

NOTICES

MEETING

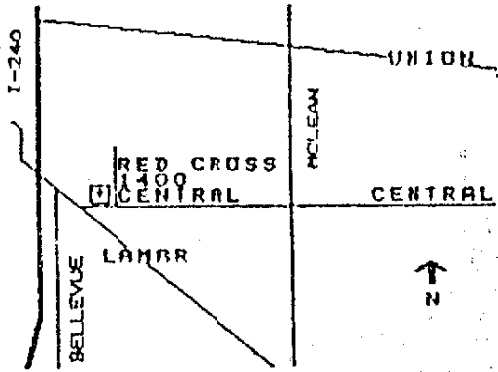
7:00 P.M.

Thursday, November 20th
Red Cross Building
1400 Central Av.

WORKSHOP

9:00 - 12:00

Saturday, November 29th
at Rick Glipson's
4570 Andrew Crossing
Memphis, Tn 38128
386-1159

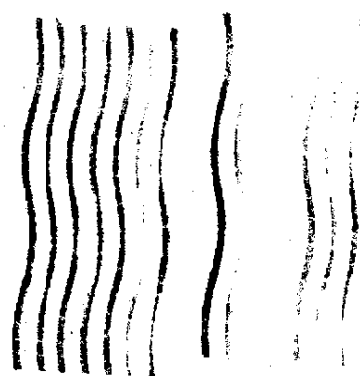
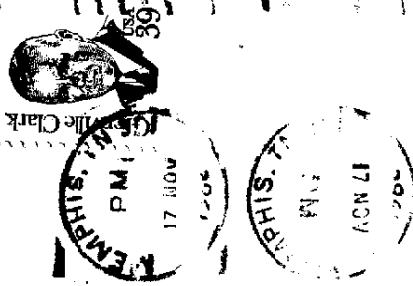


MEMBERSHIP APPLICATION

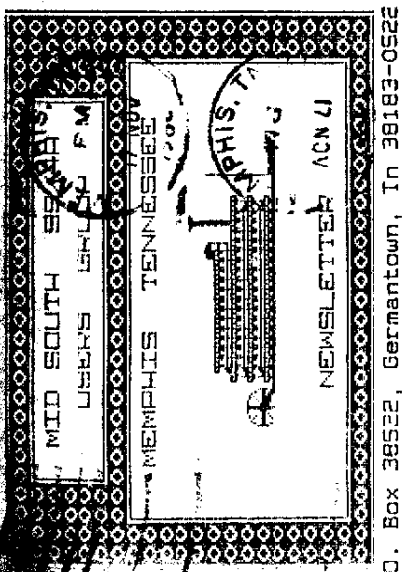
NAME _____ \$5.00 FAMILY
 ADDRESS _____ \$10.00 JUNIOR (under 15)
 CITY _____ ST _____ ZIP _____ \$10.00 ASSOCIATE (NML apply)
 PHONE () - : INTERESTS _____

EQUIPMENT, ETC. _____

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



UG 2/86
 DALLAS TI USER GROUP
 1221 MOSSWOOD
 IRVING, TX 75061



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

TIBITS

PRESIDENTS BIT

I usually start off this editorial about how hard I have been working and all of the reasons that I have missed another meeting or function that the President should have attended. Yep, I missed the Chicago TI-FAIRE, but not for the same reason. I missed the Faire because of a 10 day vacation. It started about 2:30 am one Wednesday when I was rudly awakened by this horrible pain in my left kidney. Yes, you guessed it, A KIDNEY STONE, and if you haven't had the pleasure of enjoying one of those delightful little darlings, you really don't know what you have missed. This is about my fourth encounter with one of the things so I really knew what was in store for me... I THOUGHT... The personnel at St. Francis Hospital were there to welcome me with open arms. I arrived very uncerimoniously at the Emergency room at 3:00 am and was whisked into a room about half the size of a small bathroom. I was told that a Doctor would be with me immediately. "Their immediately and my immediately ARE two different immediatelies". Shortly people started flying around me... They could tell I was in definite pain... Maybe it was all of my crying or the buckets of tears coming from my eyes. Folks, I was in a dilemma, "one minute I thought I was going to die and the next moment I was afraid that I wasn't going to die", and at the time I really didn't care who won. For the next 10 days I was awakened when I should have been asleep, put to sleep when I should have been awake, and generally turned inside out. I was finally released the Saturday of the Chicago TI-FAIRE and was allowed to go home to recover. After 10 days of total bed rest my major concern now is to regain a little of the strength that I previously had. As for the stone, it is still in me and is waiting to rise up and awake me on some other uneventful morning when I have nothing to do but start this cycle again. All in all, I am feeling better and hope to pass the stone without any additional surgical help. There were days when the pain was so intense and the pain medication so strong that I really have lost complete days of memory. I certainly hope no one has to go through the pain and suffering of a Kidney stone. G. Smith

MYARCTICLE #2

Well here I go again. This time I will be reviewing Myarcs 512K memory card. You asked for it Gary, you got it! As I stated in the first article, the card comes contained in a tan, plastic clamshell similar in looks to TIs. The only outside difference from TIs 32K card is that there is an extension on the back with what looks like a jack on it. More on that later.

Upon initial and subsequent powerups, the card acts like a standard 32K card. To utilize its greater memory for files,

ramdisk, and print spooling it must be initialized or "partitioned". This is done from command mode in Basic or Extended Basic. The command is CALL PART(X,Y,Z) where X is the amount of memory allocated to program memory, Y the amount of memory for the ramdisk and Z the memory to be used for print spooling. X must equal 32 and Y+Z must add up to 480. These are the only stipulations. To emulate a disk drive use the command CALL RMDK(X), X being a number from 1 to 5 signifying which drive you wish to emulate. To turn off the ramdisk, substitute 0 for X. To give the ramdisk a name use CALL VOL("disk-name"). This is especially useful when using a program that looks for files by diskname such as MULTIPLAN looking for "DSK.TIMP.filename". Print spooling may be used through either the parallel or serial port. Simply name it SP, SP/1, SP/2 OR SP/10 to correspond to RS232, RS232/1, RS232/2 or PIO respectively. To abort the print spooling enter CALL ABSP. The print spooling feature allows the TI99/4A to become a "multitasking" computer similar to larger computers in that the data is sent to the RAM allocated to the spooler then to the printer. This is done using CPU cycles not being used otherwise. In other words, you can start a document printing by sending it through the print spooler and then begin some other task "at the same time"! As long as the system is powered up, you could start some other program while the document finishes printing out. This is an outstanding feature which I have just begun to use since I was previously using a CorComp RS232 card which is not compatible with the print spooling function. A TI or Myarc RS232 card must be installed for this.

