THE GUILFORD 99'ER NEWSLETTER

VOL.4 NO.11 NOVEMBER 1987

Mack Jones, President — Robert Carmany, Vice President
Mike Garrett, Secretary/Treasurer — Robert Dobo, Program Library
BBS: (919)274-5760(OPUS) — ROS (919)-621-2623

The Guilford 99'er Users' Group Newsletter is free to dues paying members (One copy per family, please). Dues are $12.00 per family, per year. Send check to 3202 Canterbury Dr., Greensboro, NC 27408. The Software Library is for dues paying members only. (Herman Geschwind, Editor)

OUR NEXT MEETING

DATE: November 3, 1987. TIME: 7:30 PM PLACE: Glenwood Recreation Center 2010 S. Chapman Street.

This month our feature is going to be our annual swap fest. If you have software that you would like to share, please bring it with you. Otherwise bring plenty of empty (formatted) diskettes, we will have quite a selection to pick from.

PRES PEEKS

The last meeting went nicely. With the new resistor installed, the letters were a little plainer but I was kind of disappointed in the results. When I installed the new resistor in my own console, I could see quite an improvement. Of course I am using a Commodore monitor instead of a television set and it is a lot sharper than a television with a modulator in my opinion. The members were able to watch the operation anyhow and one or two had never seen the RF shields removed.

I would like to give a hearty welcome to Janice Snider from High Point who is our newest member. Janice is very interested in the TI, but says she likes to let someone else do the programming and let her do the using!! Welcome Janice and I hope you will find the Club to your liking.

Got a letter from Joey Martin. You all remember that Joey had to give up the meetings and his President job when he enrolled in a computer class at the Rockingham Community College. When he finishes, he will be able to program the computers that operate the various machines in business manufacturing. Joey is in the process of choosing a Clone which will be more beneficial in his schooling. Soooooo, Joey has the following for sale:

EXPANSION BOX W/ CONT. CARD, 2 DS/DD DRIVES, 32K, RS232
TI BLACK/SILVER CONSOLE $89.95
SPEECH SYNTHESIZER $29.95
TI WRITER W/MANUAL $29.95

MULTIPLAN CART. W/MANUAL $29.95
ED. ASSEM. CART W/MANUAL $29.95
EXTENDED BASIC W/MANUAL $29.95
DISK MANAGER II CART. $19.95
PROGRAM REFERENCE GUIDE $15.00
PERSONAL RECORD KEEPING $ 9.95
TERMINAL EMULATOR II CAR. $14.95

THE FOLLOWING GAME CARTRIDGES ARE $5.95 EACH

HOUSEHOLD BUDGET MANAGER-PARSEC-ADVENTURE CARTRIDGE-MOON MINE-CAR WARS-HUNT THE WAMPUS-ADDITION AND SUBTRACTION 1-NUMBER MAGIC-EARLY LEARNING FUN.

Also for sale are the following :

TI-99/4A GAME PROGRAMS (BOOK) $11.75
LEARNING 99/4A ASSEMBLY (BOOK) $17.00
BACK ISSUES OF HCM 99ER MAG.$25.00
TI JOYSTICKS (ONE SET) $ 9.95
JOYSTICK ADAPTER (TO ATARI) $ 4.95
TEACH YOURSELF BASIC (ON TAPE) $ 6.95
TEACH YOURSELF EXTENDED BASIC $ 6.95.

Joey paid a total of $1251.15 for these items and now for the good news...THE WHOLE SHOOTING MATCH FOR ONLY $500.00!!! The kid must have had too many book sessions. I've seen less for $1000.00 or more! If you need to get in touch with Joey, either call or write to the following :

Joseph S. Martin
Rt. 13, Box 286
Reidsville, N.C. 27320
(919) 342-2621 AFTER 5:30 PM WEEKDAYS

I can assure you that these are in 1st class condition, because I watched him bite his fingernails one night when Herman plugged in his Ramdisk for fear that Herman would scratch his box!!

George Jordan still has the following for sale:

PE Box with SS/SD DRIVE, 32K, CONTROLLER CARD AND MANAGER FOR $295.00
RS232 CARD WITH SMITH CORONA TP1-DAISY W/CABLE $195.00
SPEECH SYNTHESIZER $25.00 ALSO TI-WRITER, MULTIPLAN, LOGO, AND EX-BASIC.

George also has other cartridges that he didn't list the names of. You can call George at (919) 299-4999. George is wanting to Byte an Apple!

