

CLEVELAND AREA TI-994/A USER GROUPS NEWSLETTER JULY/AUGUST, 1990

OFFICERS	NORTHCOAST	TI-CHIPS	MEETING DATES
CO-PRESIDENT	STEVE BAGSTED 1-933-5977	MATT ANDEL 676-9759	NORTHCOAST 1:30 P.M. TI-CHIPS 10 A.M.
CO-PRESIDENT	BOB KAGY 1-255-2609	GLENN BERNASEK 238-6335	EUCLIDIAN ROOM N.ROYALTON LIBRARY
TREASURER	FRANK JENKINS 283-8526	LIN SHAW 235-3912	EUCLID SQUARE MALL STATE RD & RT 82
MEMBERSHIP	CHUCK POULIN 731-6473 361 E. 280TH ST. EUCLID, OH 44132	JOHN PARKEN 331-2830 4172 W. 217TH ST. FAIRVIEW PARK, OH 44126	THIRD SATURDAY THIRD SATURDAY JULY 21, 1990 AUGUST 18, 1990
SECRETARY	CHUCK POULIN 731-6473	DENNIS LIKENS 842-9627	SEPTEMBER 15, 1990
LIBRARY(DISK)	MARTIN SMOLEY 1-257-1661	HARRY HOFFMAN 631-2354	OCTOBER 20, 1990
TAPE & MODS)	TOM NELLIS 475-4067	JOHN PARKEN 331-2830	NOVEMBER 17, 1990
HARD COPY)	DICK ALDEN 1-352-9172		DECEMBER 15, 1990

As you will note on the masthead, this is a two-month issue. In prior years, we have tried to give you something extra in this issue, but I didn't this year for a couple of reasons. I didn't have the time, and also felt there was a lack of interesting articles to warrant the additional printing and postage expense.

I would like to make it known now that you will need a new newsletter editor at the end of 1990. I will have been doing this for 5 full years at that time, and would like to be able to use the limited time I have on my TI for other things. In addition to the newsletter, I am still active in getting new programs for the library, in writing up the catalogs, etc. Thus, I have very little time to explore a lot of the new programs I have purchased or to use them on a regular basis. We are lucky here in Cleveland to have regular and semi-regular contributors and rarely have to "borrow" from others to fill our space. I am sure that the membership will continue to support the newsletter as in the past and it will continue to be a quality publication.

Also, we need HELP on the FREENET. Both of our sysops have gone on to other things. Perhaps some think it not very exciting to be a SYSOP on the FREENET since we don't have uploading and downloading of programs. However, it still can be a valuable source of information and communication. If anyone who has a modem and some spare time would like to volunteer, I have the SYSOP's number and password for the TI SIG. I was trying to go in for a while and post articles and keep up, but have not been able to recently. We desperately need to beef up our visability on the FREENET.

Harry Hoffman spent most of the day at the LIMA TI Faire copying disks from the LIMA library which will go into ours. As a result, we will have over 100 new disks. Since there won't be a newsletter to publish next month, I hope to have most of these sorted and labeled and out to the groups before long.

I have just gotten around to viewing the conferences on

tape from the '89 meeting and find that the information is timeless and interesting even a year gone by. I would hope that if you have a VCR, you will take a look at these as we have them for all three years LIMA has sponsored the Conference.

You will note that the article on CABLES starts with PART 2. I didn't feel that you were missing anything by skipping PART 1 as Bruce Rodenkirch has covered most of the background in his article on communications. I thought that this second part should be helpful since sometimes it is hard to find ready-made cables for our modems and printers and we have to "do it ourselves".

Also, I just didn't have room to squeeze in a printout from Ali Ulgen to accompany his article. However, I do have the booklet he mentions and you will find that even if you do not have a 24-pin printer what can be done by some simple changes in the program. It is very instructional and can give you some good hints when you need to do some printer formatting modifications for a specific printer or program.

Have a good summer and we will be back in September. Don't forget that both clubs will meet in August as usual even though the newsletter will not be printed. Mark your calendars now so that you won't forget.

Contents	
:EXECUTIVE NOTES - TI-CHIPS.....	1:
:EXECUTIVE NOTES - NORTHCOAST.....	2:
:COMPUTER COMMUNICATIONS - Bruce Rodenkirch - NC.....	3:
:MISCELLANEOUS RAMBLINGS - Steve Bagstad - NC.....	4:
:NEWCODES - Ali Ulgen - NC.....	4:
:FORSALE.....	5:
:TI-BASE TUTORIAL - Martin Smoley - NC.....	6:
:CABLE INTERFACES - Guntis Sprene - Westchester NY....	9:

EXECUTIVE NOTES - TI-CHIPS
By Glenn Bernasek

The weather was beautiful and it was the Saturday before FATHERS DAY, but an enthusiastic contingent of the TI-CHIPS still showed for the monthly meeting.

Lin Shaw's treasurer's report showed a balance of \$900 for the month of May, and our membership chairman, John Parken, reported that we are holding at fifty three members, with four (4) renewals due for the month of June. Check your mailing label, and see if you are one of the "expiring four". It would be a shame to allow the benefits you receive through your membership in the TI-CHIPS to be lost by oversight. Remember, the most advancements for the TI-99/4A were the result of membership in user groups such as the TI-CHIPS.

John also offered to start a series of articles on '101 WAYS TO USE THE TI-99/4A' in the CLEVELAND AREA TI-99/4A USER GROUPS NEWSLETTER. He asked that if anyone has any ideas, please drop them off for him to use. It sounds like a great idea!

Harry Hoffman reported that the Chips disk library has increased by 100 FILLED disks of new titles obtained at the conference at Lima this May. From what Harry described, it looks as though the TI users might never run out of "quality" software! Glenn Bernasek reported that Harry spent the ENTIRE day at Lima producing the disk library additions he had mentioned, and Glenn asked for (and received a round of applause for Harry Hoffman.

Matt Andel led an active discussion on the conference at Lima. He reported that the Chips broke even on the 1200 baud modem which was raffled off at the conference. Glenn Bernasek said that the video tapes of this year's conference were most valuable in that one could not possibly attend every presentation. Jack Koryta volunteered to produce copies of the conference video tapes. Harry Hoffman purchased, for the Chips' lending library. This is the THIRD year that Jack has provided video tape copies for the Chips. Thank you Jack!

It was reported that Chris Bobbitt, of ASGARD, stated that the development of the long awaited PRESS was being discontinued in that the author of this program wasn't able to devote the time required to overcome the problems in developing PRESS into a working program. However, Chris Bobbitt demonstrated the item that most people were interested in anyway. That was a useful and sophisticated spell checker for the TI-99/4A and the GENEVE. This spell checker, called SPELL IT!, proved to be a very popular item at the conference. However, both Glenn Bernasek and Ron Markus strongly advised that anyone who bought SPELL IT!, make sure that they register their copy in that this spell checker is in its' infancy, and debugged or upgraded versions will most likely be coming out.