And now I'll discuss what you've probably been waiting for - The Ramdisk.

First of all, let me make sure that you understand what a "Ramdisk" is. A ramdisk is a certain portion of RAM memory that has been partitioned and set up to imitate or "emulate" a disk drive. At least as far as the storage of files is concerned. If you want to understand all of the technical details of just how it is done, you'll have to look elsewhere as that is quite beyond me. I'm not an electronics mechanic, just the driver! Anyway, for all practical purposes, the Ramdisk IS a disk drive. It just stores the files in RAM rather than on a floppy disk. If you stop and think of it, the main advantage of this should be obvious...SPEED! If you have ever seen a hard disk drive in operation, then you have a good idea of how fast a Ramdisk operates. Or think back to when you first upgraded your TI from cassette to disk operation. Like me, you were probably thrilled with loading in a program in 10 seconds instead of 2-1/2 minutes. Well now that time has been cut down to 1.5 seconds. Of course the time differential will increase the larger the program. Once you get used to using a Ramdisk, you'll want to store large files into it that you will be accessing a lot in one session at your computer. MULTIPLAN would be a good candidate for that. Here are a few comparisons of loading programs or files from a disk drive and the Ramdisk.

PGM/FILE SIZE.....	DISK DRIVE LOAD TIME..	RAM/DISK LOAD TIME
(sectors)	(seconds)	(seconds)
05	03	01
17	06	02
30	09	02.75
46	13	03.50
60	16	04.25

As you can see, the longer the file, the greater the distance between the load times for the two types of drives. This test was performed using the LF function in TI-Writer. Now that I have the latest version of Myarc's Extended Basic II, I keep the program files in my Ramdisk. So when I choose the XB II option from the menu, they load in and I am ready to go in about 1/4 of the time if I loaded them in from a floppy! Needless to say, I am quite happy with its performance, not having had any problems with my 512K card since I bought it 7 months ago. Of course you could get extra memory or another floppy drive or a print spooler for less money, but this product is really a bargain since you get all three. And I'm sure that others could think of even more uses for it than I have described here.

Now I'm excited about seeing the Gen'ève at the last Chicago Faire. NOW! And the best part of that is that with a slight modification, I'll be able to use my 512K card in the Gen'ève I hope to get soon. Just think, 1 MEGABYTE of RAM!!! And to think that I thought 32K of RAM was quite a deal just a couple of years ago. HERS...we sure have come a long way!...J. Leslie

CHICAGO TI FAIRE!

The Chicago II Faire held at Triton College in River Grove, Illinois November 1st was a fantastic success! I am writing this article about 50 miles south of Chicago on the way home in order to get a report into this month's newsletter.

The Faire this year was attended by an even bigger crowd (13 of us) from our group which included, Rick Glisson, Beery Miller, Dicko Vandenberg, Jonathan Leslie, Bob Jones, Richard Miller, Jeremy Glisson, Lonnie McClure, Michael Dorman, Pierre Lamontagne, Mac Swope and myself. The faire turned out to be a great opportunity to meet those high up and well known in the TI community. Just to name a few of the well known people at the faire was Ron Albright (Computer Shopper), Howie Rosenberg, Terrie Masters (Prez. LA 99'ers), Lou Phillips (Myarc), Chris Bobbitt (Asgard Software), Steve Lamberti (Texaments), Peter Hodie (program author), Paul Charlton (FAST-TERM author...), Jim Peterson (TigerCub Software), Ernest Chandler (Great Lakes Software) and many more...

More than 40 vendors had tables setup selling anything you could imagine for the II. Among the most well known vendors were Myarc, Asgard Software, Rytte Data, Databiotics, Horizon, Bytenaster, T.A.P.E., Tenex, Texaments, TigerCub Software, Trio Software, Great Lakes Software, Rave 99, Disk Only Software, Hunter Electronics, Competition Computer, BBS Computer Sales, Genial Traylor and many more. The Boston Computer Society and the Channel 99'ers of Canada even had tables setup with many from

other groups across the country in attendance. Even some friends of Beery in his old users group in Kentucky were there. I met many people myself that I met last year at the Faire and many new people this year.

Demonstrations by various companies and persons were carried out through out the day starting with Asgard Software who showed off several new software products one of which was an educational and entertaining space flight simulator called High Gravity (444.20), along with a program like "Print Shop" called Font Writer using instances etc., from TI-Artist. I bought Pre-Scanit and Bob Jones bought Font Writer (424) and the new GRAPHX companion III (479). I am not sure what the others bought. Font Writer uses existing TI-Artist and CS6D fonts or you can create your own with it's built in editor. Hopefully Bob will have us a review soon. Jonathan Leslie bought a program like Font Writer from someone else but I can not recall the name.

Clint Pulley (creator of C-99) gave a demonstration and technical talk of which Dicko was drooling over... A hardware technical corner was conducted by Al Stump and I think Jonathan obtained a schematic on how to install the Speech Synthesizer in the console. T.A.P.E. demonstrated their new products and Lou Phillips (Myarc) attracted the largest crowd with a demonstration of the new II compatible computer-in-a-card called a Gen'ève. Everyone seemed to be very impressed with it including myself as it has fantastic built in graphics capabilities and outstanding speed that is faster than even the IBM PC's! It likes a little more time before release date but several in our group are very seriously considering buying one. The Gen'ève uses a 9995 processor chip running at 12 MHz with a 40 or 80 column display (composite or RGB). It is capable of 2 Megabytes of addressable RAM, a mouse output port and it still uses the TI standard joystick. An IBM keyboard is included which has a built in numeric keypad and separated function keys. You no longer use your TI99/4A console but the new IBM style although you still use most of your other peripheral equipment. The system provides a means of saving all of your cartridges to disk so they are no longer needed (they are saved in the same format at the Gram Kracker). The Gen'ève has outstanding graphics capabilities with 7 graphics modes allowing 256 colors with every pixel on the screen having the possibility of being another color. Lou also mentioned the possibility of marketing the computer as a unit by itself and not just for the PEB. Several companies are already developing software for the computer such as Pecan Systems developing USCD fortran 80, USCD Cobal, USCD Pascal, USCD BASIC and many applications programs. Others include Inseobot, Databiotics, Paul Charlton (new FAST-TERM), Clint Pulley, CSI design, Fike-Creek... Lou also mentioned a Lotus 1,2,3 clone... Gen'ève will not be 100% IBM compatible but will run a similar MS-DOS type operating system and will have "datafile" read capability from datafiles created with IBM PC's and might run some IBM software that does not use assembly language... For more information contact a Myarc representative or ask Jonathan Leslie at the meeting as he is pretty well up on it's capabilities.