I hope these members can sell their material, but I also hate to see them sell, for I know we will be losing two good members also. I can understand the reasons. Apple was pretty clever when they dumped all those computers on the Schools because I know of two friends who have bought Apples so their children could have them at home also.

We got a letter from West Germany that I would like to read to you as some of you will not attend the meeting. It's from Klaus F. Holtgen and goes like this:

Hello !
We are a german TI-Users Group with 42 members and we like to have contact with other User Groups in USA.
Some details about our group: Every second month we are publishing a newsletter on disk. Also we are developing Software in the following languages: Pascal, C, XB, Assembler, Pilot, German Logo, and much more...
We were (sic) happy if you would be interested in contact us, too.
Yours faithfully,
Klaus F. Holtgen

Ps. We are waiting for response...

I see no reason why we couldn't add one more Newsletter to our list, but of course it is up to you members. Think on it and let me know next meeting.

Herman has been gone on vacation for a couple weeks, and I just got back from a week up in West Virginia, so maybe we can still get this issue out in time. We have decided to have a good old "swap-fest" at our next meeting, so if you have any good stuff that you think the membership would like to have a go at, bring it along. Also bring a handfull of blank initialized disks for anything that you might want. The reason I ask that the blanks be formatted, it takes less time to copy when you don't have to wait for that to be done at the meeting.

I want to thank Mike Garrett and Carl Foster for the excellent job they are doing on printing the Newsletter. It is saving the Club quite a few bucks in printing charges. Also, for those of you who were not at the meeting, we can also save some bucks by us transfering our bank account into my Credit Union. We will have no service charge compared to the $7.00+ charge we are now being faced with and no check charge. We now have to pay $.25 per check. All we are waiting for is a Non Profit tax exempt number. The IRS promised to send me the forms 2 weeks ago, but as of today I haven't received them. I plan to call again tomorrow.

I had thought since my twin drives are capable of quad density, that I would get me a card to handle quad, but after talking to John Willforth, I don't think I will. John brought up something that I really hadn't thought of, namely, that if I had quad, I would not be compatable with the rest of the user's drives and I would have to have a 3rd drive just for compatability with them. So it's good to talk things over with someone in the know before you do anything drastic to your system!

We have some new prices for GREAT LAKES SOFTWARE in Howell, Michigan. The EXTENDED BUSINESS GRAPHS is now only $14.95 + $1.00 shipping. At the same price is BANNER, the program that enables you to print large banners. They also have CERTIFICATE, for those of you who need to print awards, diplomas, signs or advertisements. It sells for $19.95 + $1.00 for shipping. JOY PAINT 99 has had a price slash, as it is now selling for $24.95. That's a cut from the old price..$39.95. You will also get a free copy of CLIP-ART, a $9.95 value, when you order the JOY PAINT 99.

Megatronics has the Avatex 1200 Modem for only $85.00 with free shipping! I didn't know anyone shipped free, but they do! You also get free software for communications, and free Compuserve time with each order. You can order by phone by calling 1-800-232-6342. This Modem is Hayes compatible with external AC adapter, 8 LED status indicators, auto dial and answer, tone or pulse dial, telephone jack, and standard db25RS232-c connector. Sounds good!

For any of you who are interested in cassettes, Computer Shopper has an ad that states: CASSETTE LIQUIDATION for the TI99/4A. For details and sample, send a 22 cent stamp for postage to: 665, Box 542, Plainfield, NJ 07061. Bob Carmany has the latest version of Funnelweb from down under, but he says that there are a few bugs yet to be removed. However, even with the bugs, it is better than the last version says Bob. I am hoping that we will be able to get a de-bugged copy at the meeting. I have a few new programs from an out of town friend that I will be bringing and I think you will like them.

I will hang my close on this line, so until the next meeting, let the leaves fall, not your hopes! (Submitted by "Mac" Jones)

FORTH FORUM

Last month, we looked at a routine by Dr. Richard Terry of the HV99 U6 that allowed you to enhance the appearance of your applications by drawing a box around text messages or menu presentations. I promised that we would take the application a step farther this month and so we shall!