It was also reported that a new HFDC card was coming out in competition with the MYARC card. Just when this will happen is anybody's guess.

One additional item discussed was the executive meeting held at the conference. Two subjects were reported to have

been discussed. The first was that the Lima User Group DOES NOT intend to host these conferences forever! And they would be more than happy to attend an OHIO conference hosted by some other user group. The other item was the possibility of a "REGIONAL" newsletter or article pool for the TI-99/4A and the GENEVE. This would enable all users access to all articles written on these systems. This proposal is, at the present time, in a feasibility study stage, and much work has to be done before it is formally presented for user group approval.

Matt Andel attempted to demonstrate his working copy of SPELL IT! on the club's system without success. Evidently some problems develop when copying SPELL IT! is attempted. Matt promised that he will have a working demo of SPELL IT! at the next meeting.

Les Kee demonstrated another construction program written in Extended Basic. This program used the built-in trig functions in the TI ROM to develop the X-Y quadrant coordinates required to construct a round surface above (and intersecting with) a rectangular surface. The computed values are dumped, in tabular form, to a printer.

During the swap, shop and talk time, Glenn Bernasek demo'd a new 2-Liner program of his called TIMYCAT. This routine quickly reports the disk name, sectors remaining and a column formatted report of file names on the disk to either the screen or the printer.

This report was done in the absence of Dennis Likens. Dennis was on duty with the Coast Guard this weekend, and boy did we miss him! Hope to see you all next month on the 21st! CHECK YOUR RENEWAL DATE!!!

EXECUTIVE NOTES - NORTHCOAST
6/22/90

The June 16 meeting was attended by about 25 people. Items of interest included:

-discussion of obtaining Micropendium's programs already prepared on disk, a service they offer for \$40/year. Approved was the purchase of these, so depending on how soon we begin to receive them, they will become available at regular group meetings. -the main demo was Wes Richardson covering Crazy XB, assembler fundamentals, and a math program he is working on that is not yet "ready for publication". Thanks, Wes.

Howie Winkler was unable to attend with his "new?" ramdisk(s), but will hopefully make it to the July meeting. The plan for now is to take some quick looks at new software, particularly anything people may have obtained at Lima. We'll shoot for 5-10 minute maximum per item, so if anyone wishes to show something new, please bring it to the meeting (call me first if you can). This should be commercial or fairware items.

Also will definitely discuss plans to make good use of some of our treasury funds, so bring any ideas you have to an open forum on the subject. See you on July 21 at 1:30!
Steve

Second in a series.

COMPUTER COMMUNICATIONS by Bruce Rodenkirch

As I explained in the previous issue, this is basically a reprint of an article written by Tim Margush in *HARDCOPY*, the newsletter of the Akron Area Commodore Users Group, September, 1989 issue. I will be adding comments here and there but Tim did most of the work so thanks again Tim.

CHIPSIDE CHAT by Tim Margush (cont'd)

Modems usually communicate with computers via an RS-232 (Ed. serial) interface. This standard specifies a 25 pin connection. Pins 2 and 3 are the IN and OUT signals respectively. Other important pins that are monitored by most communication software are the carrier detect (CD, pin 8), (Ed. Also called DCD), request to send (RTS, pin 4), clear to send (CTS, pin 5), data set ready (DSR, pin 6), data terminal ready (DTR, pin 20), and ring (RING, pin 22). CD is used to indicate that the modem has detected a carrier signal on the line. RTS and CTS are used in half-duplex operation to switch between send and receive modes. DSR indicates that the modem is ready to accept data from the computer via the interface. DTR is controlled by the computer and indicates to the modem that it is ready. RING is the most obvious and indicates that the modem has detected a ring signal on the telephone line. The computer usually sets DTR to cause the modem to answer a ring.

(Editor: If you are planning to make a cable to connect your TI RS-232 to a modem, the above applies with the following exceptions. For the TI, I have found that pin 2 at one connector should be connected to pin three at the other end and vice versa. Most other computer RS-232's use "straight through" wiring. Pin 1 is ground, Pin 7 is the signal ground and this means the grounding wrap on the other signal wires if your cable has these. Pin 4 does not apply. The other pins are used and should be connected to the same pin at each end of the connector with the exception of Pins 2 and 3 which are crossed. This is a total of eight wires, or seven if you wish to tie Pins 1 and 7 together.)

Using modems, it is possible to transmit data from one computer to another, even if the computers are of different types. I have used my modem on my C-64 to send (upload) a large mailing list to a large VAX mainframe computer and then, using an IBM compatible, downloaded the same file and am now using it in a database system on that machine. Bulletin boards are unattended communications programs that automatically answer incoming calls and respond to various requests from the calling computer. Our club's bulletin board runs on a C-64 computer and can store files that can be used by C-64, C-128, IBM Compatibles, Amigas, and any other brand of computer.

Tim Margush

This concludes Tim's article. I think their club BBS runs on an IBM clone at the present time. The Akron phone number is 794-0209 and it runs on 300/1200 baud. Full use of the board is reserved for members of the Commodore Users

Group.

Perhaps you are wondering why the RS-232 is needed. Why can't the computer send the data directly to the modem? The RS-232 acts like a "middle man" between the computer and the modem. It communicates with both the modem and the computer telling the computer to stop sending data when the modem or printer has an overloaded buffer or storage area. In addition to this the RS-232 compensates for the differences in electrical protocol between the modem and the computer. The output voltages to the modem and the input voltages from the modem or other external device range from +25V max to -25V min with an undefined zone of +5 to -5 volts. A signal falling in the undefined zone will be ignored! The modem circuitry is set to transmit and receive signals from an RS-232 device.

The computer operates on TTL, or Transistor, Transistor Logic. It uses a +5 volt power supply and the zero or plus states have specific voltage requirements to transfer and receive data. The computer sends binary data, the strings of zeros and ones that make up the ASCII bytes. By general agreement, in TTL a logic chip will recognize a voltage of 0.8V max as a logic zero, and 2.0V min as a logic one. The chips are designed to put out signals to the next chip of 0.4V max for logic zero and 2.4V min for a logic one. Note that there is a 0.4V difference here which is called the "Noise Margin". This means that if there is noise in the system which raises or lowers the signal pulse by as much as 0.4V the chip will still recognize the pulse. The 1.2V zone between the maximum acceptable voltages is called the Indeterminate Region and signals falling in this region are not recognized. The TTL system is not as resistant to noise as other logic systems but the computer is a relatively low level noise environment and the high switching speed obtained with TTL justifies its use. Noise is further minimized by generous use of capacitors tying the signal lines to ground providing a path for high voltage noise spikes to leave the system. The design of the board with respect to component and trace placement in addition to additional protection from surge protectors is further insurance from erratic signals.