Looking elsewhere at the Faire Texaments was selling several new items one of which was "Rapid Copy". Rapid Copy will work on

Rytec and TI disk controllers and will copy a SS/SD disk in 30 seconds! Also Character Sets and Graphics Design III was being sold of which I purchased a copy.

Ryte Data had some interesting products including a compact peripheral system (a new type of Expansion Box), 1 MEG memory card, multifunction expansion cards, an easy to use BASIC compiler (translates BASIC into assembly language for much greater speed), 32K internal memory expansion, 80 column card, 128K SRAM card and much more. Ryte Data seems to really be supporting TI owners with all these new products. They also publish a great newsletter for only \$11 a year. For more information contact Ryte Data at, 210 Mountain Street, Haliburton, Ontario EOH 1S0, Canada.

All of us together spent over \$1300 dollars worth of software and hardware at the Faire averaging \$100 a piece. We had a fantastic time and hope to return again next year. A BIG thanks go to organizers of the Faire such as Sandy Bartels, Grant and others as they had to have spend a tremendous amount of time getting together such a large event. I estimate over 2500 attended the Faire this year!

We left the Faire a few hours early in order to take a quick tour of Chicago as none of us have ever been though Chicago before. With the help of a taxi driver on our CB we drove past Lake Michigan and just drove through town and saw the Sears tower etc... I will have to say that I will never complain about traffic in Memphis after seeing what Memphis could be like in Chicago!

Then after a short tour around town some of us took a tour of O'Hara air port. Bob being an air traffic controller was able to get us into the control tower at the air port to see the radar and the incoming planes etc... It was an incredible sight!

In closing I would like to thank all the guys who drove the cars and put up with each other for this trip. I would also like to welcome some new groups whom I met at the Faire to our newsletter exchange. Do not miss this months meeting as we will be demonstrating all of our new toys we purchased. We also have ordered a professionally taped video tape of the Faire for those who were not able to attend. We will be showing it at one of our meetings and it will then be available for you to check it out of the library. Speaking of library we have some new library programs and updates to ones already in our library. Reviews will be coming up soon in the newsletters of all we bought (and we bought a bunch!) and I will have more news of the Faire next month...Gary Cox

PROGRAM BIT

6:10 Doors Open

6:30 Library Open

7:00 First prize drawing held.

7:01-7:20 First prize drawing held and general discussion of Chicago TI Faire...

7:20-7:30 Second prize drawing held and nominations taken for December election of officers.

7:30-7:45 Demonstration of my FREEMARE programs WEATHER FORECASTER, Lawnmower and DATABASE.

7:45-8:30 Demonstration of new software and hardware purchased at the Chicago TI Faire.

8:30-8:40 Drawing for all other prizes.

8:40-9:00 TBA

9:00-9:20 New RAVE 99 keyboard replacement for the TI99/4A.

9:20-10:00 Help and assistance and general play period.

10:30 Doors close.

FREE DRAWING

In order to attract more members to the meetings and just for the fun of it a drawing will be held to give away 10 to 15 cartridges to members of the group who are in attendance at this months meeting. The cartridges include games and utilities. Rules for the drawing are that you must be a member of any TI users group, a separate drawing will be held for each cartridge so you can put your name in the box of just the cartridge you want. Only one winner per family is allowed so as to give everyone a chance to win. The first drawing for the best cartridge will be held right at 7:00pm as an "early bird prize". The second drawing will be at 7:20 and the rest will be given away at 8:30.

Newsletters from other groups will be available, the library full a bunch of new programs just received will be open, cartridges will be given away so this meeting is not one to miss!

NOTE Nominations for 1987 officers will be taken from the floor this month with voting to be held at the December meeting, if you don't come to the meeting you might be nominated for something! ...Gary Cox

QUESTIONS AND ANSWERS

1) Can the use of speech and sound be done at the same time in a x-basic or basic program?

The answer to that is partially yes. However, it can only be done using the resident vocabulary built into the speech synthesizer using XB and 32K or TI BASIC with Mini Memory. I have also seen it done in TI BASIC with E/A. Then of course it can be done in Assembly Language. Using CALL SAY in XB to access speech causes the computer to stop everything else while it is speaking. However, by using CALL LOADS to access the speech synthesizer the computer can speak and do something else at the same time. See

the May 1986 Mid-South 99 User Group newsletter page 9 for details on how to do this...G.C.

2) Is it at all possible to run your speech, sound, etc. of the computer through to the speakers of a stereo?

Yes. Several methods can be used to do this. The first way is to use the Forti music card for the PEB that will allow the connection of a stereo system to play music in quad stereo through the special Forti software. Another way is to connect the sound output of the console with a stereo system. This would not give stereo sound but would give a better presentation of sound than a TV or monitor. Check with Al Doss as I think he has his computer connected to his stereo... G.C.

3) Can files be uploaded or downloaded VIA modem between two computers using tape recorders?

Partially yes. The only terminal program (program used to communicate over the phone line) available for the TI that does not run from disk is the Terminal Emulator II (TE2) by TI. The TE2 does not allow the transferring of files VIA cassette although I recall something about the manual saying data files can be transferred VIA modem using TE2 but I am not sure about this. The way you could transfer a program when you just have a cassette recorder is to load the program into memory. Then connect your modems and the sender must type SAVE RS232 and the receiver must type OLD RS232. If after 10 seconds the transfer does not start the person who typed OLD RS232 must hit FCTN 4 and retype it. During the transfer numbers will count down at the top of the screen and when the numbers reaches 0 pick up the phone as the file transfer is completed. The receiver of the program now has the program in memory and should procede to save it to cassette tape using SAVE ER1. (consult your computer manual for more details as this is discussed.)

1) I do appreciate the operation of the two programs : PF KEYS and CLOCK (memory resident programs). But I often wonder if they could possibly be loaded at the same time by altering their memory locations. Perhaps another combination is possible through programming, would this work?...Marsha Ellis

This depends on how much room PF KEYS and CLOCK occupy as only a certain amount of invisible program space is available in the 32K where both of these programs reside. This seems possible but would require someone with more knowledge of assembly than myself to do the converting work.... However, included in the Gram Cracker new XB (on the new utility disk) most functions of PF KEYS and CLOCK as well as many others are provided at the same time... G.C.

2) After loading and using FUNLWRITER and then DISK MANAGER 1000. I seem to have a problem returning to FUNLWRITER. What is the correct way to return?...Marsha Ellis

Some versions of FUNWRITER have a built in selection on DM1000 to return to the FUNLWRITER disk. If the DM1000 does not have a selection to return to FUNLWRITER you will just have to terminate

and reload FUNLWRITER.

