THE GUILFORD 99'ER NEWSLETTER

Carl Foster, President   Robert Dobo, Program Library
Joseph Martin, Vice President   Bob Carmany, Program Chairman
Mike Garrett, Secretary/Treasurer   Sandy Carmany, Education

+-+

+-+

OUR NEXT MEETING

DATE: August 5, 1986.   TIME: 7:00 PM PLACE: Glenwood Recreation Center
2010 S. Chapman Street.

A number of new or enhanced and exciting software packages for the 4A have
come out only recently. We will use this meeting to introduce the newest
version of FUNLWRITER (V3.3), the latest release of PRBASE, the database
program for the 4A, and V2.0 of Clint Pulley's "c" compiler. Be sure to
bring diskettes to the meeting and remember, these are "Fairware"
programs, we will ask for contributions for all copies and pass these on
to the authors. TI software may be a "steal" compared to what you would
have to pay for comparable software for other computer brands but don't
let us use the term literally.

TI SHOPPER
by Bob Carmany

The eagerly awaited last and definitive version of FUNLWRITER has arrived!! When I returned from vacation, there was a package from Tony McGovern waiting for me with Version 3.3 inside. This latest edition has some very welcome improvements and modifications. You can now re-enter FUNLWRITER from the DM 1000 (Version 3.1) and the package will load c-99 compiler with re-entry as well. The documentation has been completed and is thorough enough to let even the novice interface FUNLWRITER with his favorite program. The user menu has been expanded to allow the entry of five custom programs and there are other interesting improvements!! This is a truly outstanding piece of software, to say the least.

Asgard Software has a program called PRE-SCAN IT available. This program is supposed to make you XB programs run faster without the wait while the program initializes. It is $10.00 from the manufacturer. While we are on the subject of Asgard Software, several of the character fonts (4 to be exact) have the "Z" reversed in the GRAPHX COMPANION package. Fortunately, they are not in any of the interesting fonts!

UNISOURCE (one of the TI mailorder sources) has gone out of the TI market. They have liquidated their stock and will no longer be selling TI-related products. That leaves TRITON, TENEX, and TEX-COMP as the major mailorder suppliers of TI products.

Well, MYARC finally introduced their "new" computer! GENEVE is a "computer on a card" that uses an IBM-XT keyboard plugged directly into the PEB. It appears to have some serious problems --- not the least of which is the $495 price tag. It

has compatibility problems with other MYARC products -- the 128K and 512K cards have to be modified and there are software compatibility problems as well. In addition, there is no cartridge slot available although MYARC claims to include a cartridge copying program. Oh well, we will have to wait and see if it ever "makes the grade".

THE SMART PROGRAMMER is finally out! It appears to be carrying on the tradition that was started by Craig Miller. It is even the same gray color as the original!! The "old" staff is involved so it should be well worth the $15.00 subscription price.

Well, that about does it for this month's column. Until next month . . .

THE BRAIN
by Howie Rosenberg

There has been somewhat of a lack of programs for the TI-99/4A in the category of scientific. The introduction of BRAIN by Datax in no way helps to fill that gap. A demonstration version of this program has been distributed and .this review is based on that version. The demonstration was somewhat altered so that math routines would give correct answers. The author had introduced a random number into the routines as a feeble method of making the program useless. There was no need to do so the program is already of very little value. BRAIN is a menu driven collection of math and science routines. The items on the main menu are Annuities, Math and Calculus, Electrodynamics, Trajectories, Conversions, Geometry, Physics, Vectors, Tables, and help. Selection of many of the items such as Physics, Electrodynamics, and trajectories yield nothing but a message that this item is unavailable in the demonstration. No matter there is enough in the demo for the casual user to evaluate. The math and calculus option reveals a rather sparse submenu consisting of Length of arc, Exponents, factorials and logs. Every one of these functions is a single line equation which anyone who really needs can type in in a single line of code with considerably greater speed than the rather slow boot required. I might mention that the slow boot is partially do to a rather lengthy set of call load statements loading assy routines, some paranoid protection(who would really want to pirate this junk) and the piece de resistance the new lower case character set! I have seen a variety of character sets! TI's fix for TI writer followed by the one which many of us use now which originally was part of FAST TERM. Each of these sets is characterized by lower case descenders. We now have the first character set which has lower case ascenders! It is the ugliest character set I have seen. It is downright disturbing.

