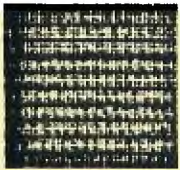




Delaware  
Nov 86.





# THE DATA BUS

ISSUE :  
VOL. 4  
NO. 10  
- - - -  
NOV.  
1986

## THE DELAWARE VALLEY USERS GROUP

DEDICATED TO THE II AND COMPATIBLE HOME COMPUTER FAMILY

P.O. BOX 6240 STANTON BRANCH, WILMINGTON DE 19804

4TH THURS. 6:30-9:30 P.M. CHRISTIANA MALL COMMUNITY ROOM

---

KENT COUNTY, DE COURTHOUSE    <CHAPTER>    DEPTFORD, NJ MUNICIPAL BLDG.

2ND THURS. 7:00-9:00 P.M.    <MEETING>    3RD MONDAY 6:45-9:00 P.M.

**MODEMS: Some Basics**

by William Gregory - Greater Orlando 99ers

A modem is a translating unit between the computer and telephone system. A direct connection from the computer and telephone system is not possible because the output of a computer are digital pulses. The telephone system is designed to transmit human speech. By converting the digital output into a series of audio tones, information can then be transmitted over the telephone lines. Converting digital pulses into audio tones is called modulation. Going from audio tones back to digital pulses is called demodulation. A unit that can do both functions is called a Modulator/Demodulator or Modem for short.

Since digital information is composed of logic "one" and logic "zero", we use two different frequency tones to indicate this. In a bi-directional telephone transmission, there would be a problem with only one set of tones. Data can be sent both ways over the same set of wires, but it wouldn't work if the same set of tones were used to transmit data in both directions. By having a Modem use two sets of tones, this problem is solved.

One set of tones, (1070) HZ for "zero" and (1270) HZ for "one" is used for originating data. Another set of tones (2025) HZ and (2225) HZ, is used for receiving data. The receiving data of (2025) HZ would be for logic "zero" and the receiving data (2225) HZ is used for receiving logic "one".

When data is transmitted over the telephone system, noise on the line will interfere with the communication if the data is transmitted too fast. By sending data at 300 bits per second maximum, this noise problem is solved. This rate is also called 300 Baud, and it results in a transfer speed of 30 characters per second. Higher Baud rates are now available. Special lines and equipment are needed for these rates.

The two most popular types of modems in use today do not require unusual equipment. One is the acoustic-coupler, and the other is the direct-connect. The acoustic-coupler is cheaper, but more likely to pick up noise interference. So: Take your pick!

**EDITOR'S NOTE**

This article is about 18 mos. old. Don't know about those special lines and with so many 1200 Baud modems, I don't think of them as all that special either any more. Wonder where we'll be in another 18 mos.

DELAWARE VALLEY USERS GROUP: NOV. 1986

DUUG EXECUTIVE COMMITTEE MEMBERS IN 1986

PRESIDENT .....TOM AUGUST  
 VICE PRESIDENT .....JIM DAVIS  
 SECRETARY .....LYNN ACQUARD  
 TREASURER .....TOM KLEIN  
 SGT. AT ARMS .....JIM FOLZ  
 DELMARVA CHAPTER CHR.....CHARLES BOWER  
 SO.JERSEY CHAPTER CHR.....ERROL LANSBERRY

DUUG BBS: (302)322-3999 Anytime  
 (609)429-7792 Anytime  
 (302)674-1449 6:00 PM-6:00 AM

-----  
 For general information, you may contact  
 TOM KLEIN Pa. (215)494-1372  
 GIL or KAY QUILLEN Del. (302)492-8189  
 JACK SHATTUCK Del. (302)764-8619  
 BUTCH FISHER N.J. (609)783-8276  
 -----

A Delaware Valley Users Group membership includes monthly newsletter (DATABUS), library and software privileges, plus other special benefits. Annual membership rates are: Family or Individual \$15; Students \$10; Newsletter only (beyond 75 mi) \$10.

PLEASE TRANSMIT YOUR NEWSLETTER COPY TO: The Data Bus Editor ---- Jim Folz, Telephone (302)995-6848, or use the DUUG mailing address shown on Page One. NEWSLETTER COPY WILL NOT BE ACCEPTED FOR AN ISSUE AFTER THE 2ND THURSDAY OF EACH MONTH.