3) I use several different LOAD programs for disk directories and have written several of my own for special purposes. Even so I am curious as to why the listing always comes up in alphabetical order. Is this routine part of the T.I.DOS?...Marsha Ellis

The alphabetical information is saved onto the disk index so the computer keeps an index on which information is provided on how the programs should be listed which is alphabetically. Maybe Dicko will write us a detailed explanation of what the computer is actually doing here...G.Cox

Editors Note: Questions or Answers for the Newsletter. Can be sent to John Craig Jr., Rt 1 box B6-A1, Atoka, TN 38004...AJ

RAPID COPY

Rapid Copy, the best new quick track copy program has just been released from Texaments. This program is completely compatible with all controller cards (TI, Myarc, and Cor Comp) and is just simply, QUICK. The quickest time for copying a disk is 30 seconds if the disk is already formatted with the TI controller. The longest time to copy a disk is 90 seconds if one is copying a double sided/ double density disc unformatted. As this is the first track copying program that will copy those unprotected disks quicker than DM1000, it is a must. The time saving itself, is almost 10 minutes on a single copy. Several other features are available while using the program. You can Format and verify your disks, catalog, and turn on verification to make sure that you get a good copy when copying a program. It also has the ability to load the respective Disk Manager for the Controller card if one exists. So for \$14.95, this program can be purchased from Texaments, 53 Center Street, Patchogue, New York 11772. Author of this program is Barry Boone...Beery Miller.

SHOPPERS CORNER

Beery Miller has a Volksmodem 12 (1200 baud) for \$150 including cable, a TI keyboard (just keys) for \$5 and a TE2 cartridge for \$8. If interested call Beery at (901) 726-5551 or leave him a message on Risky Business BBS (24hrs) at (901) 726-5625.

Micky Nance has a Myarc 512K card for sale or trade. He will take around \$200 for the card or will trade it for a Curcomp 512K. Call him at (615) 459-6721.

Jonathan Leslie has a Dizware Business Package for \$30, TI Mailing List for \$10, XB and manual for \$30. Call Jonathan at (901) 377-5709.

Check out the "For Sale" table at the meetings for other deals. Games "N" Gadgets in the Mall of Memphis is still carrying TI compatible software as the only place locally to carry software for TI owners.

XB TUTORIAL #12

By Funlweb Farm of Australia
(continued from last month)

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 LJGD) have a built-in function called WAIT. XB doesn't have this command (and the cynic might suggest it doesn't need it) 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 millisecc. 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 than 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 outward 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=A*2
50
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 it's 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. 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(S( ) :: ..... :: CALL
REF(P,AY( ),B( ))
4000 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 GO to 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 SPL 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 seen 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 TXE 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=1 5)-(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 picking up a substitute CALL KEY routine to return direction pad values even if a joystick is used.

(To be continued next month.)

PROGRAMS

Recently James Johnston (a group member in Texas) sent me the following program which runs in Extended BASIC. However, by taking out the multiple line statements (::) and typing each statement on one line by itself in BASIC it should work also.

```
100 ! LINE OF WINDMILLS
110 CALL CLEAR
120 FOR A=1 TO 4 :: X#=BEG#("1/-/",A,1):: X=ASC(X#)::
CALL HCHAR(10,1,X,32):: NEXT A :: GOTO 120
```

"Line of windmills" is a nice cute program but I thought the idea deserved more development than simply twirl around. So I spent a little time experimenting with the program adding sprites and more lines to do other things and finally ended up with the result below which gives an interesting "sprite show". You may notice by looking at my program code that it in no way resembles the original but sne if you can see the resemblance as the program runs though. Because of the use of sprites this program will only run in Extended BASIC.

```
100 ! Bouncing Dancing Sprites
```

```
110 ! By Gary Cox (Nov. 1986)
120 !
130 ! Mid-South TI99/4A Users Group
140 ! Memphis, Tennessee
150 !
160 CALL CLEAR :: RANDOMIZE :: J=16 ::
CALL SCREEN(2):: CALL CHAR(33,"0000000000FF00")
170 CALL CHARPAT(73,A#):: CALL CHARPAT(47,B#)::
CALL CHARPAT(45,C#)
180 FOR I=1 TO 28 :: CALL SFRITE(#I,46,16,50,130,12,0)::
FOR K=1 TO 30 :: NEXT :: NEXT I
190 FOR I=1 TO 28 :: CALL MOTION(#I,4,0):: NEXT I
200 FOR I=1 TO 16 :: CALL COLOR(#I,I):: NEXT I
210 FOR I=12 TO 1 STEP -1 :: J=J+1 :: CALL COLOR(#J,I)::
NEXT I
230 CALL COLOR(1,7,2):: CALL HCHAR(19,16,33,3)
240 FOR I=1 TO 28 :: A=INT(RND)::
CALL MOTION(#I,-A,INT(RND))::
CALL SOUND (10,(A+10)0,2,300,2,1000,2):: NEXT I
250 FOR I=1 TO 26
260 CALL CHAR(46,A#):: CALL CHAR(46,B#)::
CALL CHAR(46,C#):: CALL CHAR(46,B#)
270 CALL DELSPRITE(#I):: CALL SOUND(100,-7,2)
280 NEXT I
```

Try these two programs...Gary Cox

TIPS

Many users of FunWriter have discovered a problem with FunWriter that after returning from certain sections like going from the utilities back to the main menu and when Editor is selected the word rap is no longer working. This is because you are now using Editor/Assemblers Editor where there is no word rap. Although this is not the true E/A Editor it has been adjusted to simulate it where control codes can not be entered. To get the regular TI Writer editor make sure the first two selections on the menu are Editor and Formatter. Selection #2 can be changed by pressing selection #6 to change options in selection #2. So when you want the E/A editor make selection #2 say Assembler and when you want TI Writer's Editor make selection #2 say Formatter as Formatter must be selection #2 for the correct Editor to load in.

