always very reasonable and should be sent to the author for his labors. Last, but not least, is the modem itself. A modem is simply a converter. It converts the digital signals of your RS232 to analog tones for the telephone network for outgoing signals. For incoming, it converts the tones from the modem that you are connected to, back to digital signals so your RS232 can decode the signals and process them so that eventually the signals go to your screen as a character. There are modems that will suit everyones needs. I will reserve an article on modems for next time.

You say, "What do I do once I get my modem?". Well, that is a good question. This group sponsors a very popular BBS called TIBBS. BBS stands for Bulletin Board System. This is a computerized version of a Bulletin board. There are many other things to do as well. I will concentrate on TIBBS at this time. TIBBS awaits your call 24 hours a day and 365 days a year barring power interruptions. I have been the System operator (SysOP for short) for two years this March. We have come a long way and have logged quite a bit of activity.

Currently TIBBS has 4 double density double sided disk drives, a CorComp Disk controller, a CorComp Triple Tech card which has a 64k print buffer and a battery-backed clock/calendar function, an ink jet printer, and a CorComp 512k memory expansion card, and a 300/1200 baud Uniqmodem. We are currently trying to add a hard disk system. We desparately need your donations for this expansion. You can mail your donatins to me or give them to Gerald at the meetings. My address is on TIBBS in the bulletin section.

TIBBS currently has recorded over 9,500 messages and logged over 17,000 calls! There are over 100 files in the transfer section From which you can transfer to your computer. The message base has been expanded to accomodate 18 lines of text rather than 12 as

the software was originally written. I have recently added a second mini message base that is totally seperate from the ongoing message base of 100 messages. This mini message base is an ongoing story where anyone can add to their two cents worth. Everyone is having a ball on TIBBS. So why not "get online" and join the fun. The number to call is (901) 357-5425...Pierre K. LaMontagne, SysOP.

LOGIC

From the Chicago User Group's Rich Klein, here is a great article and accompaning program:

If you've ever wanted to roll through a set of numbers continuously, such as [1,2,3,4,5,6,7,8,1,2,3,4,5,6,7,8...Etc] and you hated the logic required, then perhaps the following might be useful to you:

```
100 A=1
110 PRINT A;
120 A=A+1+(A=8)*8
130 GOTO 110
```

Line 120 is the meat of this routine. By letting the computer evaluate an expression and using that result within the formula, you can eliminate three lines (or commands) of programming. Let's see how this works:

100 Sets the variable 'A' to one.
110 First line of loop prints each value of 'A' so we can see how our formula affects it. The semicolon causss each value to be displayed next to the previous value.
120 Here is where the work is done. If we remember our Algebraic hierarchy, then we know that any expression in parentheses is evaluated first; then multiplication and division and then addtion and subtraction. Functions and powers are in there also, but we're not using them here. Let's go through and solve this formula step by step. Let's also give the variable 'A' a value of seven:

```
1) A=A+1+(A=8)*8 ORIGINAL FCTN
2) A=7+1+(7=8)*8 PLUG IN SEVENS
3) A=7+1+ 0 *8 SOLVE PARENTH.
4) A=7+1+ 0 SOLVE MULTIPLIC.
5) A=8+ 0 SOLVE ADDITION
6) A=8 FINAL VAL OF 'A'
```

Let's see how this formula works when 'A' equals 8:

```
1) A=A+1+(A=8)*8 ORIGINAL FCTN
2) A=8+1+(8=8)*8 PLUG IN EIGHTS
3) A=8+1+ -1 *8 SOLVE PARENTH.
4) A=8+1+ -8 SOLVE MULTIPLIC.
5) A=9+ -8 SOLVE ADDITION
6) A=1 FINAL VAL OF 'A'
```

You can see that if 'A' equals eight, then the expression, (A=8) yields a value of -1 since the expression is true. It is then multiplied by eight for a value of -8. When this figure is added into the rest of the formula, the value of 'A' is reset to one.

When 'A' was NOT equal to 8, then the expression (A=8) yields zero, which, when multiplied by eight, still equals zero. When this is added to the rest of the formula, the result is the same as if it wasn't there and the value of 'A' goes up by one.