-----  
 An article appearing in The Data Bus may be reproduced for publication by another TI User Group as long as acknowledgement is given to the sources as indicated. We encourage exchange newsletters; mail to DUUG business address shown on Page One.  
 -----

DUUG ADVERTISING RATES FOR THE DATA BUS:  
 1/4 page = \$ 5/issue, or \$ 45/12 issues  
 1/2 page = \$ 8/issue, or \$ 75/12 issues  
 Full page = \$15/issue, or \$125/12 issues  
 -----

DELAWARE VALLEY USERS GROUP MEETINGS

Plenary meetings: Delaware's Christians Mall on Rte. 7, at I-95 Exit 4-S, in the Community Room. Enter between J.C.Penny and Liberty Travel inside the Mall.

DELMARVA CHAPTER: Kent County Courthouse, Basement Conference Rm #25, Green & State Streets, Dover, Delaware. Use the Green St. side entrance.

SO.JERSEY CHAPTER: Deptford Municipal Bldg, Cooper Ave. and Delsea Drive, (Rtes. 534 & 47), in Gloucester County. Enter and park in rear of the building.  
 -----

CONTENTS OF THE NOVEMBER ISSUE OF THE DATA BUS:

Modems: Some Basics	Page 1
NOISE on The Data Bus	Page 2
DUUG Software Library Notes	Page 3
Menu Programming	Page 4
Assembly Cross Reference ... Part 1	Page 5
Paid Advertisement	Page 6
Axref Program	Pages 7-10

NOISE on The Data Bus  
 by Jim Folz

First, let me say, that the DUUG regrets having reported IN ERROR of the passing of Clint Pulley. In fact, Clint is still with us and I'm sure he will continue his fine work on c99.

And yes, it is true. I missed Lou Phillips' presentation on the MYARC 9640. After looking forward to the presentation for so long, I ended up babysitting.

Having committed two of the greatest sins for a newsletter editor, one might ask "where does one go from here?" To the news, of course-

Lou Phillips made a presentation to the South Jersey Chapter in October. While most were impressed with the computer's capabilities, some were disappointed by the absence of a production unit. An article on the computer was not available at press time. We hope to have an article in the December issue. The South Jersey Chapter is selling chances for the MYARC 9640 computer. See Errol Lansberry for details.

At the last Christians Meeting, the Constitution was amended. Recognition of chapters and membership classes were the main areas affected. Lynn Acquard can provide details.

Elections are around the corner. If you have an interest in serving or you know of someone else that is interested let the Nomination Committee know. The committee will probably be announced at the November Christians meeting.

Remember that the Christians Mall meetings have been rescheduled due to Thanksgiving and Christmas (both fall on the fourth Thursday). The November meeting is scheduled for November 20. Currently, the December meeting is scheduled for December 18. Watch this column for changes.

Note also that a special meeting is scheduled for Saturday, November 22. Program exchange, equipment maintenance/clean-up, and programming/equipment help sessions are planned. Time - 10:00 to 2:00 Place - Calvary Episcopal Church See map on page 5.

On Sunday, November 30, the Meadowlands Computer Fair will be held at the Meadowlands Hilton Hotel (Secaucus NJ, Rte 3 to Meadowlands (Industrial) Parkway then go south. Time - 10:00 to 4:00 Admission - \$3.50 For more information call (201)533-1991

Advance Notice!!! On Saturday, March 28, the Second Annual T.I.C.O.F.F. will be held at the Roselle Park High School, Roselle Park, NJ. Admission - \$5.00 Watch for additional details.

New Products Of Note!!! Asgard Software has announced the availability of two programs written in c99 (High Gravity and Total Filer). c99 enthusiasts take note.

IEEE 488 (GPIB) Interface!!! National Instruments carries interfaces for GPIB (used by Hewlett Packard). A board is available that converts RS232 to IEEE 488 protocol. Anyone interested might try this approach. I haven't tried it. For information call (800)531-4742 (outside Texas) or (800)433-3488 (inside Texas).