Now that we have some of today's technical expertise in our brain cells, let's jump back in time to have a look at the development of data communication. When man decided that drums or the human voice, even amplified by large horns, were inadequate for sending information over many miles, he devised ways of using smoke signals, flashing lights, flags etc. to send information. These were limited to "line of sight" distances and were rather cumbersome. To send signals even further, relay stations were set up. The Russians had a 1200 mile line of semaphores set up between Moscow and the Prussian front in the 1700's. The semaphore used lights with adjustable louvres which created a pattern for each letter and number. Since each character had to be repeated for verification, it was very slow.

The discovery of electricity gave the inventors another tool and various ways of sending information were tried. The first practical method was proposed in 1753 which required a wire for each character. The system would work but was too expensive and bulky for long distances. In 1836 Samuel Morse developed an apparatus for sending information

over a single wire. He had worked out an electromechanical device that made a pencil mark on a revolving cylinder of paper. An electromagnet or solenoid moved the pencil against the paper and held it there until the electrical signal was stopped. In other words the secret was to have a CODE of marks and spaces which could be decoded at the receiving end to obtain the information.

The system was further refined in later years to use a paper tape, punched by a typewriter like machine, which could be sent through the sending machine. The holes in the tape allowed the electrical contact to be made, sending fast reliable information to the receiving apparatus which also used a paper tape and pen to record the incoming data. It was also discovered that skilled operators could decipher the message by listening to the clacking of the receiving machine. This did not replace the "hard copy" method of data transmission but supplemented it.

This concludes the second in this series which started out to be a two piece series. If you are still with me I will describe the evolution of the teletype machine into the modern age and tell about how this technology is used in amateur radio.

To be continued.....

MISCELLANEOUS RAMBLINGS - Steve Bagstad - 6/22/90

This is the conclusion of a topic I felt might be of general information for TI users. So, here goes...

Mail order shopping notes (con't from last month):

IF YOU HAVE PROBLEMS: 1) If you have not received your order as promised or if the item is defective, immediately notify the Seller in writing referring to your order by description, price, date, account number, and order number if available. Make sure you keep a copy of the letter. 2) If you complain by phone, send a followup letter to confirm what was said. 3) If you think the merchandise is defective, reread your product instructions and your warranty carefully to be sure you don't expect features or performance the product isn't designed to give. Then contact the Seller for instructions. Don't return it to the Seller until you have been instructed to do so. 4) When returning merchandise, make sure you keep the shipper's receipt or packing slip. Your right to be reimbursed for postal cost is determined by Seller policy. 5) If you have completely discussed your problem with the Seller and are still not satisfied, write to the consumer complaint agency in the Seller's state. If you paid for the merchandise by credit card, you may have rights to withhold payment under a Federal law called the Fair Credit Billing Act.

The Microcomputer Marketing Council of the Direct Marketing Association, Inc. advises that if you ever have a problem, remember to deal first with the Seller. If you cannot resolve the problem, you can also write MAIL ORDER ACTION LINE; c/o DMA; 6 E. 43RD ST.; NEW YORK, NY 10017.

Hope this series has been helpful... Steve

NEWCODES

Ali Ulgen - NorthCoast 99ers Cleveland, Ohio

I received the PagePro 99 V1.6 upgrade from Asgard on May 17. I was most excited to try the high resolution printing utility; PP-HR2. This is the first program incorporating 24-pin code in the TI/Geneve community. I must mention that the PP-HR2 utility file is a freeware program from Ed Johnson and should not be considered part of the PP upgrade. Mr. Johnson's suggested request of \$5 makes this utility an excellent bargain. We have to support people like Ed since they come up with such good stuff!! For example, if you like PP, then you probably have the PAGEPRO UTILITY DISK from Asgard. On this utility disk is a preliminary version of a FONT LINE EDITOR by Mr. Johnson. This is V0.1 and is a limited line Editor; however, it's a lot of fun and makes one look forward to the final version. Ed is also working on a utility to print PAGEPRO pages in multiple sizes such as 4 to 8 times normal, and it will be in 8 and 24-pin codes.

Having spent some time and effort with this utility, I decided to write, for others, who might like to know the simplest way to modify the PP-HR2 file for graphic modes other than the default values. It's so simple, you might be surprised: only 1 character change on line 295 will provide a different graphic mode. If your printer is an 8-pin, you can still modify the codes.

First, I must stress that any changes you make to the file be done on a backup disk. Second, while it is true that only line 295 needs to be modified to change the output, you may want to make other changes. However, I suggest you make as few changes as possible and save the results under a new filename. Stay away from the first 10 lines of the program as it is easy to destroy the system integrity of the original file. Third, if you would like to see samples of printouts in various modes and codes, ask Deanna Sheridan to see the booklet I prepared for the NorthCoast 99 members. A quick look at this picture, in just about every graphic mode possible, may save you a lot of time.

WHAT IS ON LINE 295 AND WHAT DOES IT MEAN?

```
295 IF P=1 THEN K$="K" :: L$="l" ELSE K$="* " ::  
L$="*!"
```

The variable "P" indicates which pin mode was selected by you on lines 210, 225. If you select 8-pin, then P=1 else it's 24-pin and P=2. Line 295 contains all of the codes required for i-pin, 24-pin and also single or double density mode; selected on line 240. What line 295 says is: If 8-pin, single density is selected then K\$ on line 300 is "K"(CHR\$(75)). If 8-pin, double density is selected, the L\$ on line 210 is "l"(CHR\$(76)). Note that this is low-speed. If 24-pin, single density is selected then K\$ on line 300 is "* " (CHR\$(42)CHR\$(32)). DON'T BE MISLED BY THE SPACE AFTER THE ASTERISK, WHICH IS PART OF THE CODE! The asterisk is the code for universal graphic selection, simply signals the printer that a graphics code has been selected, leaving that job to the code that appears next. In the example, the space or CHR\$(32), signals the printer

that the graphics code selected is 24-pin, single density. If 24-pin double density is selected, the L\$ on line 310 is "!.:" (CHR\$(42);CHR\$(33)). These are the only 4 possible default variations.

Here are my suggested code modifications. If you change the "L" to "Y" (CHR\$(89)), you'll get 8-pin, double density, in hi-speed. The grade-off between L and Y is that in the hi-speed mode the adjacent dots are not printed. Y is faster because it prints fewer dots than L. I think that the resolution is 7% better with the "Y" mode.

The first selection of 8-pin, single density where K\$ is "K", is a mode I'm never going to use. Change the "K" to "Z" (CHR\$(90)) for an 8-pin, quad density, low speed printout. This mode does not print the adjacent dots, the printout is smaller with a distortion that appears to squeeze the picture horizontally. Select the Wide printout since the Narrow produces too much distortion.

Change the " * " to "!.:" (CHR\$(42);CHR\$(33)) which is 24-pin, double density. You may not that this is the default selection found on the last string variable on line 295. The reason I'm doing this is that all subsequent variations will be done on the last string variable. Another reason is that the default; " * " is a single density output that is the least desirable of the 24-pin codes.