Did you know you can have more than one RS232 in the PEB? However, the catch is the second card must be modified to be accessed as RS232/3 and RS232/4 and PIO/2. This modification will be done to TI brand RS232 cards FREE of charge if you send the card to TI. Therefore, with a Y cable on the serial port you will have RS232/1, RS232/2, RS232/3, RS232/4, PIO/1 and PIO/2. Call 1-800-TI-CARES for details. However, this modification can not be done to most other cards as you can not have more than one card in the PEB at one time as the computer would not know which card to look at...Gary Cox

THE RAVE KEYBOARD

The Rave 99 keyboard is one of the best new products to be produced in the TI market to this date. It is offered in two

different styles. One version is the Model 99/101 that I bought, and the other is the 99/84 model that more closely resembles the IBM keyboard. The 99/84 model includes a numeric keypad, separate function and control keys, and an adjustable keyboard tilt angle. I chose the model 99/101 for several reasons. It includes the software (all software is on an eeprom, i.e. no diskette required) to enable the load interrupt, and reset functions, plus has 4 different keyboard modes for the TI-Writer, Multiplan, and programming user. I will spend most of this article describing the Model 99/101 instead of the 99/84 as it is more limited. The model 99/84 is now \$149.95 and the 99/101 is \$164.95. I chose to go on and get the 99/101 since when I ordered it, the price was much cheaper than what I have listed (if you ordered in advance of product production, they made it cheaper for the purchaser) and it had more features than the 99/84.

The 99/101 mimics the TI keyboard in all aspects. The installation required that I remove the old keyboard, plug in a new interface card that would sit in its place, and then plug the new keyboard into that card. It took no more than 15 minutes to perform this operation. They also include a keyboard cover for the hole that was formed with the removal of the old keyboard so you won't get dust, etc. inside of the console. This version also allows the extension of a 25 foot cord where you can sit up to 30 feet away from your console with your keyboard if you wish for only \$9.95.

Now about the features of the keyboard. I will talk about the different modes of the keyboard that are available with pressing a key on the keyboard. In mode 1, the keyboard is almost identical with the normal keyboard except it has the function quit key disabled (F11 key) so you won't have to worry about accidentally resetting the keyboard. It also has the (") available as a single key press as it is now defined in a new position. In mode 2, all of the function keys (F1-F11) mimic the keyboard just as the original keys did. This means that if you prefer to use the normal function keys while using TI-Writer or Multiplan, you are still allowed. In Mode 3, this is the TI-Writer enhanced mode. This mode has the function keys redefined (some are still the same) and has invoked several new keys on the keyboard. There is now a HOME cursor key, an ESC key to go to the options, a BREAK key to release the left margin, and the option of deleting part of a line. Also there is a key for easier windowing from left to right, a TAB key, a BACK TAB key, a delete key, a roll up/down key, backspace key, 4 direction cursor control, and 4 keys for character insert and delete, and line insert and delete. In mode 4, this is the mode for enhanced versions of Multiplan (I have not learned Multiplan so I will try to explain it as much as I can). The arrow keys give the ability to move left, right, up, and down in all cells. You also now have the ability with one key to roll up and down within your spreadsheet. The home key has been enabled along with character delete, a TAB key, and ESCAPE key. There is also a new key in this mode that has been enabled and that is the PRINT key. This will allow you to go straight to the print option from anywhere in the spreadsheet with one keypress.

There are several other features that are fully available in all modes. The numeric keypad has to be a huge bonus while using

Multiplan or any program that requires a lot of numerical data. The multi-directional keys while using Extended Basic 2.11 had made it much simpler in editing programs that I have written.

I looked upon the purchase of this keyboard as an easier tool to write papers since the keyboard has a much better response. One thing that I kept in mind was that Miller Graphics is talking about releasing a MS-Dos card. With this keyboard, it would practically operate like an IBM system should it ever make it to market. I have found one or two problems with this keyboard. They are nothing major as you are not going to lose a program or document, but instead relate to the powering up process. One minor problem I have found (I believe this is related to the Cor Comp Controller) is that occasionally it will not respond to the numbers at the main title screen. To correct this, simply typing function equals will solve the problem. The other problem is that when the keyboard is supposed to be able to support the load interrupt and reset options. The problem here lies in the fact that I apparently have one of the last modified version of the TI console. It was completely restructured from the black and silver models and has all the resistors, capacitors, and shields either moved or made differently. The problem was is that the numbering system of the components changed. I am presently waiting for Rave 99 to supply me with the new changes as it appeared that I had one of those 1 in a 1000 different types of consoles to install it in. I believe I would have to give this keyboard as a rating of an A. For those interested, the keyboard can be purchased by calling Rave 99 at (203) 242-4012 (ask for Rick) or (203) 872-9272 (ask for John) after Apr. One thing that this company is looking to do, though they are not sure if it can be done, is the possibility of an internal modem. It is just now in the "thinking about" stages and we will have to wait and see. From small outfits can grow great wonders...Beery Miller

FILES AND BYTES

from: BASICALLY YOURS by Rich Klein in the Sept 86 Chicago Times Newsletter

How can a file be recovered, if the File Descriptor Record (FDR) is all but nonexistent and the bit map is not adjusted? The first thing to be sure of is, that since the Allocation Bit Map has not been adjusted to reflect the sectors just written to, is that NOTHING be written to the disk until the file has been fixed. If the disk is written to, the exact same sectors you just wrote to will be reselected and written to again, causing your hard earned data to go silently away never to be seen again. Once data has been overwritten, it can never be recovered.

Assuming the data on disk is still intact, it can be recovered by using a disk sector editor. There are several good Editors out there, including one from our own group's Todd Kaplan. I have Advanced Diagnostics from Miller's Graphics. A calculator capable of HEX, DECIMAL and BINARY conversions is helpful. A piece of paper and writing instrument are musts. Also, 32K and whatever module the sector editor can load in will be necessary (usually Editor Assembler, Minimem, or Extended Basic).

Before we get started, it would help to know that most disk sector editors have two display modes; ASCII and HEXADECIMAL. It is very helpful to use both, although you can get by with just HEX. You would have to convert ASCII characters to their hexadecimal representation and vice-versa. Let's do it:

1. Load sector editor following directions.
2. Catalog disk with bad file on it. Since the FDR has been started with a name on it and the File descriptor index (sector 1) has its location, it will show up on a disk directory.
3. Count from the beginning of the directory down to your file. Since the files are alphabetized, the location of the file descriptor for your file will be in the same order on sector 1 (FDR). Write down the place number of your filename in the directory listing.
4. Bring up sector 1 for display on your screen (hex display). Notice the four digit hexadecimal numbers. These two byte values indicate the sector location for each of the filenames you saw when you catalogued the disk. Since they are in the same order as the disk directory you saw, if you count the four digit entries from the beginning until you reach the number you wrote down, the value at that location will indicate what sector your File Descriptor record is on. Write that number down.
5. The next thing to do is find your actual data. If your file was opened as a DISPLAY type file, then you should have no problem finding your data; assuming you know what you entered, of course. If your file was set up as an INTERNAL type file, then you should have no problems if you entered words and not just numbers. If you entered numbers, then I suggest you read up on RADIX 100 notation Pg. III-13 Basic Reference Manual. In INTERNAL format, text appears pretty much as entered, but numbers bear absolutely no resemblance to the way they were entered. If it's an INTERNAL number file, it will be extremely difficult to recognize your data. We'll proceed on the assumption that your data is text or a DISPLAY type file. Scan through the sectors and make a note of any sectors you see your data on. We'll use this info to (re-)build the FDR.
6. When you get to the last sector on the disk that you find your data on, skim through the sector until you see the first 0FF (HEX display, of course). This is the end of the file marker. Write down the byte location of this marker.
7. Next, let's use the information gathered on the sectors used to adjust the Allocation Bit Map. The Bit Map is located on sector zero. It starts at Byte 56 and continues to Byte 100 if you've got SS/SD; 145 for SS/DD or DS/SD; and 235 for DS/DD. Each byte in the Bit Map represents the status of eight sectors on the disk. This is because each byte of the Bit Map contains eight bits, or BINARY digits. Each Bit represents the status of one sector on your disk. Since a binary digit can be represented by a zero or a one, a zero indicates the sector it represents is unused and a one indicates it is in use.