DELAWARE VALLEY USERS GROUP: NOV. 1986

## DVUG SOFTWARE LIBRARY NOTES

by Jack Shattuck

November marks our first Disk/Tape-of-the-Month, with a collection of Christmas items. Some of the works are by Bill Knecht, Stephen Foster and Sam Moore, Jr., favorite 99/4A music creators. The XBasic disk selections run almost 20 minutes, if you keep hitting menu choices to proceed to the next selection. Seventeen different melodies can be heard.

We run the loaders by using RUN DSK.XMAS.PROGRAM, which allows operating from any available drive. Just use the disk name instead of a drive number, with the period in front as well, and your controller searches for the disk, not the drive.

Tape selections, written entirely in BASIC, include some identical tunes, but variations also. SNOWMAN is a math game in which correct answers help build a snowman. Two tape programs use the Terminal Emulator II speech capability; THE NIGHT BEFORE CHRISTMAS, and DREIDEL, a Chanukah top game for the Jewish Feast of Lights, which starts Dec. 26th - sorry but we have a limited number of BASIC Christmas items! (For this month, not necessarily always, we used an all-BASIC tape.)

As always, you learn from program techniques, at the same time you enjoy the programs. If you're lacking a free disk/tape, pay an extra \$1 to the software librarian at the meeting you attend. \$1 is the basic contribution for any disk/tape-ful of programs you obtain from the library.

DECEMBER TOPIC: A VARIETY OF CATALOG/LOAD PROGRAMS for the Disk; Tape items not yet known.

In our last column, we compounded the tragedy of Tom Weitholfer, PILOT 99 author who had died of Cystic Fibrosis, by identifying the loss as the C-99 creator, Clint Pulley, instead. Fortunately, Clint is alive and well, and has provided advances from his initial pioneering work.

Our comments on InfoWorld's article about the CompuServe (CIS) battle with a discordant Apple BBS Sysop got a fair amount of play when colleague Art Byers of Central Westchester NY 99ers uploaded The Data Bus article onto the CIS TI Forum.

It seems that InfoWorld was ensnared in the misinformation campaign developed by the SYSDP who was charging subscribers to his own board, passing on CIS programs without authority and for his own profit, and had the chutzpah to advertise on CIS's own message base. The subject programs were NOT just public domain, as the Sysop had claimed.

CompuServe clarified its policy via a lengthy statement, which they put up without a menu access charge for a while.

Basically, CompuServe provides a value-added service by its structured data base, while authors retain copyrights to programs. Some authors will choose the unrecompensed distribution means of release to public domain channels. Whenever an individual releases a large group of programs to others, which originally were obtained via the CIS board, he passes along not merely a program or two but rather negates the copyrighted uniqueness of the CIS operation, for which subscribers have paid a fee to obtain. When the offender charges a fee himself, he violates the contract under which CIS provided the availability of those items to him. Further, in use of not just public domain but also commercially released and distributed programs, an offending party violates federal law protecting

creative and commercial rights of the author, in passing them along without due compensation or the author's consent.

We have discussed copyright issues enough in pages of The Data Bus to feel this special CompuServe quote is significant enough to repeat here:

" -- MAY I DOWNLOAD PROGRAMS FROM COMPUSERVE FORUM DATA LIBRARIES AND SHARE THEM WITH A FRIEND, OR UPLOAD THEM TO ANOTHER BULLETIN BOARD SYSTEM?"

In keeping with the spirit of the development of public domain information and shareware, it is not CompuServe's current policy to prevent casual redistribution of this type of information -- this is low volume and low frequency use or redistribution of information where no commercialism is involved. THIS MEANS THAT A CUSTOMER MAY DOWNLOAD A FILE AND SHARE IT WITH OTHERS FOR NO COMMERCIAL GAIN -- EITHER VIA A BULLETIN BOARD SERVICE, DISKETTE, OR OTHER MEANS. [ Emphasis here provided by The Data Bus. ]

"A subscriber may not, however, download a large number of files for redistribution via any means, nor is it acceptable for a subscriber to update another bulletin board regularly with files obtained from CompuServe.

"It's important to note that CompuServe cannot grant redistribution rights for programs clearly copyrighted by the author, unless specifically authorized to do so. Such permission must be obtained directly from the author of the program.

"In addition, mass distribution of public domain information or shareware is also prohibited. Mass distribution is defined as high frequency and/or high volume transfers."