The menu selection tables yields a submenu containing trig tables, TMS 9900 instruction set, color codes, metric conversions, ASCII codes, and a table of chemical elements. None of the above really warrants any expenditure by the average user. Those who do need them might find them of value if they were in a rapid, easily accessible format which they are not. Take the element and isotope tables. Upon selection six screens are presented in succession with a key press to access the next screen. It would make more sense to have access to the required data upon entry of a keyword(a data base!). A program which does no more than provide menu driven solutions to very simple equations and tables is certainly not worth the $49.95 price tag that DATAX places on this junk. If it were not for their advertisements in Micropendium and the expense they have gone to to distribute this demonstration, I would consider it as a juke. It has become a custom, to include a report card in software reviews. i have been opposed to this practice as grading is subjective unless one can establish a benchmark with which to compare the program. I make an exception in this case because I think that a record may have been established for a commercial product in the TI marketplace. Report Card

Utility F

Ease of use B

Documentation D(inadequate help files)

Value F

Final Grade D-

Conclusion: Of no value at the present price. For say $10 may prove minimumly useful for a very small segment of the user population.

DISK DRIVES
by Ken Hamai

Well now, let's see here...if I dig around this closet for awhile, I think we got something on...Yep, here it is, disk drive bits and bytes.

First, a little bit/byte of history...I vaguely recall back in the dark ages of pre-minicomputering in the the late 1960's, the earliest small computers, if we can call them that, only spoke Fortran and used punched paper tape for storage of information. The holes in the tape stood for logic 1 and the absence of a hole for logic 0. I remember when I was in college, that paper tape and all the neat little holes made swell confetti that was a hell of a mess to clean up if you had your room "papered" by gremlins with the stuff. Them little yellow dots would stick to everything in the dormitory rooms during the winter because of the static electricity. Needless to say, the use of paper tape was bulky, noisy, because the machinery had to punch all those holes in the tape, slow, not easily corrected, and messy, especially if you were the dorm nurd.

Then came magnetic tape. This was a major improvement. It was convenient and cheap, could hold large amounts of data and was faster than the paper tape. The one major problem with this is well known to all frequent users of the TI 99/4 cassette tape OLD CS1 loading routine, the tape operated in sequential fashion. If the user was at the end of a tape and the program he wanted was at the beginning, he had to wait a frustratingly long time for the tape to rewind. This is almost as bad as waiting for Multiplan to Recalc!

The early '70's saw the invention and use of the floppy disk drives. The idea was simple: instead of tape, use a rotating disk and instead of waiting for the tape to rewind, move the read/write head directly over the location of the desired program. This is just as one would move the tone arm on a phonograph to select a certain tune from a LP that contained several other tunes.

It wasn't until 1973 when the Big Blue (IBM) became the first company to announce the use of a flexible disk with read/write capability into a system. The diskettes were 8 inches in diameter and the drives were about as big as the 99-4/A console. The latest thing in 1979 was to have a $50,000, 64K memory, word processing computer in your office typing pool with two of these drives built in. Currently these beasts are becoming increasingly rarer but you can still see a few of them around.

The 5 1/4 inch disk drive as we know them were introduced by Shugart Associates in 1976. This name should be familiar to all 99er's with the TI PHP 1250 drives for the Peripheral Expansion Box (hereinafter referred to as PEB). TI used the Shugart Model 400L drives for this peripheral...Nothing like the best!

Okay (for you Forth programmers...OK...for Ramon...AY), now that we got the closet cleaned out, let's cover a few things on the Shugart 400L drive. If you have one of these you know this is a single sided drive, capable of storing data on 40 tracks, up to a maximum of 90K with TI equipment. This is a very well built and reliable unit, like the PEB, almost bulletproof, and should last for a long time under normal home computer usage. It is one of the quietest operating drives manufactured due to the design which uses the stepper motor and a snail cam to move the read/write head over the disk. The drive is also equipped with a solenoid operated pressure pad which presses the disk unto the head only during read/write operations. Only a few manufacturer's provided drives with this extra. This is a nice feature since it keeps the head and diskettes from unnecesary wear. The TI disk system spins all drives connected to the controller even though it is only accessing one of the drives. If it wasn't for this solenoid, any disks in the other drives would also be rubbing against the heads unnecessarily. The distinct click of the solenoid can be heard whenever the drive light switches on or off during disk operations.