There are only a few restrictions on what you need to use this application --and there is even a "TEST" built in! First of all -GRAPH must be loaded to enable the application to work (see last month's newsletter). In addition, the two screens from last month must be typed in as screens #14 and #15 (or change the reference in screen #21 to the appropriate xx LOAD). Finally, there are a couple of auxiliary definitions that have to be worked into the screens somewhere (the blank lines could be removed and they will fit on one of them). See your TI-Forth manual for the editing instructions of how to delete a line with (FCTN-3). Here are the definitions that you will need:

```
: 2DROP DROP DROP ; : 2DUP OVER OVER ; : 3DUP OVER OVER OVER ;
```

Without any more delay, here are the rest of the screens for -WINDOWS.

```
SCREEN #17

0 ( WINDOWS -Blanking window 11FEB7 R.  Terry )
1
2 0 VARIABLE SAVED-DATA 1000 ALLOT
3
4 : CONTINUE BEGIN ?KEY 32 = UNTIL ;
5 : 1VDP-ADR ( Expect col,row--start vdpad )
6            40 * + ; ( Leaves start vdp of window )
7
8 : BLANK-WINDOW ( Expect col,row,width,height )
9            0 DO ( Blank out height lines loop )
10              3DUP ( Copies of col,row cnt-width )
11              SWAP 1 + SWAP   ( Increment each row )
12              32 HCHAR        ( Use col/row+1,width->blanks )
13           LOOP
14            DROP DROP DROP ; ( Leaves nothing on stack )
15

SCREEN #18

0 ( WINDOWS -Save window data 11FEB7 R.  Terry )
1
2 : SAVE-WINDOW ( Exp height/width/col/row )
3              1VDP-ADR ( Upper left start vdpaddr )
4              ROT ( Height to top for loop )
5              0 DO ( Read box by row hght times )
6                2DUP ( Dup width/1vdppos )
7                OVER I * ( Offset from start save addr)
8                SAVED-DATA + ( Start addr to read from )
9                SWAP ( 1vdp address to top stack )
10                I 40 * + ( vdp adr of start each row )
11               ROT ( Now adr/vadr/cnt for VMBW )
12               VMBW ( Write line of saved data )
13             LOOP
14                2DROP ( Drop copy of 1vdp/width )
15                  ; ( Leaves nothing on stack )

SCREEN #19

0 ( WINDOWS -Rewrite window 11FEB7 R.  Terry )
1
2 : WRITE-WINDOW ( Exp height/width/col/row )
3            1VDP-ADR ( Upper left start vdpaddr )
4            ROT ( height to top for loop )
5            0 DO ( Read box by row hght times)
6               2DUP ( Dup width/1vdppos )
7               OVER I * ( Offset from start save adr)
8               SAVED-DATA + ( Start addr to read from )
9               SWAP ( 1vdp address to top stack )
10               I 40 * + ( vdp adr of start each row )
11               ROT ( Now adr/vadr/cnt for VMBW )
12               VMBW ( Write line of saved data )
```

```
13              LOOP
14               2DROP ( Drop copy of lvdp/width )
15               ; ( Leaves nothing on stack )
```

SCREEN #20

```
0 ( WINDOWS -Sample program 11FEB7 R.  Terry )
1
2 : FILL-4 PAGE 959 0 DO 52 EMIT LOOP ;
3
4 : MS61 11 11 AT ." MESSAGE/OR"
5     11 13 AT ." PROGRAM " ;
6
7 : TEST FILL-4
8     15 20 5 5 SAVE-WINDOW
9      5 5 20 15 BLANK-WINDOW
10      5 5 24 19 DBOX
11      MS61
12      CONTINUE
13      15 20   5 5 WRITE-WINDOW
14      CONTINUE
15 ;
```

SCREEN #21

```
0 ( WINDOWS -Load screen 11FEB7 R.  Terry )
1
2 14 LOAD 15 LOAD 17 LOAD
3 18 LOAD 19 LOAD 20 LOAD
4
5
6
7
8
9
10
11
12
13
14
15
```

BASIC CORNER

III. SUBPROGRAM PARAMETER LISTS

In the last chapter we saw how subprograms fitted into the overall workings of Extended Basic. In this chapter we are going to go into the details of writing subprograms. Most of the fiddly detail here concerns the construction of the parameter lists attached to CALL and SUB statements, and some of the little traps you can fall into.

Any information can be transmitted from the CALLing program to the CALLed subprogram via the parameter list, and anything not transmitted this way remains private for each program, with the exception of the DATA pool which is equally accessible to all. If something is mentioned in the parameter list then it is a two-way channel unless special precautions, provided for in XB, are taken. In this case the CALLing program can inform the subprogram of the value of a variable, but not allow the

CALLed program to change the value of the variable as it exists in the CALLing program. Arrays however, numeric or string, can't be protected from the follies of subprograms once their existence has been made known to the subprogram through the parameter list.