--- From CIS "GO COPYRIGHT" File, 21 October 1986

SOFTWARE PROGRAMMING CONTEST RUNS THROUGH 12/31/86

Computer Shopper is having a software contest with entries taken through December 31 postmarks. Categories are: 1) BASIC/XB; 2) Assembly; and 3) Other, eg., C/99, FORTH, LOGO, etc.

All submissions must be entirely original; on disk; not previously sold commercially (including fairware) nor presently on any national network such as CompuServe; with only one entry per person regardless of category. Documentation is desired, either in the program or otherwise; source code must be included for judging. However, it will be kept confidential, as the judges (Howie Rosenberg, Ron Albright, Jon Zittrain) plan to upload entries onto CompuServe's TI Forum.

Prizes include hardware, software, plus many subscription items. Write for an entry form from Computer Shopper TI Forum, P.O. Box F, Titusville, FL 32781, or get a form at a DVUG meeting before the end of the year.

TI'S SOFTWARE IS STILL UNDER COPYRIGHT RESTRICTION

Attorney Gary Honeycutt of Texas Instruments' Patent Department in their legal office in Dallas, Texas, has responded to an inquiry as to status of copyright restrictions on TI-99/4A software which is no longer sold by TI.

The answer is the same as we reported in The Data Bus (Vol.3:8), Sept. 1985. Triton and others still do software marketing under TI copyright and retain that protection, including for basic tutor tapes and disks. Only TI Forth is public domain; User Groups' members may use both TI-Multiplan and TI-Writer upgrades. Otherwise, call Triton.

DELAWARE VALLEY USERS GROUP: NOV. 1986

Menu Programming  
by Jim Davis

A routine part of many BASIC language programs is user selection of an option. "User friendly" programming requires that the user not be required to remember obscure or arbitrary numbers or codes. Thus good programming practice displays a menu for the user giving the option codes or uses mnemonics that are easy to remember. In the latter case it is helpful to display for the user the mnemonic codes available only when an incorrect code has been used. After all, what is appropriate for the novice user of a program can be very tedious for the expert user. Yet it is desirable to accommodate the wants of all users. What follows are some examples of menus for use in BASIC programs.

The simplest menu is a list of options each associated with a number. The user keys in a number which selects the option;

```
100 PRINT " INPUT NUMBER CODE"
110 PRINT " 1 = DOG SHOW" ::PRINT " 2 = PONY SHOW"
120 INPUT " 0 = EXIT":I
130 IF I <= 0 THEN GOTO 999
140 ON I GOSUB 1000,2000
150 GOTO 110
```

The alphabetic codes can be used. If the codes are in order then the following is applicable;

```
100 PRINT " INPUT CODE"
110 PRINT " A = DOG SHOW" ::PRINT " B = PONY SHOW"
120 INPUT " X = EXIT" : I$
130 IF I$ = "" THEN GOTO 100 !TRAP FALSE REPLY
140 IF I$ = "X" THEN GOTO 999
150 I = ASC(I$) - ASC(A) + 1 !MUST USE UPPERCASE
160 IF I < 1 GOTO 100
170 ON I GOSUB 1000,2000
180 GOTO 100
```

If the alphabetic code are not sequential, then string commands can help;

```
...
130 INPUT I$
140 I = POS("ABLMNQRSZ",I$,0)
150 IF I = 0 THEN GOTO 100
160 ON I GOSUB 100,000,300,400,500,600,700,800,900
```

Also, key words can be picked out, and the whole key word need not be input; (note line 40 is used for alignment and would be deleted after the MENU\$ string was set up)

```
40 REM 1234567890123456789012345678901234
50 MENU$= " ONE TWO THREE FOUR ONCE EXIT"
100 REM USE A BLIND MENU
110 INPUT I$
120 I$= " " & I$ !ADD A SPACE IN FRONT
130 I= POS(MENU$,I$,1) !FIND FIRST MATCH
140 I2= POS(MENU$,I$,I+1)!MULTI OCCURANCE?
150 IF I2=0 THEN 190
160 PRINT " MISMATCHED KEY WORD"
170 PRINT MENU$
180 GOTO 110
190 ON (8+I)/6 GOSUB 210,1100,1200,1300,1400
,2000,999
200 GOTO 110
210 RETURN
999 STOP
1100 PRINT "=1":RETURN
1200 PRINT "=2":RETURN
1300 PRINT "=3":RETURN
1400 PRINT "=4":RETURN
2000 PRINT "ONCE NOT ONE":RETURN
```