As you can see the formula is a little misleading. If you try to solve it from left to right, you might get the impression that the variable, 'A', equals one more than it actually does and that it will be incorrect. If you remember the way expressions are evaluated (order of hierarchy), and also that the value of a variable is NOT changed until the entire expression is evaluated, then you should have no problem creating expressions of incredible complexity.

130 This line transfers control back to line 110 to create an endless loop. To stop this example, simply press <FCTN 4> (CLEAR).

This is a simple illustration of what you can do with logical expressions. While you were reading the above, I was working on a short routine which, I believe, adequately illustrates the concept of evaluating expressions. Let me explain step by step what is to be acomplished before presenting the routine in its entirety.

What I propose, is to create a counter of sorts, which will count seconds from zero to fifty nine, detect any carries and count up to fifty nine minutes. I'm not concerned with the accuracy of this 'clock', because the way we will construct it is more important. This counter will count units, carry over into the tens and count them up to five tens and nine units. From there, it will count in the minutes column, generate a carry into the tens of minutes column, and count from zero to fifty nine minutes and fifty nine seconds before resetting to zero. Some of you might think that this is a simple series of decision making statements (IF-THEN-ELSE). If it was, then it would defeat the topic of this column. I intend to accomplish this task without any conditional statements whatsoever. This will be accomplished with a series of mathematical expressions which are arbitrary in their very nature. That is, each statement will be executed in the same sequence each time and will generate a different (or the same) number each time.

The format for the display on the screen will be as follows:

00:00

Each digit will represent the value of different variables. Each digit will be placed on the screen with an HCHAR(row,col,char) statement. Each variable will be the result of one mathematical formula. Let's see how to construct this formula.

We'll figure our counter from right to left, since units that are on the right are lesser than those on the left. The first digit

(units, seconds) will be called 'A'. We need to count these units from zero to nine. First we'll initialize the variable "A" to zero as follows:

```
100 CALL CLEAR
110 A=0
120 B=0
130 C=0
140 D=0
```

It is not actually necessary to initialize variables, but it makes the listing more readable and is generally accepted because some BASICs require variables to be initialized before they can be used. TI BASIC assumes any variables (numerical) not initialized to have an initial value of zero. Line 100 clears the screen for neatness.

Now we need to come up with a formula to count from zero to nine and repeat. The formula in the last example is adequate with small changes to allow for counting zero. We still want a maximum value of nine, but we need a range of 10 digits (0-9). If we use the following formula:

```
160 A=A+1+(A=9)
```

If we bear in mind the previous example, and examine this formula we see that:

1) 'A' is the value we want to adjust.
2) '1' is the basic increment we want.
3) (A=9) generates -1 if true or 0 if false and indicates the max val needed for the digit.
4) 10 is the range of digits. When multiplied by our true/false value(A=9), adds 10 or 0 to our expression.

If A=9 then our expression would be 9+1+(-10) or zero. If 'A' was equal to less than nine, then our expression would simply be A+1+0 or A+1. So this expression is OK.

Next, we must figure out how to count tens of seconds. We must increment this digit once each time the units column resets to zero. We can use the original formula with more changes. Let's start with the original formula as we used for the variable 'A'. Instead, we'll use the variable 'B':

```
170 B=B+1+(B=5)*6
```

Remember 'B' must cycle between zero and 5, with a range of only six digits. So the expression (B=5) sets up the maximum value before the formula resets to zero.

But we want the variable 'B' to be incremented only when 'A' resets back to zero. Obviously, if we increment only under certain conditions, we can only reset under certain circumstances. What is needed is a way to "turn off" the part of the formula that alters the value of 'B' when 'A' does not equal zero. Hmmmm....

If you multiply a number or expression by zero, you get zero. If

you multiply a number or expression by one, you get the same number or expression. Bearing this in mind, and considering that a logical expression ((A=0) for example) returns a zero, if false, and a -1 if true, what if we could enclose the part of our formula that adjusts our variable, 'B', and multiply it by the logical expression (A=0)?

```
170 B=B+(1+(B=5)*6)*(A=0)
```

It's a step in the right direction. If A does not equal zero, then the expression (A=0) would return a zero, and when multiplied by the portion of the formula enclosed in parantheses would result in a zero as well, effectively turning off the incremental portion of the formula. It is solved as follows:

The only problem with this is that if the expression is true(A=0), then it will yield a result of -1. If we use this value, it will result in B being reduced by one instead of being increased by one. If we use the ABSolute value of the expression, (A=0), then if the expression is false we get zero. But, more importantly, if the expression is true, the negative sign is stripped from the result, leaving a positive one. This is exactly what we want.

```
170 B=B+(1+(B=5)*6)*ABS(A=0)
```

With this formula, B is only incremented when A is reset to zero. One thing to be certain of, you must be aware of exactly what any variable will contain at any given moment. If not, you may encounter a situation not allowed for in your formula(s). This is especially true when your formula contains more than one variable. If we solved the B formula before the A formula, then neither formula would work properly. Sometimes it is useful to construct a table with the possible combinations of all the variables contained in your formula and solve the formula for each different combination.

The next two variables, C and D look just like A and B except that when checking C we need to make sure that both A and B are reset, and with D, that A, B, and C are reset.

```
180 C=C+(1+(C=9))*ABS((A=0)*(B=0))
190 D=D+(1+(D=5)*6)*ABS((A=0)*(B=0)*(C=0))
```

If any fo the variables checked don't equal zero, then the absolute valie is zero. When multiplied by the other expression in the formula it too becomes zero. There is only one combination for these formulas that will produce a non-zero result. If you look closely at the 'C' formula, you may notice that the ABS() function is not necessary, since -1 times -1 equals 1, but I left it in for symmetry. You may remove it if you wish.

Well, so far we've got this

```
100 CALL CLEAR
110 A=0
120 B=0
130 C=0
140 D=0
```



```
+
+
+
+
+
200 A=A+1+(A=9)
210 B=B+(1+(B=5)*6)*ABS(A=0)
220 C=C+1+(C=9))*ABS((A=0)*(B=0))
230 D=D+1+(D=5)*6)*ABS((A=0)*(B=0)*(C=0))
+
```

We've got our housekeeping done (100-140), and we've got our formulas (200-230). How can we see the results of our hard work? Let's put some HCHAR() statements in between these sections to display the numbers.

As you may know, the first value in the HCHAR() statement indicates which row the character is to be placed, while the second is the column. The third value is the ASCII value of the character you want in the position specified. Since our variables contain actual values rather than ASCII values, we must add forty eight to each variable when in the HCHAR() statement as a character. The first HCHAR() statements will place a colon between "minutes" and "seconds". All that's left is to loop back through the program so it can run continuously. We will loop back to line 160 because it is not necessary to redraw the colon every time the loop executes.

This is the complete listing in twenty eight columns so that your screen should look exactly the same:

```
100 CALL CLEAR
110 A=0
120 B=0
130 C=0
140 D=0
150 CALL HCHAR(12,15,58)
160 REM
170 CALL HCHAR(12,13,D+48)
180 CALL HCHAR(12,14,C+48)
190 CALL HCHAR(12,15,B+48)
200 CALL HCHAR(12,17,A+48)
210 A=A+1+(A=9)
220 B=B+(1+(B=5)*6)*ABS(A=0)
230 C=C+(1+(C=9))*ABS((A=0 )*(B=0))
240 D=D+(1+(D=5)*6)*ABS((A=0) *(B=0)*(C=0))
250 GOTO 160
```

This clock runs a bit fast. About three of its "minutes" are counted every minute of real time elapsed. It wasn't intended to be accurate, rahter than to illustrate logical expressions. As the program is written, all the values are redrawn regardless of whether they are changed or not. This is because there are no conditional statements (IF-THEN-ELSE) in the program. All the lines are executed in exactly the same order each time and the loop execcutes in exactly the same amount of time each time. This is regardless of how many numbers are changed each time....

Rich Klein

I hope you enjoy this very informative article, reprinted from the Chicago Times. I have played with this little program and found that probably the most accurate timing device in TI Basic is the CALL SOUND() statement. I modified the listing of the program so that you could place a CALL SOUND() in line 160 replacing the REM statement. With the program in Extended Basic, I found that the following CALL SOUND() gave results very close to the real time. You may have to do more adjusting if you are running it in Console Basic. Since the Loop will execute every second you can subtract the time value of the CALL SOUND from one thousand and see how fast the logical expressions work....
Replace 160 with:

160 CALL SOUND(968,20000,30)

I used 20000 and 30 so that the sound could not be detected. All I can say is thanks Rich, for a very enlighting piece of programming expertise! Have fun making your own counters and using them in other ways in YOUR programs...Pierre,K. Lamontagne

IN THE NEWS

A lot of news this month!