About maintenance...Keep in mind that it is usually the mechanical parts in a computer system that break down or wear out first. Disk drives are many times the first things that go on the fritz. For the average home computer user, maintenance should be limited to periodic cleaning of the drive head with one of the commercial head cleaning kits and maybe annual removal from the PEB or other enclosure to clean out any accumulated dust.

One must avoid the temptation to oil the drive, especially the rails on which the read/write head slides. The drive is designed to operate without oil on the rails. Oiling can actually gum up the drive. This is due to the fact oxidation will cause a thin layer of oil to become gummy. The effect is even worse when the oil is heated or warmed. To this gummy substance, one only needs to add dust which is normally found in most environments, and what one now has is one helluva mess inside the disk drive. This could keep the drive from operating properly and would require a major teardown to clean up. In

the PEB the situation is even more aggravated because the airflow generated by that humongous loud fan that is keeping the cards nice and cool is sucking in air through the equipment, including any airborne dust and dirt. Sort of reminds me of those TV commercials during the "Cold and Flu season" where they show all those cartoon germs and buggers flying around your head...

Equipment that is transported often or used heavily can be subject to troubles which show up as disk errors during read/write operations. One of the culprits is disk drive rotating speed as covered in Ron Rutledge's article which was reprinted in the April issue of the ROM. The other more likely problem of disk read/write errors is head alignment. On the 400L Shugart, alignment is easily accomplished in most cases by loosening the locking screws on the stepper motor and slightly adjusting it back and forth until it reads the disk that you want it aligned to.

For instance, let's say all your other 99er buddies have disk drives that can read a particular disk. Yours can't read the disk at all. The chances are good that your drive is no longer in alignment. What you need to do is to use that disk to line up your drive. If you really want to get picky, the ultimate test is to get one of those industry standard diskettes and calibrate your drive to that. I find that is not really necessary in most cases and especially for single density operation. I have had satisfactory results on 40 track drives just by using a TI original disk version of a program to align the head. If you can get your drive to read original unmodified TI disks, the chances are good that your alignment is okay...(OK).

I have to caution you that this is not something to do if you are not mechanically inclined or when your wife's calling you to dinner. There could be a considerable amount of adjustments involved, depending on how far out of whack the drive is. One time I bought a used double sided drive which took about an hour to align because both heads were out of adjustment. I learned how to do it by tinkering, but then again, I also repair antique clocks as a hobby. You might say the same caution should go for Rutledge's hints. If you do not fully understand what he is recommending, then...Don't do it! Above all, a good rule is: Don't fix what ain't broke!

Rutledge's article also covered the special resistor pack required to connect up a second disk drive to the PEB very well, but he failed to mention that the resistor is only needed if you are using the 400L as Drive 1. If you have another brand or model drive, and are using it as Drive 1, then the special TI resistor is not needed. In any event, the required resistor is readily available at any electronics store (you all know my favorite). Also, he mentioned a required adapter board. This special board is only required if you use the original TI hookup. This hookup included special keyed connectors that allowed connecting the cables in a certain configuration to control the drives without doing any internal jumpering at the disk drive strapping pack. A cable and jumpers could be easily made up to run any number of add on drives up to the maximum of 3 for TI controllers without the adapter board. All you need is some 34 position connectors, some cable, a vise and a little bit of know how.