Note that "F" alone will match "FOUR" but at least three characters are required to match "ONE" vs. "ONCE". This was set up for key words not longer than 5 characters. All key words in the list must be padded with extra characters so that they are the same length. Alternatively you could make an auxiliary table with the correspondence between position in the table and the subroutine number. Unless you convert the lower case to upper case, you will need to duplicate the key words in lower case. The ON command argument rounds to the integer used in the selection of the GOSUB, so the "8" and "6" in line 190 will need to be changed if the key word length changes.

Yet another way to handle the key word search is to count the number of separator characters;

```
50 MENU$= " ONE TWO THREE FOUR ETC"
100 PRINT MENU$
110 INPUT I$
120 I$=" "&I$
130 I=POS(MENU$,I$,1)
140 IF I=0 THEN GOTO 100
150 J=POS(MENU$,I$,I+1)
160 IF J=0 THEN GOTO 200
170 PRINT "MISMATCHED KEY WORD"
180 GOTO 100

200 J=1
210 FOR K=1 TO 10
220 J=POS(MENU$," ",J)
230 IF J=I THEN GOTO 280
240 J=J+1
250 NEXT K
260 ON K GOSUB 1100,1200,1300,1400,2000,999
270 GOTO 100
999 STOP
```

With video terminals one can use "Soft Keys". In fact the replaceable strip of commands at the top of the keyboard are a form of soft key. It is "soft" because each key's meaning changes with the program. The keys can be made even more dynamic by putting a definition menu on the screen which can change as often as needed within the program.

```
10 REM SOFT KEY
100 CALL CLEAR
110 DISPLAY AT(22,1):"1 2 3 4 5 6 7 8
9 0"
120 DISPLAY AT(23,1):"ONE THREE ONCE"
130 DISPLAY AT(24,1):" TWO FOUR EXIT"
140 CALL KEY(0,I,S)
150 IF S=0 THEN GOTO 140
160 IF I<49 OR I>57 THEN GOTO 140
170 ON I-48 GOSUB 1100,1200,1300,1400,2000,99
9,200,200
180 GOTO 140
200 RETURN !NULL FOR UNUSED KEYS
999 STOP
1100 DISPLAY AT(15,1):"=1":RETURN
1200 DISPLAY AT(15,1):"=2":RETURN
1300 DISPLAY AT(15,1):"=3":RETURN
1400 DISPLAY AT(15,1):"=4":RETURN
```

You can use the SHIFT and FCIN keys, but the numeric sequence is less convenient. The ASCII equivalents are listed below;

KEY	NORMAL	SHIFT	FCIN
1	49	33	3
2	50	64	4
3	51	35	7
4	52	36	STOP
5	53	37	14
6	54	94	12
7	55	38	1
8	56	42	6
9	57	40	15
0	48	41	188
+/=	61	43	ALL GONE!@#!

DELAWARE VALLEY USERS GROUP: NOV. 1986

ASSEMBLY CROSS REFERENCE USING DEBUG MODE  
by Norm Sellers

Three months ago, I published what I call the DEBUG MODE method of writing an assembly program which starts with a SKELETON program and three 'COPY' files. Code for a particular new program is simply added to this skeleton in the form of a main routine followed by 'BL' routines consisting of special beginning and ending statements.

I wanted to give you a sample of the DEBUG MODE in action so I decided to write a simple CROSS REFERENCE program using DEBUG MODE. It prints a cross reference listing of the variables used in a portion of error-free assembly source code using the DEBUG MODE. The program got somewhat large even when half of it is written in E/B. This program was written using "structured" programming techniques and consists almost entirely of code equivalent to "structured" code where the things like "DO WHILE" are simulated by TISS Jumps. Routines are short and easy to understand and there is NEVER any branching from one routine into the middle of another routine. YUK! The DEBUG MODE does not check for this and it could create a monstrous BUG. Anyway, anyone doing it should loose computer privileges for a month.