Myarc is expecting to be shipping their new hard disk controller cards by March 15th 1987. Their new controller card will replace your existing controller card and will control up to three 5 1/4 inch IBM compatible hard drives with the capacity of up to 240 Megabytes of storage space! The hard drive controller will also control up to 6 DS/DD floppie drives. Myarc will be selling the controller card in the price range of $250 but will be offering a package deal where you can get both the hard disk controller, 20 MEG hard drive and connecting cables all together for one price. The controller uses a Unix-like directory tree structure so that organizing and locating files are easy. A built in date and time clock markes files with the time of their last update. A disk manager is included with the controller. Check with a Myarc distributor for exact pricing. (Tenex is already advertising it in their winter/spring catalog.) Gerald Smith already has the card and 2 hard drives on order and should have it real soon! This is really an advancement in storage capabilities for the TI99/4A. I would like one so bad that I can hardly type because of the slobber on my keyboard...

Not-Polyoptics has released their Spad XIII flight simulator. According to their description you fly a World War I fighter plane set in wartime France. The program is written in assembly language and uses a full 48K of memory. Features include an all-graphic world in 3D perspective, full acrobatic control of the airplane with algorithms that mimic all of the physics of flight, continuous instrument readouts with graphic throttle and stick controls, joystick or keyboard inputs, dogfights with enemy planes, bomb enemy hangers and down enemy observation balloons. Scenery includes the Eiffel Tower, Seine River, trenches, French

Villages, clouds and more... The program requires Extended BASIC, 32K and a disk drive. Cost is $41.95 from Tenex at 1-800-348-2778. I have one on order and will have a review as soon as I receive it.

The Triton Turbo XT TI99/IBM compatible system is to be shipped this month according to several reports. A lot of controversy has been brewing about the system because it seems to just be a Turbo XB interfaced with the TI keyboard in which case why not just buy a Turbo XB system by itself? We should have it to demonstrate this month so we can find out exactly what it's capabilities are and any possible problems that it might have. If you are thinking of buying one come take a look at it First.

From LGNA Products comes 99 Fortran Development Package. It programming environment using the Fortran language for the TI99/4A including an interactive full screen Editor, an optimizing FORTRAN Compiler, a Linker, a Symbolic Debugger, Save/load utility and user extendible object library. According to my information it provides more screen and sprite functions than XT and runs hundreds of times faster than XT. Requires 32K and either E/A, Mini-Memory, XT or TI-Writer Cartridge. Available on disk from Tenex for $49.95. Tenex also sells a language called Pilot which allows the writing of programs with just eight simple commands utilizing very fast speed... It sells for $49.95 and requires XT and 32K...

Byterraster Computer Services of 171 Mustang Street, Sulphur, LA 70663-6724 has been named exclusive distributor of the MG Explorer program which allows exploring 99/4A operations while programs are being run. Explorer is widely recognized as the premier aid for 99/4A Assembly Language programmers. Price will be in the $25 range.

The Bongard Group of P.O. Box 53171, Lubbock, TX 79453 is offering a hardware manual for the TI99/4A for $19.95. The manual describes the console design, custom chip operation, TMS 9900 R/W Organization, TMS 9900 Instruction Set, Interfacing Pitrall's, Console Schematics, PEB Card Descriptions, GROM Simulator Design and the Extended BASIC Module description and schematics.

Techni-Graphics of 1058 Perda Lane, Des Plaines, IL 60016-5724 has released version 2.0 of its PC-Keys program, a program giving the TI99/4A programmable control keys. The new version recalls for $23.99 plus $1.50 shipping according to Jim Kryzak of Techni-Graphics. The new version features a new utility menu, a 11 digit display calculator which uses a windowning feature, allows the user to add, subtract, multiply, divide and compute square roots, and has a memory key, a disk catalog which now uses a windowing feature and allows the option of printout to a printer, a note pad with a windowing feature which can be output printer, and a setup utility program for such functions such as set output device, edit keys 1-9 and change screen color.

CSGD user disk #4 has been released by Texamonts. The disk which costs $10.95 plus $2 postage contains 16 new fonts and 61 small graphics... The disk requires the CSGD programs. For more information contact Texamonts at 53 Center St., Patchogue, NY

11772 (516) 475-3480.