FILEMOVER

Here is a short program from Jim Peterson (Tips from Tigercub) that is for all of you with just one disk drive. This program will transfer any number of D/V 80 files from one disk to another in one pass. The program will optionally allow you to rename the files as well. The only restriction is that the total of the files must not exceed 42 sectors. At any rate, here is the program:

```
100 DIM M$(2000),F$(25),C$(25):: CALL CLEAR :: T$=CHR$(1)
110 DISPLAY AT(8,6):"TIGERCUB FILEMOVER" :: DISPLAY AT(15,1):"PRESS ENTER WHEN 120 F=F+1 :: IF F)25 THEN 130 :: DISPLAY
AT(12,1):"FILENAME? DSK"&I$ :: ACCEPT
    AT(12,14)SIZE(-12)BEEP:F$(F):: IF F$(F)<>T$ THEN 120
130 F=F-1 :: FOR J=1 TO F :: ON ERROR 260 :: OPEN #1:"DSK"&F$(J),INPUT ::
DISPLAY AT(12,1):"READING "&SEG$(F$(J),3,255)
140 X=X+1 :: LINPUT #1:M$(X) :: C=C+LEN(M$(X))
150 IF C)10000 THEN DISPLAY AT(20,1):"INSUFFICIENT MEMORY FOR "&SEG$(F$(J),3,
255) :: GOTO 190
160 IF EOF(1)<>1 THEN 140
170 X=X+1 :: M$(X)=T$ :: CLOSE #1
180 W=W+1 :: NEXT J
190 X=0 :: DISPLAY AT(15,1):""::  DISPLAY AT(12,1):"INSERT COPY DISK AND PRESS":
"ENTER"
200 CALL KEY(0,K,ST):: IF ST=0 THEN 200 :: DISPLAY AT(13,1):""
210 FOR J=1 TO W :: IF F$(J)=CHR$(2) THEN 230
```

```
220 DISPLAY AT(12,1):"FILENAME? DSK"&F$(J):: ACCEPT AT(12,14)SIZE(-12)BEEP:
C$(J)
230 NEXT J :: FOR J=1 TO W :: IF F$(J)=CHR$(2) THEN 250 :: OPEN #1:"DSK"&C$(J),
OUTPUT :: DISPLAY AT(12,1):"SAVING "&SEG$(C$(J),3,255)
240 Y=Y+1 :: IF M$(Y)<>T$ THEN PRINT #1:M$(Y):: GOTO 240 ELSE CLOSE #1
250 NEXT J :: END
260 ON ERROR STOP :: DISPLAY AT(22,1):"CANNOT OPEN "&SEG$(F$(J),3,255)::
F$(J)=CHR$(2):: RETURN 180
```

FORTI
by Randy Ainsworth

Ever since I first heard about a P-Box card that would let you have 12 voices for music, I've wanted to get my hands on one. As you know, the 4/A console is limited to 3 notes and one noise. According to Ron Albright in the Orphan Chronicles, the FORTI card was developed by TI employed engineers, but TI elected not to buy the rights to it and market it. (The card was developed as part of a TI incentive program for employees to create new products for the 4/A; if the company liked the ideas, they would purchase the rights to them for later sale.)

(I understand that at the big CES show in Las Vegas in 1983, the FORTI card was playing an impressive rendition of "Chariots of Fire" in the TI booth.) After Black Friday, the rights to the FORTI system were purchased by Don Bynum, former Chief engineer of the Consumer Products Group. Anyway, the FORTI card has 4 sound chips on it and is driven with FORTH-like software which is supplied when you buy it.

The card is not housed in the usual metal case; instead the case is made of a special plastic similar to what is used for photographic underwater cases....very durable. Curiously, the TI logo is embossed on the outside of the case. Documentation is quite thorough, coming with two demo disks and about 36 pages bound in a 3-ring binder.

To begin, open up your P-Box and put the card anywhere your heart desires. There are RCA jacks on the back for connection to a stereo (even quad if you were suckered in on that a few years ago). I would suggest that you use the stereo connections as it makes the sound considerably better (depending upon the quality of your stereo) and the sound coming through your monitor's speaker doesn't sound right anyway.

To run the software, use E/A #3 and load the program just like you would load FORTH. When the menu screen is displayed, you will see a number of FORTH-like commands. To hear the sample tunes, just type in ALBUM and it will play the selections that have been pre-created. Some sound quite nice, others are a bit weak, but all tend to show off some of the different capabilities of the card. While the music is playing, several colored bars are displayed in an interesting manner (depending upon the music being played) which are a graphic representation of what the sound chips are doing.

Creating your own music is not too difficult, although I would recommend some musical knowledge. Even if you know nothing about music, you should still be able to convert sheet music if you study the docs a bit.