To find the status of any given sector, divide the sector in question by eight. Add 56 to the resulting value. The integer portion of the result is the byte number in the bit map, and if you multiply the fractional portion of the result by eight, you will get the bit position for your sector in that byte. Read the value at that byte and enter it into your calculator. Next, convert the number to BINARY and read a one or a zero in the bit position (rightmost is bit position zero indicating units and increasing to the left). You will see a zero if it is a sector that your "lost" data is on or if it is an unused sector. If you want to place a one in that position to indicate the sector as used, simply add or "OR" that Binary value to the present one on your calculator. Then convert that value back to HEXADECIMAL and write it to the disk.

```
Example: Bin. 1101 1111   Hex. DF
          OR   10 0000   20
          -----
          111. 1111     FF
```

--NOTE: Some and probably most sector editors require that you "EDIT" a sector and then "WRITE" that sector to disk in two separate operations to prevent accidental erasure of valuable data. See the documentation that should have been provided with your Editor program.--

8. Once you have updated the bit map, you can rest easy. Your data won't be lost if you write anything else to the disk from this point on. The next thing to do is update the File Descriptor Record for your file. The file type and record length and data cluster pointers need to be established.

Let's do the easy part first and set up our file type and related info. We'll go byte by byte and explain each as we go along. We'll also examine our possible options for each byte.

- BYTES 00-09 Filename. If filename is less than ten characters in length, the remaining bytes are filled in with space characters (Hex 20, CHR# (32))
- BYTES 10-11 Reserved. All zeros.
- BYTE 12 File Status.
 - HEX 00 = DISPLAY /FIXED
 - 01 = PROGRAM
 - 02 = INTERNAL/FIXED
 - 80 = DISPLAY/VARIABLE
 - B2 = INTERNAL/VARIABLE
 Write protected files have Hex 08 added to them.
- BYTE 13 Maximum records per sector. 256 divided by record length. Convert to hex and place here.
- BYTES 14-15 Number of sectors used. Count your sectors found and place hexadecimal result here. This is the number you would see on a disk directory, minus one because this count does not include this (FDR) sector. BYTE 16 End of File Offset. This is where you would place the byte count in the last sector of your file where you found the Hex FF. This is used by VARIABLE and PROGRAM type files to keep track of the end of useful info on the last sector of the

file.

BYTE 17 Logical Record Length. This is either the length you specified when opening the file or a default value. Convert record length to Hex and place here.

BYTES 18-19 a) Number of Fixed Length Records or;
 b) Number of Sectors used by Variable Length Records. If you have Fixed length records, then you'll have to go back to the sectors and count each record and place the Hex result here. Sorry about that. Make sure that if you are repairing a file as you read this; be sure you write this sector out before going back or you will probably lose the changes you already made to this sector.

If you are using Variable Length Records, specify the number of sectors used by your file (not including this one, of course). There are two bytes in this field and the value must have the bytes reversed before being placed here.

For example: Hex 100 sectors used equals Hex 00 01.

BYTES 20-27 Reserved. All zeros.

BYTES 28-255 This is the hard part. THE DATA CHAIN POINTER BLOCKS This is where we tell the disk system where all the parts of our file are. When we finish discussing this block, we'll get into this section more.

If you look back at byte 12, you may understand why a broken file comes up as a DISPLAY/FIXED 0 file. This is because a file type of zero indicates this kind of file and, since the rest of the sector is filled with zeros, the record length also comes up zero. Next time you see this on a disk catalog you'll know this is a broken file. I had a problem once when using the IEII module and downloading a file. It seems that no matter how many times I tried, the module would not close the file. Naturally, that's the filetype I saw every time I catalogued the disk. This was before I owned Advanced Diagnostics, and at the time, I didn't know what this meant.

Data Chain Pointer Blocks Each block is a 3 byte (six Hex digits) group that indicates the starting sector of this block and an offset that indicates how many additional sectors make up this block. There are just two problems with this. The starting sector and the offset are intertwined and, if there are any blocks previous to this one, then the offset from the previous block is added to this one.

To untwine the sector info from the offset, copy down the first three byte value in three pairs of two hex digits. Now take the right digit from the center pair and place in front of the first digit of the first pair and take the left digit of the center pair and place it after the third pair of digits. You now should have a pair of three digit Hex numbers. The first number is the starting sector and the second is the offset. They are three digit values because you can express a value up to 7095 with three hex digits. Since there are only 1440 sectors on the most crowded disk the II can handle, then this is plenty. Even a QUAD density disk would have only 2880 sectors.

20

So the starting sector is easy. It merely indicates the sector that this block starts on. The offset for the first block is the last sector of the first block minus the starting sector of that block. The offset for any block after the first is the previous offset plus the additional sectors that this one provides to the file. Let's see. If we've got a program that starts at sector 39 and the first block ends at sector 41, then 41 minus 39 is 2. Even though sectors 39, 40 and 41 add up to three sectors, the offset is two. The next block starts at sector 46 and ends at sector 46, occupying one sector. The offset for this block is 3: 2 from the previous block and one for this block. Let's encode this data. Sector 39 is Hex 027 and the offset is 002.

SECTOR	OFFSET
027	002
0 27	00 2
27 0 2	00
27 20	00

So we place Hex 27 20 00 at bytes 28, 29 and 30 in the FDR for our file. We'll encode the second block as follows:

Sector 46 is Hex 02E and our total offset is 003:

SECTOR	OFFSET
02E	003
0 2E	00 3
2E 0 3	00
2E 30	00

So we place Hex 2E 30 00 in bytes 31, 32 and 33. If we were to fill up the rest of the FDR with data chain pointers then we would have about 76 blocks of data on the disk. Big file!