Let's for starters take a very simple but useful example, where a program needs to invoke a delay at various points. Now some BASICs (and TI LOGO) have a built in function called WAIT. XB doesn't have this command so you have to program it. It can be done by a couple of CALL SOUNDs or with a FOR-NEXT loop. Let's use an empty loop to generate the delay, about 4 millisec. each time around the loop, and place the loop in a subprogram.

```
230 CALL DELAY(200)
670 CALL DELAY(200/D)
990 CALL DELAY(T)
3000 SUB DELAY(A):: FOR I=1 TO A :: NEXT I ::SUBEND
```

This is easier to follow when editing your program then using a GOSUB, and you would need to enter the subroutine in every subprogram since GOSUBbing or GOTOing out of a subprogram is verboten. Also it's less messy than writing the delay loop every time. The example shows several different CALLs to DELAY. The first supplies a number, and when DELAY is CALLed, the corresponding variable in the SUB list, A, is set to 200. This is a particular example of the kind of CALL from line 670 where the expression 200/D is first evaluated before being passed to DELAY to be assigned to A. Variable D might for instance represent the level of difficulty in a game. The CALL from line 990 invokes a numeric variable T, and A in the subprogram is set to the value of T in the CALLing program at the time when the CALL is executed.

Nothing untoward happens to T in this example, as the DELAY subprogram does nothing to change A. Now it may not matter in this instance if T did not retain its value after the subprogram CALL. Suppose instead the delay was to be called out in seconds. Then a subprogram on the same lines DELAYSEC might go

```
230 CALL DELAYSEC(2)
990 CALL DELAYSEC(T)
4000 SUB DELAYSEC(A):: A=A0
4010 FOR I= 1 TO A :: NEXT I :: SUBEND
```

Now after DELAYSEC has been executed with the CALL from 990, T will have value 250 times its value before the CALL. This won't be a bother if you don't use T again for its previous value. If the CALLing program specifies a numeric constant as in line 230, or a numeric expression, the change in A in the subprogram has no effect on the main program. Suppose you can't tolerate T being changed in line 990 (and this kind of thing can be a source of program bugs). You will find that XB allows for forcing T to be treated as though it were an expression, thus isolating T from alteration by the subprogram, if T is enclosed in brackets in the CALL (not SUB) list. Suppose DELAYSEC is also called from line

```
970 CALL DELAYSEC((T))
```

If this CALL in line 970 is followed by the CALL from line 990, T not having been altered in the meanwhile, the same delay will be obtained, but if the order of CALLs were reversed the second delay would be 250 times the first. In the language of XB this is known as "passing by value" as distinct from "passing by reference". This can only be done for single variables or particular array elements, which behave like simple variables in CALL lists. Whole arrays cannot be passed by value, but only by reference. Expressions and constants can only be passed by value, and its hard to see what else could be done with them. In the example as written, a different variable name was used in the SUB, but if you remember the little experiment in the last chapter you'll see that it wouldn't make any difference if T had been used in the SUB list instead of A.

Now let's complicate things a little by flashing up a message on the bottom line of the screen during the delay interval.

```
200 CALL MESSAGE(300," YOUR TURN NOW")
270 CALL MESSAGE(T,A$)
3000 SUB MESSAGE(A,A$):: DISPLAY AT(24,1):A$
3010 FOR I=1 TO A :: NEXT I :: DISPLAY AT(24,1):""
3020 SUBEND
```

The SUB parameter list now contains a numeric variable and a string variable in that order. Any CALL to this subprogram must supply a numeric value or numeric variable reference, and a string value or string variable reference, in precisely the same order as they occur in the SUB list. In the little program segment above, line 200 passes constants by value and line 270 passes variable references. There is no reason why one cannot be by value and one by reference if so desired.

This process can be extended to any number of entries in the parameter list, provided the corresponding entries in the SUB and CALL lists match up entry by entry, numeric for numeric, string for string. The XB manual does not say so explicitly, but it appears that there is no limit apart from the usual line length problems, on the number of entries in the list. This is the only apparent difference between the parameter list in XB subprograms and the argument lists for CALL LINK("xxxxxx", , ...) to machine code routines in XB, and Minimemory and E/A Basics.