Actually, the FORTI docs start with a bit of an introduction to time signatures, key signatures, octaves, and all that other musical stuff. (At this point, I should probably note that I play guitar and a bit of synthesizer, so I do have some musical knowledge.)

First, you have to create a voiceline with the program's editor. In reality, this is the 64 column editor of FORTH, so you will need to have a monitor with good enough resolution to handle the small letters (I did not find it to be a problem unless you're using a TV).

A voiceline is just FORTI's way of noting what you want one of the chips to play (they call each voice a "musician"). If you are playing a melody, this would be one voiceline. While creating the voiceline, the user can program in a slur or even change the envelope (attack and decay characteristics). This is all done quite simply in the editor.

Once you have saved the voiceline to disk, you can go back and play it so you're sure it sounds the way you want it. It's a bit tricky to get all the notes worked out in the proper time signature, but it just takes a little persistence. Next, you start to create the conductor. This is what determines how everything comes together.

You can assign a voiceline to several voices to fatten the sound if you want. Three of the "musicians" can act as drums

or extra low bass parts to fill up your sound also. The documentation has information on how to get fancy with your music such as grace notes, accents, and some of the other things that make music interesting.

In all, it's a fairly easy system to use. It can be a little complicated if you're not transposing something out of a book or from a printed page, but it's still not too hard to get the hang of it. One complaint I have is that you cannot do chorusing. (Chorusing is where you add a slightly delayed, slightly out-of-tune voice to fatten up the sound and give it a liquid sound) I suppose you can create a similar effect using the envelopes though.

Once you have created several tunes you can set up the software so that it will play them all by just typing in ALBUM. One thing I really like is that since the computer's sound is going through your stereo, that means that you favorite games will also go through the stereo. My favorite is BLASTO....those explosions are really neat sounding.

The presence of the card does seem to mess up some of the sounds when running other programs. Whenever I use P-TERM, I usually get a buzzing sound when I set the program up for 1200 baud. Other XB and assembly programs sometimes react a little funny with sounds. Also, if you play your FORTI composition through the speaker in your monitor, some of the notes come out like white noise patterns instead of tones, using the stereo eliminates this problem.

In all, it seems more like a "gee- whiz" piece of hardware rather than anything really useful, but if you're into music it does sound nice through a good stereo. The price ranges from $150 to $200 depending upon the dealer. My feeling is that the card should be priced lower, say around $100. I am told, however, that costs do not allow such a low price at this time.

For further information, contact: Texas Peripherals Rt. 1 Box 150 Lancaster, Texas 75146 There is a demo tape available (or at least there used to be), but you really need to hear one through the stereo to appreciate the sound.

I should point out that the sounds produced by the FORTI card are the same tones you get in your regular programs. That is, you can't get violins or horns out of it, you are limited to whatever our sound chips can produce.

FORTH FORUM
by Bob Carmany

Following up on the idea of having a second character set available in Forth, here is an approach that was suggested by Tim MacEachern of Wycove Systems. This approach re-defines the EMIT word and recodes the alternative character set to have it available along with all of the standard print operators. It translates all of the printable characters from 64 to 127 into an italic set stored in 128 to 195. Incidentally , since the characters available go to 255, you could even add a third set!!
Here is the code:

```
0 64 VARIABLE CHAR-OFFSET
1 :(AEMIT) ( char --)
2 DUP 63 > OVER 128 < AND
3 IF ( translate to possible alternative char set )
4 64 - CHAR-OFFSET @ +
5 ENDIF
6 (EMIT) ;
7 : CHAR-SETS ( -- )
8 64 CHAR-OFFSET !
9 ' (AEMIT) CFA 'EMIT ! ;
10 : NORMAL ( -- : Use normal EMIT code )
11 ' (EMIT) CFA 'EMIT ! ;
12 : ITALIC ( -- : print in italics )
13 CHAR-SETS 128 CHAR-OFFSET ! ;
14
15
16
17
10
19
20
21
```

That is the Forth Forum screen for this month. If you are interested in some more information or Forth applications, there are several sources that you can use. MICROpendium has had a series of articles by Howard Arnold regarding the creation of customized fonts in TI-Forth. Mr Arnold, incidently, is from our sister group in Winston-Salem. There is an ongoing column in THE SMART PROGRAMMER by Mariusz Stanczak dealing with both Wycove and TI-Forth and DR. DOBB'S JOURNAL has a once-a-year Forth spectacular issue. And, of course, there is the national Forth Interest Group. So long 'til next month!!