PROGRAM DESCRIPTION:

This program consists of an E/B driver program and a three entry ("CLR","ADD","GET") assembly subroutine. The E/B portion handles all disk and printer I/O and decides which program portion of the source program to cross reference ("ADD"). The E/B portion first clears the tables ("CLR"). It allows starting at a particular line number since the cross reference tables only handle 146 entries. For large programs, the BASIC driver automatically sections the cross reference listing when the tables become filled until the complete listing is printed. The last line processed when the tables got filled may be partially processed so it is processed the next time also.

The E/B program reads a line and counts it in an E/B counter. The assembly subroutine must also be given the line number with every record 'ADD' added. If the statement opcode is 'COPY', the E/B program opens the second file and starts reading and processing it the same way the assembler does.

When all of the program source to be cross referenced has been read, or when the tables are full, a loop of CALL LINK("GET",RECS) is executed. Note "GET" expects 1 argument while "ADD" expects 2 arguments. These RECS are printed (you may have to change line 160) as the sorted cross reference listing. When there are no more lines to 'GET', LEN(RECS)=0. This condition signals to end the 'GET' loop. If the full condition did not start this print, the E/B program is complete, otherwise the next portion of source is processed.

The assembly subroutine consists of many counters and indexes. Counters start at one while indexes start at zero. The comments try to show the type of each variable with 'IDX' or 'CNT' placed on the respective lines. The subroutine is absolutely loaded starting at >24F4 so the program listing explicitly shows the address of everything. This is handy when using the DUMP program published 2 months ago.

You may Cross Reference a program, then DUMP some of the DEBUG MODE tables (ie trace tables or counts of 'BL' routine executions within the cross reference subroutine) and some cross reference tables (ie TBL, variable names) using my DUMP

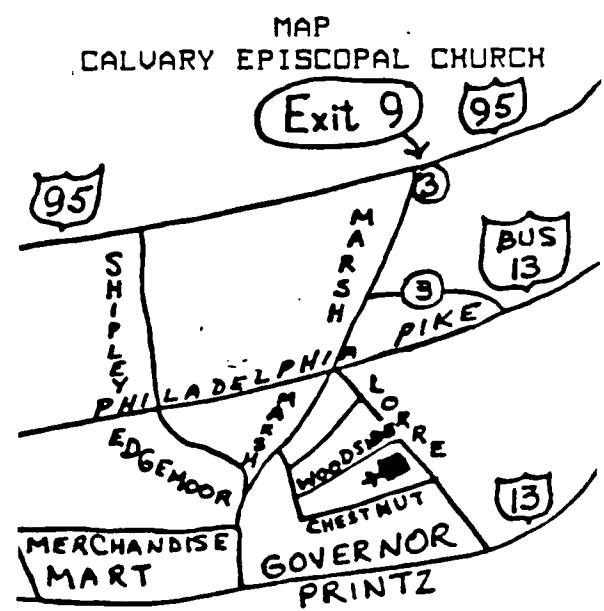
program. Try introducing a "BUG" someplace with CALL LOAD then run and DUMP the tables. This study will give you experience at determining from the DEBUG tables where execution stopped.

Then go forth as expert assembly programmers writing terrific programs we can all enjoy.