One little freedom associated with built-in subprograms is not available with user defined subprograms. Some built-ins, such as CALL SPRITE permit a variable number of items in the CALLing list. Parameter lists in user defined subprograms must match exactly the list established by the SUB list or an error "INCORRECT ARGUMENT LIST in ..." will be issued. To compensate for this inflexibility user defined CALLs allow whole arrays, numeric or string, to be passed to a subprogram. Complete arrays may be passed by reference only. Individual array elements may be used as if they were simple variables and may be protected from alteration by bracketing in the CALL list. An array is indicated in the parameter list by the presence of brackets around the array index positions. Only the presence of each index need be indicated as in A(). MATCH(,,) indicates a three- dimensional array MATCH previously dimensioned as such, explicitly or implicitly. Don't leave spaces in the list. If the subprogram needs to know the dimensions of the array these must be passed separately (or as predetermined elements of the array). TI Basics are weaker than some others in that they do not permit implicit operations on an array as a whole, a very annoying deficiency.

Arrays may be DIMensioned within subprograms. This will introduce a new array name to the program, and an array or variable name from the SUB parameter list can't be used or an error message will result. In the following code the main program passes, among other things, an array SC to subprogram BOARD (perhaps a scoreboard writing routine in a game).

```
100 DIM SC(2,5) :: ....
450 CALL BOARD(P,A$(),SC(,))
4000 SUB BOARD(P,A$(),S(,)):: DIM AY(5):: ....
4080 SUBEND
5000 SUB REF(V,A(),B(,)):: ....  :: SUBEND
```

BOARD generates internally an array AY() which is passed to another subprogram REF (maybe this resolves ties) along with SC(,), which BOARD knows as S(,), and REF in its turn as B(,) -- the same name could have used in all places. There is however no way that the main program or any subprogram whose chain of CALLs doesn't come from BOARD can know about the array AY(). This would hold equally well for any variable or array, string or numeric, first defined within BOARD and whose value has not been communicated back to the CALLing program via some other variable mentioned in BOARD's parameter list.

By following this line of reasoning you can see that there is no way for a subprogram whose chain of CALLs does not come through BOARD to know about array AY(). The only way around this is for AY() to be DIMensioned in the main program (even if this is its only appearance there) and the message passed down all necessary CALL-SUB chains.

This idea of DIMensioning an array only within a subprogram is particularly useful if the array is to READ its values from DATA statements and to be used in the subprogram. This could be done again from any other subprogram needing the same data, without having to pass its name up and down CALL-SUB chains. Remember that DATA statements act as a common pool from which all subprograms can READ. If the array values are the results of computations then these values must be passed through the CALL parameter lists.

For completeness note that , although the XB manual has nothing to say about it, IMAGE statements for formatting PRINT output are accessible from any part of a program in the same way as DATA statements and not confined to the subprograms in which they occur as are DEF entries.

It is not necessary to have any parameters in the list at all. Subprograms used this way can be very helpful in breaking up a long program into more manageable hunks for ease of editing. We shall also see in later chapters that there can be other benefits as well.

One more XB statement for subprograms remains, the SUBEXIT. This is not strictly necessary as it is always possible to

write SUBEND on a separate line and to GOTO that line if a condition calling for an abrupt exit is satisfied. Like a lot of the little luxuries of life however, it is very nice to have and makes programs much easier to read and edit. It does not replace SUBEND which is a signal to the XB pre-scan to mark the end of a subprogram. SUBEXIT merely provides a gracious and obvious exit from a subprogram (awkward in some Pascals for instance). The next chapter will demonstrate typical examples of its use.

IV. USEFUL SUBPROGRAM EXAMPLES

In the previous chapter we used as an example a DELAY subprogram which could, with a little refinement, be used to substitute for the WAIT command available in some other languages. You can extend this idea to build up for yourself a library of handy-dandy subprograms which you can use in programs to provide your own extension of the collection of subprograms that XB offers.

For our first example let's take one of the more frustrating things that TI did in choosing the set of built-in subprograms. If you have Minimemory or E/A you know that the keyscan routine, KSCAN, returns keyboard and joystick information simultaneously, while XB forces you to make separate subprogram CALLs, KEY and JOYST, to dig it out. Since these GPL routines are slow it is difficult to write a fast paced game in XB that treats keyboard and joysticks on an equal footing as is done by many cartridge games. On the other hand in games where planning and not arcade reaction is of the essence there is no reason why the player(s) should be forced to make a once-and-for-all choice and not be able to use either at any stage of the game.

The subprogrammers approach to this problem, once it realised that it can be done (and we have commercial XB games where the writers haven't) is to write the game using joysticks, but replacing JOYST by a user defined sub-program JOY which returns the same values as JOYST even when keys are used.