The last variable string on line 295 is; L\$ = "!.:". Change it to "!.:" (CHR\$(42);CHR\$(39)) which provides a dump that is 24-pin triple density in low speed. The printout in this mode is 4" by 6 3/4" if you select Wide. A dark picture results in a printout that is too grainy. This is due to the fact that this is the only triple density mode which prints the adjacent dots. I can't see a need to use the overstrike feature with this code.

If your printer supports it, change the "I!" to "!.:" (CHR\$(42);CHR\$(40)) which is 24-pin hex density, low speed. Even though this code does not print the adjacent dots, the results are dark, small (2" wide), and distorted.

The last code to try changes "!.:" to "!.:" (CHR\$(42);CHR\$(38)). This is a code I tried by mistake, keyed 38 instead of 39 while working on the file. I would not have tried it otherwise, since this mode is supposed to be a display mode called; CRTIII. The manual does not say

anything about this code at all. It does provide an MD of 90 dots per inch, which produces an 8" wide printout that is distorted. If you select Normal, the output is very similar to the output as; 24-pin triple density, low speed which printed the adjacent dots. It's the same size dump, but the quality of the CRTIII (CHR\$(38)), is better than the CHR\$(39), the grainy output is replaced by a smoother finish. In spite of the small printout size, this is my second choice.

The PP-HR2 utility file sets the maximum number of dots per line at 360 for Normal/Narrow printouts and 720 for Wide printouts. Some of you may have read that in the TI-ARTIST PLUS addendum page this is called the XPRMAX (RECORD #3). You may also be familiar with the suggestion that this figure be around 1080. I'll show you how to change this figure after the warning.

WARNING: PP-HR2 is a short program file that is deceptive since it boggles up stack free memory space when running. A picture size, and the 8 or 24-pin mode, and Horizontal Dot Density per Inch of the code selected, as well as the XPRMAX figure all combine to quickly fill the stack free area, and you run out of room with an error message. Take these things into consideration when you make the following changes. The variable "E" on line 260 is set a 1 when you select a normal printout and 2 when a wide picture is selected. Every increment of 1 to the value of E adds 360 dots to the XPRMAX value. If you change the normal value to 3 and the wide value to 4, the XPRMAX values become 1080 and 1440. Suggest that you try E=3 with 24-pin triple density (CHR\$(42);CHR\$(40) or "!.") and E=4 with 8-pin, quad density (CHR\$(27);CHR\$(90) or "Z"). When E=4, you'll run out of room quickly depending upon your other selections. Beyond E=4, I'd be surprised if you get anything to print at all.

If I look at my sample picture only, I have to pick 8-pin, double density, hi-speed (CHR\$(89) or "Y") as my favorite. Am i disappointed with 24-pin? Yes, when printing graphic picture, I am. However, text with 24-pin is much better. NOTE: My 8-pin printouts should be different than the ones you'll get on an 8-pin only printer. The picture printed does have a limited effect on the outcome, and thus has to be considered.

FOR SALE:

2 Consoles - \$30 ea
CorComp Mini Expansion box with 32K, Disk Controller, RS232 \$150
2 DSSD Teac Drives - \$100
14 in Samsung monitor - \$100
12-baud Avatex Modem - \$50
Star NX 1000 Printer - \$150
Speech Synthesize - \$20
Many cartridges and software
ED KENNELLY 252-0621

OR SALE:

MYARC 128K Expansion Card - \$75 - Jerry Reising
1-933-3354 - AVON LAKE

TI-BASE - Version 3.0
TUTORIAL 20.1.1 By Martin Smoley
NorthCoast 99'ers - June 16, 1990
Copyright 1990 By Martin A. Smoley

PROGRAMS
CF#1
CF#2
CF#3
CF#4
CF#5
CF#6
S1/D
S1/D
S1/D
S1/D
S1/D
S1/D
S1/D
S1/D
S1/S
S2-/D
S2-/D
S2-/D
S2-/D
S2-/D
S2-/D
S2-/D
S2-/S
S3--/D
S3--/D
S3--/D
S3--/D
S3--/D
S3--/D
S3--/D
S3--/D
S3--/S
S4---/D
S4---/D
S4---/D
S4---/D
S4---/D
S4---/D
S4---/D
S4---/S
S5----/D
S5----/D
S5----/D
S5----/D
S5----/D
S5----/D
S5----/D
S5----/S
Dbn''/D
Dbn''/D
Dbn''/D
Dbn''/D
Dbn''/D
Dbn''/D
Dbn''/D
Dbn''/D
Dbn''/D
Dbn''/S

I am reserving the copyright on this material, but I will allow the copying of this material by anyone under the following conditions. (1) It must be copied in its entirety with no changes. (2) If it is retyped, credit must be given to myself and the NorthCoast 99ers, as above. (3) The last major condition is that there may not be any profit directly involved in the copying or transfer of this material. In other words, Clubs can use it in their newsletters and you can give a copy to your friend as long as its free.

SOME IMPORTANT STUFF ABOUT DISK FILES

Last month I tried to give you the idea that you can break down a large Db into several small Dbs and use them all in a normal manner. Actually you can, if you perform some regular disk housekeeping tasks. If you ignore these tasks your system will run slower and slower, and eventually you will have disk problems. Here is my explanation of the situation. The disk storage system for the TI is designed to use every possible space on your diskette to store data. Under normal conditions the system will start at the beginning of your disk and add the new data you wish saved to the end of the line, or next available blank section. You could compare this to adding toy train cars onto the end of a child's train set. Last month I created Dbs 74LS'S1, S2, S3, S4, S5 and several CFs. Let's suppose that these files are stored on your disk. I might USE S1 and APPEND some data. This new data is stored on the end of our data train. If I now USE S2 and APPEND some data, this data goes on the train after the APPENDED S1 data. Now I decide that the Command File named CF#3 is not what I want, so I Modify it and add two new lines of code to it. When it is saved back to disk the major portion of CF#3 will be placed in its original place, but any new data that is left over will be stored after the stuff I APPENDED to S1 and S2. As you can see, the data on a disk would eventually become a tangled mess, as I have attempted to depict in the right hand column on this page. This bad situation can be made much worse by using the computer against itself. An example of that would be to write a Command File (CF) that would use all five Dbs (S1 .. S5) at the same time, with the CF having the ability to move through all five Dbs and APPEND multiple records with little or no control by the operator. This could create a tangled mess of such proportions that the system would lose track of the data and start declaring disk errors. I won't go into the situation of deleting files from a disk and how that space is reused, but believe me it will make a bad situation much worse. I try to keep this situation in mind whenever I write CFs or do major data handling procedures. My corrective measures are as follow. Make a rough estimate of the size Db you will need and fill the Db immediately. This can be done by using a CF, such as SUBNUM2 from last month to fill an empty Db with partially blank records, or by manually using an empty Db and, in the APPEND mode, holding down the enter key until the desired record number is reached. After that you can use the EDIT mode to enter your data. The EDIT mode does not require additional disk space that will jumble the files. The places you need to worry are operations that add or subtract from your disk, such as APPEND or DELETE for records, or adding or deleting lines from a CF. In addition to good user habits you should make a File Copy of your working disk at least once a month. This is my procedure. I place my working disk in drive #1 and fire up DM-1000 from my HORIZON RANDISK. I press 1 for File Utilities, 1 for Copy/Move/etc., and 1 for Disk drive #1. After DM-1000 has given me a catalog screen of Drive #1 I press A for all and DM will place a C, for Copy, in front of all the files in Drive #1. Pressing (FCTN 6) tells DM to Proceed and after placing a new blank disk in Drive #2, I enter 2 as the copy destination. After looking at the disk DM tells me it's not initialized and I answer 'Y' to proceed. I enter a disk name that includes the current date so I can distinguish it from the rest of my junk and let the copying begin. A file copy takes more time, but it will reorganize all the files as you see them on the left side of this page in a neat close order. Now I use the new copy as my working disk until the next time I recopy the entire batch of files using this procedure. I also use this procedure on my randisk. I copy all the files off the work section of my randisk, as above, then I use DM to delete all the files in that section of the randisk, "be careful!", and then I copy all the files back to that section of the randisk. I consider this very necessary because a randisk can be jumbled and re-jumbled for many months or even years before any attempt might be made to clean up the files. Deleting and resaving any file, even Extended Basis or DV/80 Files, will eventually mix up the disk storage patterns. Well I have wasted a lot of time explaining why you should use File Copy to clean up your work disks once a month, so I better get back to the applications of TI-Base.