SEE ATTACHED PROGRAM LISTING

1984															
JANUARY							FEBRUARY								
S	M	T	W	T	F	S	S	M	T	W	T	F	S		
			1	2	3	4							1		
5	6	7	8	9	10	11	2	3	4	5	6	7	8		
12	13	14	15	16	17	18	9	10	11	12	13	14	15		
19	20	21	22	23	24	25	16	17	18	19	20	21	22		
26	27	28	29	30	31		23	24	25	26	27	28			
MARCH							APRIL								
S	M	T	W	T	F	S	S	M	T	W	T	F	S		
						1				1	2	3	4	5	
2	3	4	5	6	7	8	6	7	8	9	10	11	12		
9	10	11	12	13	14	15	13	14	15	16	17	18	19		
16	17	18	19	20	21	22	20	21	22	23	24	25	26		
23	24	25	26	27	28	29	27	28	29	30					
30	31														
MAY							JUNE								
S	M	T	W	T	F	S	S	M	T	W	T	F	S		
				1	2	3			1	2	3	4	5	6	7
4	5	6	7	8	9	10	8	9	10	11	12	13	14		
11	12	13	14	15	16	17	15	16	17	18	19	20	21		
18	19	20	21	22	23	24	22	23	24	25	26	27	28		
25	26	27	28	29	30	31	29	30							
JULY							AUGUST								
S	M	T	W	T	F	S	S	M	T	W	T	F	S		
			1	2	3	4	5							1	2
6	7	8	9	10	11	12	3	4	5	6	7	8	9		
13	14	15	16	17	18	19	10	11	12	13	14	15	16		
20	21	22	23	24	25	26	17	18	19	20	21	22	23		
27	28	29	30	31			24	25	26	27	28	29	30		
							31								
SEPTEMBER							OCTOBER								
S	M	T	W	T	F	S	S	M	T	W	T	F	S		
			1	2	3	4	5	6				1	2	3	4
7	8	9	10	11	12	13	5	6	7	8	9	10	11		
14	15	16	17	18	19	20	12	13	14	15	16	17	18		
21	22	23	24	25	26	27	19	20	21	22	23	24	25		
28	29	30					26	27	28	29	30	31			
NOVEMBER							DECEMBER								
S	M	T	W	T	F	S	S	M	T	W	T	F	S		
						1				1	2	3	4	5	6
2	3	4	5	6	7	8	7	8	9	10	11	12	13		
9	10	11	12	13	14	15	14	15	16	17	18	19	20		
16	17	18	19	20	21	22	21	22	23	24	25	26	27		
23	24	25	26	27	28	29	28	29	30	31					
30															

NOTE CHANGED CHRISTIANA MEETING DATES



It all began on August 17, 1985 with a unique new graphics package that startled the 99/4A world with it's fresh approach, and was hailed by MICRO-pendium as the best value... ever. Thus was born GRAPHX Companion.

On June 1, 1986 we again introduced a package that all the "experts" said would never make it. GRAPHX Pictures is still turning heads and showing everyone that graphics can be useful as well as beautiful.

Now introducing something so fresh and new, that you would call it simply Asgard Software's logical next step...

## GRAPHX Companion III

This package is much more than just more of the same old thing. Don't even consider the fact that it contains the largest collection of clipart to date, or that it contains a vast library of all new fonts (most of which include complete upper and lower case with numbers and symbols). All you have to remember is that it is from Asgard Software, which means (as always) that it's software with a difference.

With this package you can now make pictures that border on the magnificent with our set of imaginative borders, or you and your computer can make beautiful music together with our music symbol library, and you can even play with moving pictures with our new animation sequence. The possibilities are endless as GRAPHX Companion III will help you explore the limits of TI-99/4A graphics, and GRAPHX.

If you have GRAPHX, you will not want to be without this latest addition to our growing collection of graphics oriented software. At only \$9.00 a copy, how can you not afford to be without it?

*Asgard Software*

*Box 10306*

*Rockville, MD 20850*



For information or to order call (301)345-2492 (8AM - 8PM E.S.T.)

DELAWARE VALLEY USERS GROUP: NOV. 1986

```

TITL 'XREF for Assembly SOURCE'
* DSK2.AXREF
AORG >24F4
.....
      CROSS REFERENCE
      TABLE FOR AN ASSEMBLY
      PROGRAM SOURCE
.....
      PROGRAM
      WRITTEN BY
      NORM SELLERS
.....
      DATE ASSEMBLED:
TEXT   '10/30/86'
.....
      VERSION
TEXT   '1.0'
.....
      USES   DSK2.AXREF   ALL COPIES
      SKELA   SKELA       MAIN VARS
.....
      DSK2.AXREFB   SPEC ROUT
* DISK LABEL   INPT RINS
* STRUCTURE   DSK2.OXREF   OBJECT
*             OSK1.AXREFLST LISTING
.....
DEF   ADD,GET,CLR
COPY "OSK2.SKELA1"