The first step in telling whether keys or joysticks are being used is to check the keys, and if none have been pressed then to check the joysticks. If a key has been pressed then its return, K, has to be processed so that the direction pads embedded in the keyboard split-scan return the corresponding JOYST value. A subprogram along the lines of the one used in TEX-BOUNCE does just this.

```
900 SUB JOY(PL,X,Y):: CALL KEY(PL,K,ST):: IF ST=0
THEN CALL JOYST(PL,X,Y):: SUBEXIT
910 X=4*((K =4 OR K=2 OR K=15)-(K=6 OR K=3 OR K=14))
920 Y=4*((K=15 OR K=14 OR K=0)-(K=4 OR K=5 OR K=6))
930 SUBEND
```

PL is the player (left or right joystick or side of the split keyboard) number and is unaltered by the procedure. The simple-minded approach for converting K to (X,Y) values by using the XB logic operators (one of the more annoying omissions from console Basic) seems to work as well as any. The subprogram as written checks the keys first but balances this out by putting the processing load on the key return.

This is as good a time as any to sharpen your own skills by working out alternative versions of this procedure, and also by writing one for mocking up a substitute CALL KEY routine to return direction pad values even if a joystick is used.

(TO BE CONTINUED NEXT MONTH)

MODEM TALK

I am happy to report that both of our BBS's are operational again and that both feature TI file downloads. Here is a quick rundown of what is new:

CFS1ARC.TI, CFS2ARC.TI and CFS3ARC.TI: This is the complete Creative Filing System package including documentation. In the most recent issue of MicroPendium the reviewer of Data Base packages for the TI states: "CFS is the most feature packed

nd interesting application.....". This is a shareware program authored by Mark Beck. It certainly is worth a download.

DM37ARC.TI and DM38ARCTI. These are the most recent versions of DM1000, the disk and file manager for the TI. For all intents and purposes nothing much has changed compared to version 3.5, which has been around for a time. Changes in the most recent versions are minor bug fixes and efforts to make DM1000 compatible with Myarc and CorComp controllers and exotic hardware such as quad-density drives.

FW1ARC.TI and FW2BARC.TI. This is the most recent incarnation of Funnelweb, now in Version 4.0. This is not yet a final version and according to Bob Carmany there are still some bugs that need extermination. This version of Funnelweb has been greatly enhanced with many added convenience features. A CONFIG program can now be invoked to set up hardware and color choices, etc. instead of having to know how to edit the LOAD file, to name just one. The two Main Menu screens have been expanded to allow the direct loading of two E/A programs of your choice. The SD (Show Directory) function has been greatly enhanced with many added convenience functions. Unfortunately with the added capabilities, FW has also increased in size. Not much but just enough that the FW UTIL1 file will no longer fit into my 8K GRAM 7. This forced me to experiment a little bit and I found out that the V4.0 TI Writer files will work with the V3.4 of FW (Normally you are not supposed to mix and match FW component files). This still gives me FW in just one GRAM and all the nice new SD and TIW features. Neat....

SVPARC.TI This is a real strange bird of a program which originated in France and came to us via Australia. When you run LOAD a very high powered single bit graphics program is lurking in the background of XB. Try some of the demo files (MER, ETOILE, LISSAJOUR) and watch the 3D-like graphics coming up on your monitor! The file includes docs (in English).

PICARC.TI This is an upload of PICASO, another highpowered graphics program for the TI.

To access the downloads, on OPUS at the Main Prompt, key in F for files, followed by A, Area change, if necessary, followed by 15 for the TI file section. A password is no longer required to download TI files.

With the ROS board key in F for files followed by N for NewIn to get the latest listing and brief description. S for send will start the download. Also, key in C for Change file area followed by TI99 to see some of the older files. S for send, again will start the download. Remember to key in N when the system asks for 1K downloads. Unfortunately the TI presently will not support 1K or Ymodem downloads.

Several bright programmers are now working on a new ARChiver version which not only will combine any number of files into one convenient up/download but compress and uncompress all files. Once this is working the transmission time will be cut in half and we will also gain additional file space on our boards. At the moment I have only seen beta copies of this software and we are not quite ready yet to switch...so stay tuned!

One more item, both of our boards feature message bases FOR YOUR USE. If you have any questions, drop me a MESSAGE. I just plumb love to read mail and would dearly love to hear from you. So, please don't call me, drop me a message and you will have an answer within a very short period of time!