EXPLORING c99
by Ron Albright

I have been exploring c99 for the TI of late. Written by Clint Pulley (38 Townsend Avenue, Burlington, Ontario, Canada L7T 1Y6) and available as Fairware, the language is a full-featured version of "small c". I have found few limitations with the language (lack of floating-point and math routines are the major ones), and have been able to do some nice routines with the language. Briefly, C is a very popular programming language through which, it has been estimated, 70% of commercial software for other machines is written. So what makes it different? It is a "compiled" language. That means, once you have written your program in c99, you run a companion program called a compiler. The compiler takes your C source code and generates assembly source code. The resultant code can then be run through the TI Assembler to generate object code, which executes just as fast as if you went through the strenuous (to me, anyway) task of writing assembly source code to start with. C is much easier to learn that Assembly language and is efficiently compiled with the c99 compiler. I have seen some programs written with c99 alone (there are a few on Compuserve; a simple text editor and a word-counter for TI Writer files by Warren Agee, a program similar to the TI Writer formatter, and a graphics demo by yours truly) and they are indistinguishable from pure assembly language, because the end-product is just that. If there is any interest, I will address the language more in depth in some more starter-level tutorials. I am no expert,by any stretch of the imagination, but I am learning and plan to spend a great deal of time with the language. Warren Agee, of Livonia, Michigan, has uploaded several tutorials to Compuserve which are mid-level in their scope and excellent in their content and expertise and these will help you as you get further into the language. It is a marvelous programming tool and, hopefully, this simple file will help you get started. Learning a new language is never easy, but it is time we all advanced beyond BASIC and started working in another environment. c99 provides a reasonable alternative. I could never think in reverse, so I gave up on Forth; I am too dense to learn assembly language. Pilot is too slow and requires too many disk accesses. Besides C is used in so many other machines and for so many other applications, it has to be good. Let's begin by seeing what we have to work with.

First, equipment-wise, you need the following: console, monitor, 32k memory expansion, at least one disk drive and controller, the Editor/Assembler package (cartridge or disk version) and, of course, the c99 system disk. A printer is nice (see below) but is certainly not imperative for programming purposes. Ideally, you would have two drives as this makes the work much easier, as does having at least double-sided drives (but ain't that always the case!). If you have double-sided drives, you can save yourself a lot of disk-swapping by, first, of course, making a backup of the c99 system disk and, secondly, copying from the Editor/Assembler disk, the files ASSM1, ASSM2 (the files for assembling source code) and EDIT1 (for the E/A Editor) on to the c99 system disk. But, if you have a single-drive or single-sided system, don't despair...things will work just fine with what you have.

Once you have gathered your tools, you should get a disk directory printout of the c99 system disk. Pulley even provides a disk catalog program on the system disk (called "SD" and running out of E/A 5 on my disk) but it doesn't print to printer). You will notice that there are a long of files in all shapes and "colors" (D/V 80, D/F 80, and PROGRAM files) and we will first go over what is important and what is not. Some of the files you will be using a lot, others seldom if at all, at least to start. Here are some of the files you should have and what they are for. I will list them in order of importance and probably frequency of use.

C99C,C99D,C99E

These are the compiler files. They are the heart and soul of the c99 system. There are PROGRAM image files and are run from Editor/ Assembler option 5. Unlike some PROGRAM image files, these CANNOT be run from option 3 of the TI Writer module (but they can be loaded with FUNLWRITER V3.3 as a menu option) The first thing I did with these files is rename them to be UTIL1, UTIL2, UTIL3. Then, when you chose the LOAD and RUN option from E/A (option 5), you only have to hit "Enter" and the files will be loaded by that name as a default without typing them in.

CSUP

This file is very important. It is a D/F 80 (which always means it runs from E/A option 3) which must be loaded immediately after you load you completed, assembled program. We will discuss this more later, but suffice it to say that your c99 program will never run if you don't load this file after it and with it.