PROGRAMS
CF#1
C
CF#3
CF#4
CF#5
CF#6
S1/D
S1/D
CF#3
S1/S
S2-/D
S2-/D
S2-/S
S3--/D
S3--/D
S3--/S
CF#6
S4---/D
S4---/D
S4---/S
S5----/D
S5----/D
S5----/S
Dbn''/D
Dbn''/D
Dbn''/S
S1/D
S2
S3-
S4---/D
S5----/D
CF#2
S1/D
S2-/D
S3--/D
S4---/D
S5----/D
Dbn''/D
S3--/D
S2-/D
CF#3
S4---/D
S1/D
S5----/D
CF#1
CF#2
CF#3
S3--/D
S4---/D
S5----/D
S1/D
S2-/D
CF#4
S5---
S3--/D
S2-/D
CF#3
S4---/D
S1/D

Continued Next Page.

**TI-BASE - From INSCEBOT
TUTORIAL 20.1.2 By Martin Smoley
NorthCoast 99'ers - June 16, 1990
Copyright 1990 By Martin A. Smoley**

I took a look at the tutorials for last month and the month before and found them to be very confusing. I hope to clear up some of the confusion in the next couple months, please hang in there.

First, I hope you realized that the printouts of (74LS'S1 .. SS), as shown in the lower left corner of 19.1.1, are not complete. This multiple printout was produced by the CF named LSPRNT/C which is listed below. SET PRINTER=DSK2.LSPRNT redirects the printout to a disk file named LSPRNT, that's nothing new, but SET CRLF=OFF is new. This new command allows you to turn off the Carriage Return and Line Feeds when desired. In the past, when a disk file was needed, you had to go in with FunnelWeb and remove all the extra CRs and LFs to get the proper printout. That was a real pain in the neck for me while writing these tutorials. PRINT (f) is my symbol to print condensed format. From that point to the ENDWHILE is merely a demonstration of using a DOCASE within a WHILE loop instead of IF statements. It is just as easy to write IF LOOP = 1, USE DSK2.74LS'Sn, ENDF, for this situation, but it seems that I demonstrate IF statements every month. One important line is PRINT ALL ;FOR (CRS>0). This actually tells TIB to look at every record in the currently active Db and PRINT ALL the records which contain a CRS field that hold a value greater than (>) zero.

```
*      05/08/90      LSPRNT/C
CLOSE ALL
SET PRINTER=DSK2.LSPRNT
SET CRLF=OFF
PRINT (f)
LOCAL LOOP N 3
REPLACE LOOP WITH 1
WHILE LOOP<6
  DOCASE
    CASE LOOP = 1
      USE DSK2.74LS'S1
    BREAK
    CASE LOOP = 2
      USE DSK2.74LS'S2
    BREAK
    CASE LOOP = 3
      USE DSK2.74LS'S3
    BREAK
    CASE LOOP = 4
      USE DSK2.74LS'S4
    BREAK
    CASE LOOP = 5
      USE DSK2.74LS'S5
    BREAK
  ENDCASE
PRINT ALL ;FOR (CRS>0)
CLOSE
REPLACE LOOP WITH LOOP + 1
ENDWHILE
SET PRINTER=PIO.CR.LF
SET CRLF=ON
RETURN Copyright Martin Smoley 1990
```

In this situation, if you SET RECNUM OFF at the beginning of the CF and SET HEADING OFF after the first loop, you would give the appearance of one continuous file, not 5 separate DBs. After all, this set of tutorials is designed to show you how to use several smaller Dbs instead of one very large Db.

On the next page (20.1.3) I have listed LSEDIT4/C along with its sub-CFs, \SCRN/C, \ED/C and LSUSE/C. I have condensed the print because I didn't want to waste the space and also because it is a new version of LSEDIT3/C from last month (19.1.3). If the small print is confusing, you can compare it to last month's CF to check most of the code. The reason it is listed again is because I have made some changes, and also because it contains the main theory on how to handle multiple Dbs as one unit. This theory will be presented and re-presented in an effort to show you how simple it really is. The LSEDIT CFs in this series use this basic idea, open all five of the LS series Dbs at the same time, ask the operator which item they wish to edit, decide in which one of the Dbs that item might be found, go to or SELECT that area, search for the item and if found, edit that item. This is basically the same as if (by thinking) you decided which Db contained the item you wanted to edit, USED that Db, held down (FCTN 5) to leaf through the DB and EDITed the record if you find it. I'd like to take a closer look at LSEDIT4, even though it's a waste of time for those of you who understand TIBs language. First I always CLOSE ALL Dbs, so I know what is going on in the system. INSTALL ADD DSK1.\ED and \SCRN will take those disk files and ADD them to VDP Memory where TIB will use them as normal CFs, but faster. I'll cover the INSTALL stuff again later. LOCAL ITEM N 5, is the item we will tell TIB to search for. REPLACE ITEM WITH 1999 is my way of holding TIB in the WHILE loop in the middle of the \SCRN CF. You will see it as WHILE (ITEM<999) .OR. (ITEM<1688). It's kind of backwards and hard to understand, but simply stated the idea is this, if the number contained in ITEM is not between 998 and 1689 TIB will keep you locked in this loop until you enter a part number that is. This is an attempt on my part to assure a valid part number search. The exception to that rule is -1 which is the return path. LOCAL LOOP N 2 and REPLACE LOOP WITH 1 are my way of creating an endless loop. In Boolean, 1 means true. Therefore, the statement (WHILE LOOP) will loop forever, because when tested the answer for loop will come back true. You could get out of this loop by replacing LOOP with zero (0), somewhere in the CF, but we won't do that. Just before WHILE LOOP is the statement DO DSK1.LSUSE. This statement runs the CF LSUSE to set up our five Dbs in slots 1 through 5. DO \SCRN, is the command to run the CF named \SCRN from VDP RAM. This CF puts up the complete entry screen and asks you to enter the item number you want to edit. If you enter -1 the CF will terminate, but if you enter a valid number it will be stored in ITEM and TIB will jump back to LSEDIT4 and proceed down to the five major IF sections. IF ((ITEM>999).AND.(ITEM<1100)) seems odd, especially if you entered a zero (0), or possibly (01). You must not forget that we set the position of the LS series from 1000 to 1999, so zero (0), or 74LS00, will be searched for as COPNM 1000. If zero was entered, the first IF statement would be true. This would cause slot 1 to be SELECTed and \ED to be executed. The first command in \ED is FIND ITEM. All the Dbs should be SORTed ON the COPNM field. The sort should have been done automatically at the end of SUBNUM2 (19.1.2) last month.