The Orphan Survival Handbook has been shipped this month. It is a collection of material from user group newsletters in a 200 page loose leaf book which can be placed into a notebook and added to as updates arrive. The handbook contains schematics, hardware hacks, programs, tips and tutorials and sells for $16.95 from Disk Only SoEtware, P.O. Box 4170, Rockville, Maryland 20850. (301) 340-7179. I hope to have a review of it in this months or next months newsletter.

The New England TI Faire will be held April 4, 1987 in the Boston area. For more information contact the Boston Computer Society, TI99/4A Users Group, One Center Plaza, Boston, MA 02108 or phone Walt Howe at (617) 692-2702 or leave a message on the communications networks: CIS #70277,3530, GEnie WALT.HOWE SOURCE TIA054.

The TI99/4A Fest-West 87 will be held May 16 and 17 in Los Angeles. For information contact the LA 99'ers, P.O. Box 3547, Gardena, CA 90247-7247 or call Terrie Masters (Prez) at (213) 271-6080.

March 28, 1987 TICOFF (Texas Instruments Compter Owners Fun Fest) will be held in Roselle Park, New Jersey. For information write to: TICOFF 87, c/o Bob Guellnitz, Roselle Park High School, 185 W. Webster Ave., Roselle Park, NJ 07204 or call Art Byers at (914) 528-5402, Guellnitz at (201) 302-5963 or the TICOFF BBS at (201) 257-2607.

If you are wondering where to get Joypaint 99 reviewed by Jonathan Leslie last month Tenex has it for $34.95. I also have it and it is truely a great program! I would suggest though getting Joypaint Pal as well which lets you save and load pictures in other formats... Requires XB, 32K and disk system.

Tenex has mailed their winter/spring catalog full never before seen and exciting software and hardware. They also have new versions of old programs and price reductions on others. If you do not have one of their 50 page catalogs call them at 1-800-348 2778 or (219) 259-7051 and ask for a FREE catalog...Gary Cox

MODEM TIPS

FAST-TERM TIP:

A common problem in XMODEM transfers with new users of FAST-TERM can easily be corrected. The problem I have found many have run into is a misunderstanding of how to use the FCTN SHIFT X to get into XMODEM transfers. When the time is appropriate (see last months newsletter for specific details) the keypress of FCTN SHIFT X must be followed in the following sequence. First press and hold FCTN, then press and hold SHIFT and now press X while still holding down FCTN SHIFT and you will be in XMODEM transfers. Many people have been pressing all three at the same exact moment which

the computer is unable to respond to three inputs at the same time so it does nothing. Others have not held down the FCTN key while pressing SHIFT or have pressed SHIFT before the FCTN key. Note you still must follow another sequence of operations before you go into XMODEM transfer mode such as entering the program name into FAST-TERM (FCTN N) and telling the BBS you want to download etc... Last months newsletter gives exact steps on these operations.

BBS TIP:

While accessing a BBS you can either pause a display or abort a function using special commands. Each BBS does this differently. For TIBBS a CTRL S will pause the display so that you can then read it. FCTN Q will in return restart the display. To abort a function such as reading a text file pressing 6 will bring you back to the previous menu. On Risky Business the pause and restart keys are the same as on TIBBS (CTRL S and CTRL Q) but the abort function is different as it is CTRL C. Some BBS's use a P to pause the display and ENTER to restart. Fido BBS's use CTRL K to abort... Most BBS's use some combination of the above commands for control so just experiment with it to find out what the BBS uses

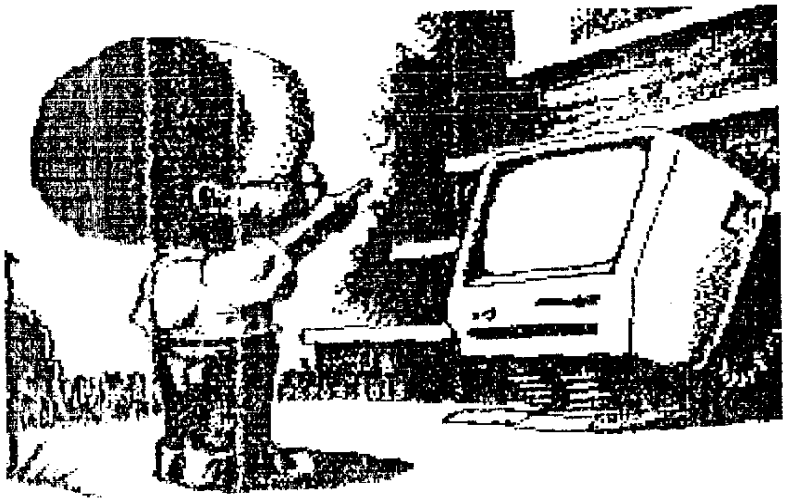
CALL WAITING:

If you have call waiting and someone calls you when you are connected to a BBS or a service such as Compuserve you will loose your connection. The call waiting tone that you normally hear when you are talking on the phone will for a split second interrupt connection between your modem and the services modem. If this happens in many cases the modem on the other end will think you have hung up and it will hang up loosing your connection. Sometimes when you are connected to a service such as Compuserve it will continue to be adding to your online time thinking you are still online because you improperly logged off. However, to prevent this your call waiting can be disabled by the following command. This will disable your call waiting for just ONE call. For touch tone phones dial and for rotary dial phones dial 1170 and you will receive another dial tone and you then dial the phone number. If you are using an auto dial modem just enter (or 1170) a comma and the phone number. For example the following command for auto dial modems will disable call waiting for that one call ATDT ,3575425 . If the line is busy you will again have to dial the disable code each time you dial the number as the disabled effect is only good for one call and then it is reset back to normal. You do not necessary have to use this just for when you use your modem but anytime you do not want to be interrupted by call waiting...

CABLE PROBLEMS?

A common problem when purchasing a modem from a non-TI retailer is that the cable your purchase may not have the pins correct. Usually the problem is with pins 2 and 3. Pin 2 on the RS232 should go to pin 3 on the modem and pin 3 on the RS232 should go to 2 on the modem and this is always the case and usually the source of the problem. Compare your RS232 manual to the manual with your mocem to see which pins your modem requires for proper operation. If your modem requires a terminal ready signal (it may

have a TR light on the moden if it requires a terminal ready signal) either force it on with the dip switches (for an Avatex 1200 switch down #6 and for the Avatex 1200HC switch down #1) or connect pin 6 on the RS232 to 20 on the modem and pin 20 on the RS232 to 6 on the modem and this should give a terminal ready signal...Gary Cox



PROTECTION

There are strong FEDERAL LAWS against duplicating copyrighted programs. Please do not break these laws!

DISCLAIMER

The Mid-South 99 Users Group is not affiliated, sponsored by nor has any relationship with any company implied or otherwise. Any mention of a company or product is not an endorsement of either the company or the product.

NEWSLETTER INFO

Visitors and potential members may receive 3 free issues of TiDbits while they decide if they wish to join (no obligation). A Dollar sign ($) indicate that your dues are due. Please pay your dues to be able to continue to receive the newsletter and other benefits of the group. You will note a letter and date on the top of your address lable. The letter Y indicates if you are a member and the date indicates the last time you paid your dues. One year from the date your dues are due!

CALENDAR

MEETINGS: March 19, April 16th May 21st (3rd Thursday!)
WORKSHOPS: March 28, April 25th May 23rd (4th Saturday!)

OFFICERS

Mac Swope	President	901-363-3880
Michael Dorman	Vice-President	501-732-5126
Gary Cox	Secretary	901-358-0667
Gerald Smith	Treasurer	901-363-6273
Pierre LaMontagne	TIBBS Sysop	901-357-8110
David Ferguson	Co-Librarian	901-795-3287
Rick Glisson	Co-Librarian	901-386-1159
Adolph Butler	Chairman Cassette Library	901-948-0732
Al Doss	Editor - Newsletter	901-743-6781
Gary Cox	Chairman - Program	901-358-0667
Gary Cox	Information and Assistance	901-358-0667
Mac Swope	Chairman - Equipment	901-363-3880

24HR TI BULLETIN BOARD

TIBBS (Mid-South User Group)	300/1200 bd	901-357-5425
Risky Business (Beery Miller)	300/1200 bd	901-726-5623

GROUP MAILING ADDRESS

Mid-South 99 Users Group
P.O. Box 38522
Germantown, Tn. 38183-0522