C99MAN1,C99MAN2,C99MAN3

These are the D/V 80 files that contain the documentation Clint Pulley provides with the c99 system. They are not going to go very far in teaching you how to program in c99. Like the manual TI provided with the TI Forth system, they are simple a brief tutorial on how the different files work, and what they do, what the error messages mean, ect. They are quite adequate for their intended purposes. Pulley tells you up front "This manual assumes a knowledge of standard C or the availability of a suitable reference." That translates into "If you have never programmed in C, go buy a book!" I will recommend a couple at the end of this piece. Far enough, Clint! If you have a printer, print these files out for future reference. If not, find a friend who does. You will need a hard-copy of these files.

C99SPECS

A terribly important D/V 80 file. This short file tells you what c99 supports and, more importantly, what it does not support, when compared to standard C. Why is this important? I have yet to find a book that addresses only "small c", the version of C (more limited than "big C") that c99 is modeled after. All the texts I am aware of cover the full C language. Small c and c99 do not have all the functions of C. When you look at program listings out of these texts, you will quickly become frustrated if you try to type them in verbatim as they are already. Many program statements in C will give you errors in c99. You have to study this file when typing in program listings out of books to avoid these errors. For example, C supports "floating-point" arithmetic; small c and c99 do not. There are other examples covered in this file; print it out. You will need it.

GRF1DOCS

This is the documentation for the graphics routines supported by the current version (2.0) of c99. Print it out.

There are several other files that are, for the most part, files to be included in your c99 source codes as you use certain functions. We will go into this in some depth later, but you will use an "#include dsk1.filename" in your source files to copy these files into your source codes. For example, if you used some graphics commands in your source file to draw some sprites or such, you would need to use "#include dsk1.grf1refs" in your source code as a line before you started using the graphics commands. Else, the compiler won't understand what they mean and give you a multitude of errors. If you use commands to access disk files, you would have to use "#include dsk1.stdio" (for "standard input and output") before you started opening and reading from disk files. Notice the use of lower case in these #include statements. The compiler can use lower case, unlike the E/A Assembler which only accepts upper-case. Just keep the list of the other files as they will be used as you start to type in programs

How does one enter programs with c99? You can do it two ways. You can use TI Writer, but always use "PF" to disk rather than "SF" and throw in the "C DSKx.filename" syntax to clean all the control characters out. Or, preferably, you can use the Editor of Editor/Assembler. We won't do a program this time, as you have enough to do for now.

What about recommended books? I strongly recommend "C PRIMER PLUS" by Waite, Prata and Martin (Sam's Publishing, 1984). It is 500 pages and costs about $22. It is the "Going Forth" (Brodie) for C. It is easy to read, starts at a beginner's level and is chock full of example programs. Some usable with out dialect of small c, some not (at least without some conversions). I went though two other books on C before I found this tome. It is the best I have seen. If you know C, the bible (but much too advanced for me) is "THE C PROGRAMMING LANGUAGE" by Kernighan and Ritchie (Prentice-Hall, 1978). I found a back issue of Byte magazine also useful. The August, 1983 issue is devoted to C and contains some very nice articles and tutorials. You can still get a copy of this from Byte.

There you have it. The first chapter in the " Beginner's Guide to c99". If you are interested in more, let me know. I am just beginning myself and we can stumble through this together. My first swim in the waters was just great. Join in and learn c99.

(Editor's Note: Ron wrote this article prior to the availability of Version 2.0 of c99 and Version 3.3 of FUNLWRITER.

The updated version of c99 now has a floating point routine library and has been upgraded substantially to be more than just a "small c" implementation. Clint Pulley and the McGoverns have worked together very closely so that FUNLWRITER now makes an ideal environment for work with c99. It is easy to configure a dedicated disk that allows loading with only Extended Basic, contains the EDIT files of FUNLWRITER to enter "c" source code, the "c" compiler files and all the "include" and object files needed to compile, assemble and run "c" programs. Since the E/A cartridge is no longer required, "c" is now available to a much larger TI community. Even though Ron's article is not quite up to date as far as the software requirements go to work with "c", all his other comments and observations are applicable to the latest version as well.)

GUILFORD 99'ER USERS' GROUP NEWSLETTER

P.O. BOX 21691   GREENSBORO N.C.   27420