Continued Next Page.

**TI-BASE - From INSCEBOT
TUTORIAL 20.1.3 By Martin Smoley
NorthCoast 99'ers - June 16, 1990
Copyright 1990 By Martin A. Smoley**

If the item is not found EOF will be flagged. This means the statement IF (EOF) would be true, and you would see the line ITEM NOT FOUND, and returned to \SCRN that you might enter another number. If the item is found, the EOF flag would not be up, and you would go to the ELSE part of that IF statement and EDIT the field that TIB found for you. This stuff is quite intricate, in the way it jumps from one location in the CF to another and back, but the general idea of what's going on is not complicated.

Let's look at some of the new INSTALL stuff which I have discovered, even since last month. If you look at the beginning of LSEDIT4 you can see that I have ADDED both \ED and \SCRN to the new VDP memory location which you can do with version 3.0 of TI-Base. This is great for people without RAM disks because the access speed is much faster than that of a

normal disk drive. The major discovery I made for myself, in the last month, is that the INSTALL area is more or less the same as disk space. As you can see by \ED and \SCRN it appears that INSTALL will support Comments, RETURNS, WHILE/ENDWHILE loops, IF/ELSE/ENDIF statements and I suppose, just about anything you can do in a disk based CF. The INSTALL space should be used for CFs or Macro Commands that you need frequently. \SCRN and \ED will be used for every loop of LSEDIT4. As I demonstrate in LSEDIT4, you also have the option of ADDing and REMOVE-ing CFs with each new major CF. \ED/C and \SCRN/C used approximately 1,000 Bytes of VDP Memory, but I still had roughly 1,000 bytes left to use. I already had \RES and several other small CFs in the INSTALL area, so there is ample space available for entry and informational screens. I have learned several tips about INSTALL. Do not attempt to REMOVE or ADD items to the INSTALL area using a CF that resides in the INSTALL area. REMOVE or ADD should be done from a disk based CF. REMOVE items from the INSTALL area in reverse order that they were ADDED. If you ADD, \ED then \SCRN, you must REMOVE \SCRN, then REMOVE \ED, or TIB will get lost. Also remember that DO \ED is the syntax in a CF, where as \ED is entered at the dot prompt for INSTALL area CFs.

! 06/06/90 LSEDIT4/C

```

CLOSE ALL
INSTALL ADD DSK1.\ED
INSTALL ADD DSK1.\SCRN
LOCAL ITEM N 5
REPLACE ITEM WITH 1999
LOCAL LOOP N 2
REPLACE LOOP WITH 1
DO DSK1.LSUSE
WHILE LOOP
DO \SCRN
IF ITEM<0
INSTALL REMOVE \SCRN
INSTALL REMOVE \ED
DO \RES
RETURN Copyright Martin Smoley 1990
ENDIF
IF ((ITEM>999).AND.(ITEM<1100))
SELECT 1
DO \ED
ENDIF
IF ((ITEM>1099).AND.(ITEM<1200))
SELECT 2
DO \ED
ENDIF
IF ((ITEM>1199).AND.(ITEM<1300))
SELECT 3
DO \ED
ENDIF
IF ((ITEM>1299).AND.(ITEM<1400))
SELECT 4
DO \ED
ENDIF
IF ((ITEM>1399).AND.(ITEM<1689))
SELECT 5
DO \ED
ENDIF
REPLACE ITEM WITH 1999
ENDWHILE
RETURN Copyright Martin A. Smoley 1990

```

```

! 06/09/90 \SCRN/C for INSTALL
SET TALK OFF
CLEAR
SET HEADING OFF
SET RECNUM OFF
WRITE 6,8,"Enter the right hand digits"
SET INVERSE ON
WRITE 2,6,"          ";
"          "
WRITE 3,6," 74LS Series Integrated ";
"Circuits "
WRITE 4,6,"          ";
"          "
WRITE 8,16,"          "
WRITE 9,16," EXAMPLE "
WRITE 10,16,"          "
SET INVERSE OFF
WRITE 12,6,"Manufacturer      You"
WRITE 13,6,"Part Number       Enter"
WRITE 15,12,"74LS221 = >221 < ENTER"
WRITE 17,12,"74LS01 = >01 < -1 "
WRITE 19,34,"TO QUIT"
WHILE (ITEM<999).OR.(ITEM>1688)
WRITE 22,3,"Enter ITEM Number = ) <"
READ 22,24,ITEM
IF ITEM<0
CLOSE ALL
RETURN Copyright Martin A. Smoley 1990
ENDIF
REPLACE ITEM WITH ITEM+1000
WRITE 22,3,"Company Part No. = "
WRITE 22,23,ITEM
WAIT 3
ENDWHILE
WRITE 21,3," Press FCTN 8 Then FCTN 9 "
WRITE 22,3," After Each Record Edit "

```

```

! 06/09/90 \ED/C for INSTALL
FIND ITEM
IF (EOF)
WRITE 21,3,"          "
WRITE 22,3,"          ITEM NOT FOUND "
WAIT 2
RETURN Copyright Martin Smoley 1990
ELSE
EDIT
ENDIF

```

```

! 05/06/90 LSUSE/C
SELECT 1
USE DSK2.74LS'S1
SELECT 2
USE DSK2.74LS'S2
SELECT 3
USE DSK2.74LS'S3
SELECT 4
USE DSK2.74LS'S4
SELECT 5
USE DSK2.74LS'S5
RETURN Copyright Martin A. Smoley 1990

```

Because of the speed of my RAM DISK, I use the INSTALL area mainly for Macro Commands rather than the CFs, but no matter how you use them the new features that have been added to TI-Base are fantastic. The ability to run a CF by typing \SCRN at the dot prompt instead of DO DSK6.\SCRN to me is wonderful. When you start to get the hang of this, reread page 8 of the May Newsletter, about Macros. I think you'll start using this feature more and more.