I hope that this article will be useful as well as informative to you. I know that I enjoyed finding out just what all that Hex jibberish means on sector 0! Until next time...Jonathan Leslie

THE UBIQUITOUS AMP 57-30360

I know that most of you immediately recognized that designation, so I won't even bother to mention that it is the part number for the standard "Centronics Parallel" printer connector on the "parallel printer cable". I won't even ask how many of you know that there are 36 pins in that connection. However, I will ask the much more trivial question of "How many and which pins need to be connected in order to get a 'Centronics Parallel' connection functional?" (For those of you who remember the answer from 'Not to worry...it's RS-232', the answer is not the same. For those of you who don't...don't guess 2.)

If you have never wondered about the parallel cable (and don't intend to start now), skip to the next article. For those of you who are still reading, the answer is a bit like the one for RS-232c...it depends on how well you want to have the printer connection to work. Probably the absolute minimum is 10...A data, 1 ground, and the Data Strobe...but I wouldn't want to rely on that any more than I would on the 2-pin RS-232c connection. What I

will present is a "Centronics Parallel" pin-out which should (i.e. 99% of the time) cover your needs.

First, let's cover the initial set of pins. The previously mentioned pins (Data Strobe, Data, and Ground) need to be augmented by the Acknowledge, Busy, Paper End, Select, Input Prime, and Fault signals. This should be a total of 6 pins. There, that wasn't so difficult, was it? What? You want to know what to connect where and why?

Okay...I guess that was a little brief. Well, The Data Strobe signal is connected to Pin 1. (By the way, all pin references are at the printer end of the cable.) This is the computer's signal used to co-ordinate the reading of the data lines. Speaking of Data Lines, Data Line 0 (used to transport the Least Significant or 0 Bit) is connected to Pin 2. Data Lines 1 through 7 are connected to Pins 3 through 9 respectively.

Now we'll skip to the Acknowledge Line. The Ack Line is connected to Pin 10 and is used to indicate that the printer is ready to receive data. Oddly enough, right next door at Pin 11 is a signal that is used to indicate that the printer is NOT ready to receive data... the Busy Line. (In some cases it is possible to omit one or the other of these signals, but it is not necessarily true that the printer is ready to accept data if it is not busy.) The line at Pin 12 is the Paper End Line. This pin is handy for the obvious reason that a printer without paper may have cause to alarm the operator; however, there is a similarly useful signal at Pin 32...the Fault Signal. The Fault Signal is used to indicate that one or more of the following conditions is true:

1. Paper End is detected,
2. Ribbon End is detected,
3. The printer top cover is open,
4. Carriage, Character, or some other error.

(It is worth noting that the separate Paper End signal gives the ability to distinguish a "normal" error from an "abnormal" error...After all, if you are using single sheets of paper, you probably know that the printer is out of paper whenever it finishes a page.) Returning to where we left off, Pin 13 has the Select Line attached to it so that the computer can know if the printer is selected (some times called "On Line" or "Ready"). The final active line is the Input Prime line which is connected to Pin 31 and is used by the computer to trigger a printer reset or initialization.

You may have noticed that I skipped over and avoided the Ground connection. There is a reason for that. There are actually about 14 ground connections; however, you can usually get away with making only one connection and then cheating by "jumping" the others. What you do is pick out one of the Grounds, I generally use 16, (you have a choice of Pins 16 or 17 or 19 through 30) and connecting the remaining wire to it. Then you take a series of short wires ("jumpers") and connect 15 to 17, 17 to 19, 19 to 20, ..., 29 to 30.

To summarize, (other than the Ground wire which is passive) the

active computer to printer signals consist of the Data Strobe, B Data Lines, and the Input Prime signals. The printer, in turn, provides the Acknowledge, Busy, Paper End, Select, and Fault signals to the computer.

Of course, I suppose we could find a way to connect up the printer through the Cassette Tape Port...MYARSE DOES IT AGAIN...R.D.Wilson II

MYARSE DOES IT AGAIN!

After several months without any movement in the market place, MyArse has once again poured forth with some startling revelations. Your raving reporter has had the glorious opportunity of exploring MyArse in person and, I must admit, it was quite an experience!

The latest sot filler for your Mobious Expansion Device is the Stereo, Eight-track Back-ground Unit (Catalogue number SETBACK, List price \$179.95). This unit provides answers to many of the questions that have plagued programmers for a long time...and questions for answers programmers have never even asked. The unit is capable of recording and playing back programs and/or data on any of eight channels, which means that you will experience less cartridge insertion problems because when the tape rewinds you have a total combination of eight files (0 to B programs and B to 0 data) immediately available to the read/write heads. In addition, for those of you who have found that programming is more easily accomplished to music, the binary information is written in a low density format which uses individual musical notes for data storage, thereby providing the background music so essential to the truly artistic, epic code.

But that's not all, so don't fill out your order slip yet! In addition, those intrepid engineers at MyArse have designed a modification for the standard II Extended BASIC cartridge designed to assist another minority group. The Spelling-Optional Liberal Error Decoder BASIC (Catalogue number SOILED BASIC, list price \$199.95) was designed for those of you who never can remember how to spell the reserved words (or choose to use reserved words for variable names). This marvelous system employs a pseudo-expert system to provide BASIC code liberally adjusted by SWAG-Function analysis to determine the intended meaning of any character string bracketted by one or more spaces. (The Beta-test models had some initial difficulties with the space character when bracketted by spaces but, by employing a basic WAG-Function algorithm, these difficulties have been overcome.) Although the actual algorithm is patented, I understand that it is based on the Von Neumann-Nabelnachsinnenbesitzergreifung theory.

MyArse may have outdone itself in responding to the needs of the II Users when it developed the Virtual Dynamic Tape back-up system (catalogue number Virtual-DTS, list price 14.95). This is another card to be inserted in your MOPEd and you will never understand how you ever lived without it. Those of you who have had the embarrassment of having obliterated some program or data which you had talked a friend into spending hours entering and which you were going to copy (not copy over) will be glad to know that a

solution has come from MyArse. The Virtual-DTS card will be providing you with a virtually constant virtual back-up of everything going on in your system. No more will you have to suffer the embarrassment of meekly answering "I was going to..." when asked that critical question "You did make a back-up, didn't you?!!!" You can now confidently answer "YES!" because you WILL have a (virtual) back-up. To restore from the virtual back-up, all you need to do is insert a diskette or cassette in the appropriate device and enter "RESTORE VDT\$,dev,typ,amt" where

dev = DSKn for diskette drive n
CAS for cassette tape;
typ = ASCII or ASCII data file
BINARY for binary data file
BASCY for mixed binary and ASCII file
BASIC for a BASIC program
PROGBIN for a binary program image;
amt = a decimal number of bytes to be restored.