Continued Next Month.

INTERFACING COMPUTERS WITH DEVICES
by Guntis Sprene

This is part two of the series on connecting your computer to whatever you'd like to connect it to. This time I'll try to explain the RS-232 interface so that you should be able to hook-up a MODEM or serial printer to your computer.

First a few warnings. The RS in RS-232 stands for (R)ecommended (S)tandard. Which means that in some cases serial devices will have connectors other than the standard DB-25 (a good example is the 99/4A which has two ports sharing one connector and the AT style connector which uses a 9-pin instead of 25-pin connector. In most cases mis-wiring a cable will cause no harm; if you are at all uncertain as to what you are doing, seek help (I assume NO responsibility for any damages resulting from using this information!!).

The RS-232 interface is specified using a DB-25 connector. It has 13 pins on the top row and 12 on the bottom. The connector is shaped so that it can be mated only one way. The standard includes two "designations": DTE (Data Terminal Equipment - usually a computer or other terminal) and DCE (Data Communications Equipment - usually a MODEM). The "designation" of the interface determines how the pins are used. That will (hopefully) become clearer later.

The pins are specified as follows:

- 1 -- frame ground (FG)
- 2 -- transmitted data (TD) ->
- 3 -- received data (RD) <=
- 4 -- request to send (RTS) ->
- 5 -- clear to send (CTS) <=
- 6 -- data set ready (DSR) <=
- 7 -- signal or logic ground (SG)
- 8 -- data carrier detect (DCD) <=
- 9 -- 22 -- ring indicator (RI) <=
- 10 -- 23 -- data rate selector (->)
- 11 -- 24 -- external transmitter clock
- 12 -- secondary DCD <=
- 13 -- secondary CTS <=
- 14 -- secondary TD ->
- 15 -- transmitter clock <=
- 16 -- secondary RD <=
- 17 -- receiver clock <=
- 18 --
- 19 -- secondary RTS ->
- 20 -- data terminal ready (DTR) ->
- 21 -- signal quality detect (SQ) <=
- 22 -- ring indicator (RI) <=
- 23 -- data rate selector (->)
- 24 -- external transmitter clock ->
- 25 -- busy ->

The -> arrow shows a signal going to the DCE; the <= arrow shows a signal going to the DTE. In an ideal world, DTE and DCE devices can be joined by a straight through cable with the appropriate gender connectors. The cable would be wired:

DTE device	1	-----	1	
	2	-----	2	DCE device
	3	-----	3	
	4	-----	4	

In the 'real world' things are seldom as easy. An explanation of each pin's function follows.

pin

- 1 - frame ground. A connection to the AC safety ground.
- 2 - transmitted data. Data sent from the terminal device.
- 3 - received data. Data received by the terminal device.
- 4 - request to send. An indication by the terminal that it has data to transmit.
- 5 - clear to send. Response from the communications device that data can be sent to it by the terminal.
- 6 - data set ready. An indication that the communications device is ready. Usually raised after power-up (and after self-test passes).
- 7 - signal ground. A common reference for all signals.
- 8 - data carrier detect. An indication from the communications device that a valid MODEM tone has been received.
- 20 - data terminal ready. An indication from the terminal equipment that it is ready. Ususally controlled by the ON-LINE switch on 'dumb' terminals; software controlled with computers and terminal programs.
- 22 - ring indicator. A signal from the communications device that it has detected a ring on the phone line. Used to inform the terminal that it may have to raise DTR so the line can be answered.

One thing that can't be stressed enough is that you should be familiar with your devices' manual(s). I would even recommend that before buying something you check out the section in the manual that deals with the interface. A good manual can sometimes make life lots easier.

Many newer printers 'force' the condition of certain pins if no external signal is present. For example, if nothing is connected to DCD the printer may 'make' DCD high; this can be done with software or hardware (here I assume that everyone knows that all modern printers have a microprocessor and ROM that controls the printer).

This is where the 'magic' starts. The data output of a DTE device is pin 2; the data input to a DCE device is pin 3. If the devices to be connected are dissimilar (one DTE and one DCE) then the cable will have pin 2 wired to pin 3. The same thing holds true for RTS, CTS, DSR, DTR, RI, DCD so the cable is straight through as shown on first page.

If both device are the same (two DTE devices, for example) then pin 2 will be wired to pin 3, pin 3 will be wired to pin 2 (pin 2 [the output of a DTE] to pin 3 [the input of a DTE]). That's the easy part.

The terminal ready output of one device has to go to the data set ready input of the other device. This would be pin 20 wired to pin 6; and pin 6 wired to pin 20.

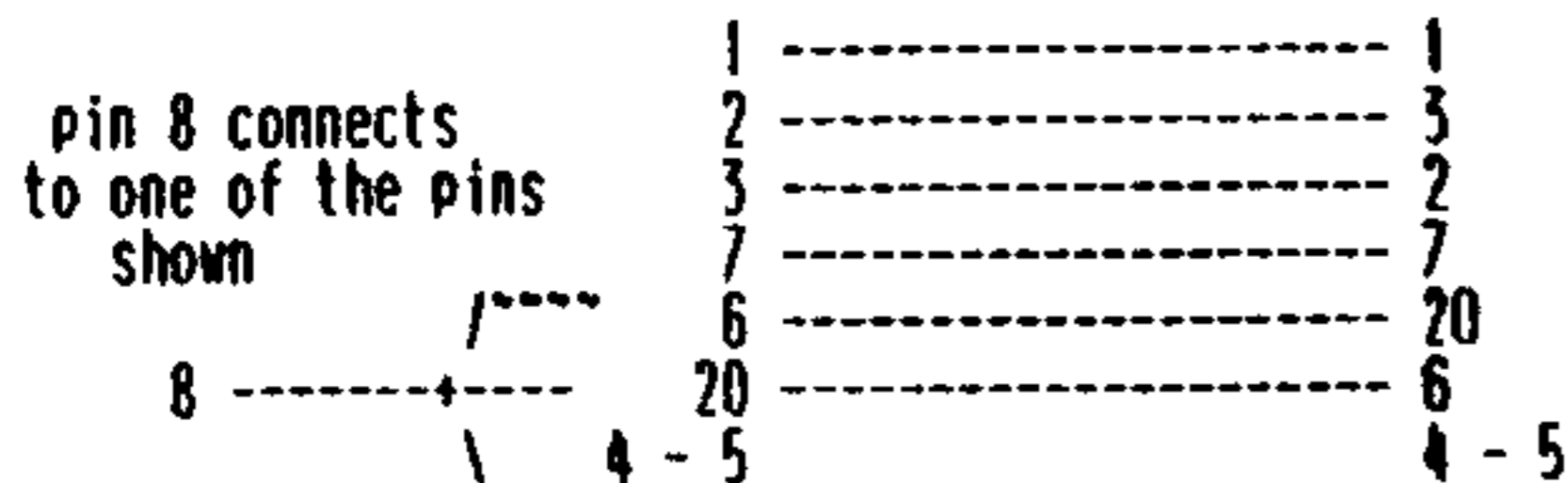
The request to send of a device goes to that same device's clear to send. Or, pin 4 would be connected to pin 5 at each end.