The Virtual-DTS will then create a restored file of the appropriate style and length on the indicated device using a slight variation on the SWAS-Function previously mentioned.

Well, I've got to go analyze the latest output from MyArse, so I'll have to close for now...R.D. Wilson II

MAILBAG

Looking through some of the recent newsletters that we have received from other groups across the world I read in the September Ottawa newsletter about a constructing a simple device to kill the color on the TI using a switch to switch between black and white and color. Not having color is sometimes nice when color is not needed as the display usually looks better without the color. There was also an article on how to make Multiplan perform similar to Lotus 1 2 3...

The HV99'er of Australia continues to have an excellent newsletter (more like a magazine). The September issue has some good tips on adventure games and the usual tutorials on BASIC, Assembly, disk sector access and more...

For those interested in Computer Aided Design (CAD) take a look at the Sept Pittsburgh PRU newsletter.

We receive many other great newsletters including K-Town of Knoxville, a very big LA 99'ers, Ottawa of Ontario Canada, Chicago, Tips from Tigerclub by Jim Peterson just to name a few! Check them out at the newsletter table at each meeting...Gary Cox

NEWSLETTER EXCHANGE

I know this subject has probably been covered a number of times before, but I would just like to add a few thoughts for your consideration. Our newsletter "Tibbits" is mailed out to a number of other user's groups every month. In exchange for our newsletter these other user's groups send us a copy of their newsletters. Some groups we have exchanged newsletters with for

years, others for just a short time. The newsletters we receive range in size from a few pages that can be read in just a very short time to what appears to be magazines dealing with the 'highly technical' inner workings of the TI-99/4A and must be pondered over for days until the understanding becomes apparent. So in these newsletters there is something for everyone from Gary to Bicko. There is much more information contained in the newsletters that we receive than we could possibly reprint in our own newsletter.

The point I would like to stress is that if you have not taken time to browse through our group's library of newsletters from other user's groups you may have missed the answer to some question you may at one time have had or you may have missed a program you had been looking for or some new idea that you may not have thought of. At any rate the newsletters that we receive are all very good and very informative, from the smallest to the largest and we should all be glad to have them.

The newsletters are always available at the meetings for your inspection and may be carried home as long as they are returned promptly at the next meeting. The newest of the newsletters are usually sorted out on one of the tables and the older issues of the newsletters are filed away in folders in a box where they can be easily viewed.

Lately I have been working on an index for information contained in the newsletters by subject so that any one having a question, for instance, on the new 'C' programming language or new software or hardware available, would be able to go to a newsletter that would cover that particular subject. It will be a monumental task, but not impossible, and may be very beneficial for our members. I guess this is just one way a person can get involved in supporting the group and I don't mind because the benefit I receive from the group far out-weighs the cost of the time and material to do this. I just hope you appreciate it by coming by the newsletter table and checking out some of the newsletters we receive.

And a special note to all of you who send your newsletters to our group on the exchange basis we would like to THANK YOU for your time and interest in keeping the TI-99/4A one of the best supported computers you can have today!...db

EDITORS NOTE

I have been Editor for the Newsletter for approximately six months now and I now can relate to the workings of a newspaper. I have always heard of STOP THE PRESSES AND DEADLINES. I never really understood the workings of a newspaper. I had always been told that the article submitted was not always as submitted. Now that I have been editing the newsletter, I can sympathize with the editors. Most of the time, it is time to go to press and there is a page empty just as there is now and it is either leave it blank or write some article to fill in the space. That is just what this is.

There are some ups and downs with being the Editor of the

Newsletter. The ups are getting the finished Newsletter in the mail and thinking that we have put out one of the best Newsletters around and feeling mighty proud. We have one pretty terrific group. Several people really extend themselves to make our club what it is. Knowing that we are recognized in other parts of the world is very important and makes it all worthwhile. Without the help and cooperation of several people this would not be possible. A great big thanks to all the work of some of the members who have submitted articles Gary, Michael, Jonathan, Richard, Ralph, David, Deery, John, Dirk and last but not least The HV99'ers of Australia and especially the people of Funnelweb Farm and anyone else who submits an article for next month's Newsletter.

I would like some input from the group as to whether we should keep the 24 page Newsletter or reduce it to no newsletter or somewhere in between. I am not gifted as a writer as you can guess but I do enjoy seeing that it gets out every month and I am willing to give it my best but my best is limited. I can only continue with your help!!!!

In order to be able to get the Newsletter out on time, I must have the articles by the 1st Thursday of the month. (This includes maps, too)....Al Doss

TIME-IN / TIME-OUT

by db ferguson

E T I R P E L E D L	BREAK
C S M K E L I E I C	CGINC
N D A C A I T B S J	CONTINUE
E A G B X E T I T N	DELSprite
U E E N R R O A T	DISTANCE
Q R B N I O T B N I	IMAGE
E U I N C N I O C N	LIST
S P G E T M E T E U	OPTION BASE
E N O I T I S O P E	POSITION
R A N D O M I Z E D	RANDOMIZE
	READ
	RESEQUENCE
	SPGET
	STRING
	SUBEXIT

Circle each letter in the words as you find them in the puzzle. The unused letters will spell a very uncertain "TIME" of the year!

PROTECTION

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

24h* TIBBS-BULLETIN BOARD
901-357-5425 300/1200 baud

DISCLAIMER

The Mid-South 99 Users Group is not affiliated, sponsored by or 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 GREEN newsletter and/or dollar signs (\$\$\$) 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 label. The letter 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!

PLEASE NOTE LARGE TYPE IS AVAILABLE PH. 743-6781

CALENDAR

MEETINGS: November 20, December 18, January 15 (3rd Thursday!)
WORKSHOPS: November 29, December 27, January 24 (4th Saturday!)

OFFICERS

Gerald Smith	President	901-363-6273
Dick Vandenberg	Vice-President	901-274-1892
Ralph Wilson	Secretary	901-382-0795
Mac Swope	Treasurer	901-363-3080
Pierre LaMontagne	Chairman - Library and Sysop	901-357-8110
Adolph Butler	Chairman Cassette Library	901-948-0732
Al Doss	Editor #1 - Newsletter	901-743-6781
Gerald Smith	Editor #2 - Newsletter	901-363-6273
Gary Cox	Chairman - Program	901-358-0667
Gary Cox	Information and Assistance	901-358-0667
Mac Swope	Chairman - Equipment	901-363-3080

GROUP MAILING ADDRESS

Mid-South 99 Users Group
P.O. Box 38522
Germantown, Tn. 38193-0522