Usually, the data carrier detect pin needs to be high for a device to receive data. This can be accomplished by connecting pin 8 to either pin 6 or 20 or 4 or 5. Selection of where to connect pin 8 should be determined by reading the manuals of both devices. If pin 20 is used for a busy indication as well as terminal ready, then pin 8 should be

be connected to 4 5. If one of the devices toggles request to send, the pin 8 should be connected to either 6 or 20. Note that if both devices are the same (DTE DTE or DCE DCE) then pin 8 for one device should not be wired to pin 8 of the other.

The signal grounds of both devices are ALWAYS tied together. Frame grounds may or may not be tied together (if there is space I'll discuss the various problems which occur when frame and logic ground are tied together).

From the above, a cable to connect to similar devices (sometimes called a MODEM eliminator) would be wired:



If the basic cable doesn't work, then it's troubleshooting time. I'll serial port and a serial printer, both DTE devices. The ideas can be extended to other devices.

In most cases, the status and control pins (all pins except 1,2,3 7) use a negative voltage (low) to indicate false and a positive (high) voltage to indicate a true condition. For example, Ring Indicator goes from low to high when the MODEM detects a ring, and from high to low when the ring is gone. When a terminal or printer is OFF-LINE the DTR is low; DTR is high when the device is ON-LINE and ready to print (usually - sometimes error conditions could be signaled by the BUSY indicator; the TI 810 printer uses pin 11 for BUSY/FAULT conditions and has an option to have DTR high as long as the ON-LINE LED is lit or to have DTR also go low when a BUSY/FAULT condition occurs).

PROBLEM ONE: Lost data. You sending a file to the printer, it prints, but every so often parts of your document are missing.

SOLUTION: The computer is not sensing the printer's busy status. Determine how the printer signals busy (using pin 20 (DTR); pin 25 (seldom used); or some other pin (the TI 810 printer uses pin 11)) and wire the cable accordingly. The wire that signals printer busy could be tied to the computer's pin 6 (DSR), 5 (CTS), or pin 8 (DCD) (or another pin if the interface is non-standard, such as the serial ports on the 99/4A). Sometimes the printer and/or computer use X-ON/X-OFF busy control. In this case the printer transmits a control character when its buffer is full, and another when the buffer is empty. The usual codes are hex 13 for busy, hex 11 for not busy (or resume sending). If this occurs, then either the printer or computer will need to be reconfigured to match the other device.

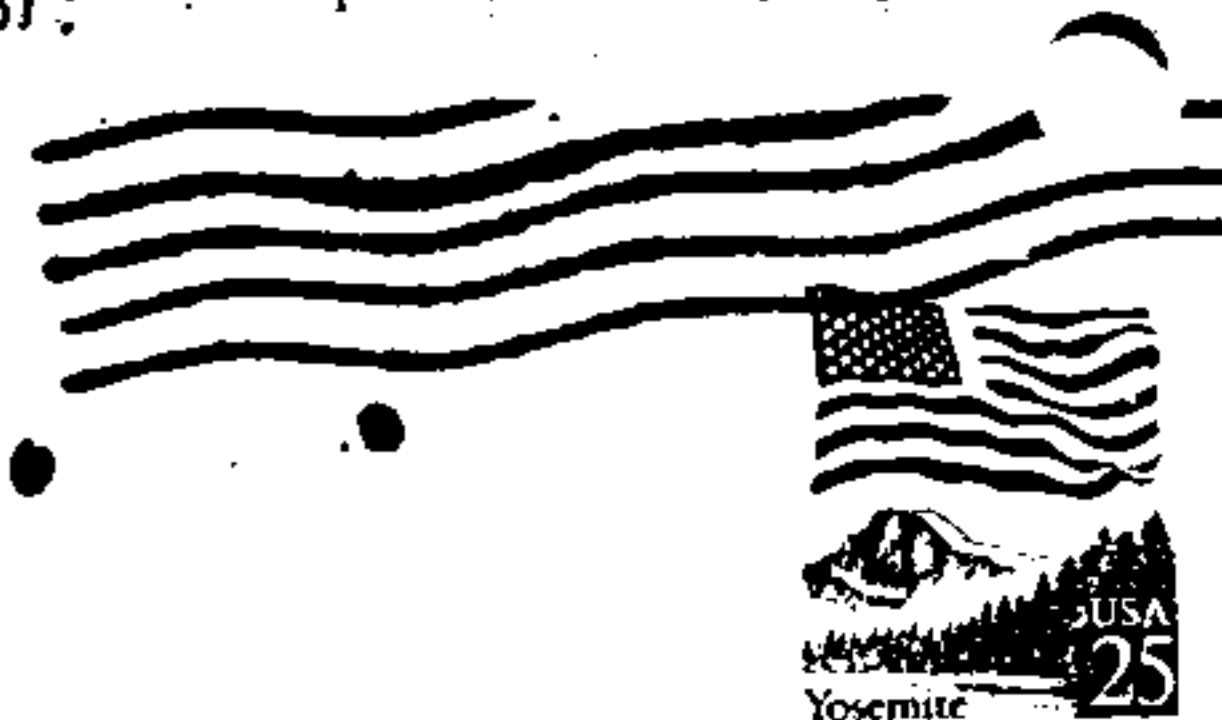
PROBLEM TWO: The computer sends data, but the printer doesn't print. The computer returns a status indicating that printing has finished, but nothing has actually printed.

SOLUTION: The printer is not receiving signals that indicate it should print. The presence of data on pin 3 is not enough. DCD, DSR, and/or CTS may have to be high. Consult the printer manual to determine which signals have to be high for the printer to print, and make the needed connections. Solving this problem may cause problem one to appear. Also, try reversing pins 7 and 3 on one end of the cable.

PROBLEM THREE: The computer is told to print, but just sits there, or, eventually, returns a 'device not ready' message.

SOLUTION: The obvious one is to make sure the printer is ON-LINE. If it is, then the computer is not receiving the device ready status from the printer. As in the above case, read the serial port documentation to determine which pins need to be high and make the needed connections. Correcting this problem may cause problem 2 or 1. (NEXT SECTION - PARALLEL INTERFACES)

CLEVELAND AREA 99/4A USERS GROUPS
C/O DEANNA SHERIDAN
20311 LAKE ROAD
ROCKY RIVER, OH 44116



CHECK YOUR EXPIRATION DATE.
THIS MAY BE YOUR LAST ISSUE!

FIRST CLASS

FIRST CLASS

FIRST CLASS

FIRST CLASS

Exp Date: 91/07