```

```

OTHER DATA DEFINITIONS FOLLOW
.....
BLCNIS DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
BLCEND EQU $
BARG# EQU 0      * BARG
RLAB# EQU 2      * RLAB
GARG# EQU 4      * GARGS
RCLR# EQU 6      * RCLR
CTAGI# EQU 8     * CTAGI
CTAGJ# EQU 10    * CTAGJ
* # EQU 12
ISTNW# EQU 14    * ISTNW
ADISU# EQU 16    * ADISU
SRCH# EQU 18     * SRCH
RGET# EQU 20     * RGET
PINDS# EQU 22    * PINDS
PNXT# EQU 24     * PNXT
LBLK# EQU 26     * LBLK
PRST# EQU 28     * PRST
SOPT# EQU 30     * SOPT
ADNW# EQU 32     * ADNW
HX2DC# EQU 34    * HX2DC
TOPLN# EQU 36    * TOPLN
CLENT# EQU 38    * CLENT
PARG# EQU 40     * PARG
LDUEN# EQU 42    * LDUENT
PFULM# EQU 44    * PFULM
FNDAR# EQU 46    * FNDAR
INTFT# EQU 48    * INTFTN

```

```

***** VARIABLE DATA ALLOCATIONS
.....
FULL DATA 0      * FULL TBL INDICATOR
DROP DATA 0      * -1 IF DROP UENTRY
IM DATA 0        * SAUD ENTRIES
TSU DATA 0       * INDEX TO SAVE
I DATA 0         * RECDRD NUMBER
AGN DATA 0       * -1 :AI ", "
SIGNF DATA 0    * -1 IF HIT<>0
*                * ELIM LEADING ZEROS
SUR6 DATA 0     * SAVE IDX FOR LNS
SUR8 DATA 0     * SAVE TAG(I)
.....
PRIRST BSS 1     * 1-PRT REST
CLDRTN BYTE 0    * 1-ADD,2-GET,3-CLR
LENT BSS 1       * CNT LNTH OF ENT

```

```

ENT BSS 18      * ENTRY FROM SOURCE
BSS 1
LUENT BSS 1     * CNT LNTH OF UENT
UENT BSS 6      * SHORTEND ENTRY FOR XREF
PENT BSS 6      * PREV PRINTED ENTRY
FND DATA 0     * 1-FND,0-NOT FND IN SRCH
SRID DATA 0    * 1-SORTED.
LPLINE BSS 1    * LENG OF PRINT LINE
PLINE BSS 73    * PRINT LINE
LNSIND DATA 0  * LINE # CNT TO PLINE
TSVIND DATA 0  * WHICH TBL ENTRY TO PRT
WORK DATA 0
CHRNUM DATA 0,0
OIUWRK DATA 0,0
MLNS BSS 146    *
*
TAGS BSS 146    *
*
LREC BSS 1      *
REC BSS 255     *
LNS BSS 2920    *
TBL BSS 876     *
ENDDAT EQU $
NUMREF EQU >200C
STRASG EQU >2010
STRREF EQU >2014
FAC EQU >834A
MSG2 TEXT 'FULL, STOPPED AFTER LINE-'
PLNO TEXT '
*
CONSTANTS
BLK DATA >2020
COM TEXT ','
LPAR TEXT '('
RPAR TEXT ')'
AI TEXT ''
AST TEXT '* '
GT TEXT '>'
PLS TEXT '+'
MNS TEXT '-'
ACHR TEXT 'A'
SQ BYTE >27    * SINGLE QUOTE
MLEN DATA 6   * SZE OF TABLS
TMAX DATA 146
MONE DATA -1
TEN DATA 10
THREE DATA 3
FOUR DATA 4
D100 DATA 100 * IN INTFTN
AMLNS DATA 10 * MAX REFS/ENT
UMBW EQU >2024
STATUS EQU >837C
KSCAN EQU >201C
PAGE
.....
* MAIN PROGRAM *
.....
ADD MOVB @ONE+1,@CLDRTN
JMP $+16
GET MOVB @TWO+1,@CLDRTN
JMP $+8
CLR MOVB @THREE+1,@CLDRTN
COPY "DSK2.SKELA2"
CB @CLDRTN,@TWO+1
JEQ EGET
CB @CLDRTN,@THREE+1
JEQ ECLR
.....
* MAIN ROUTINE CODE FOLLOWS
*
EADD LI RIN,BARG * GET BASIC
BL @BLROUT * ARGUMENTS
MOV @FULL,RO * IF TABLES FULL
JNE EADD1 * EXIT
LI RIN,RLAB * HANDLE
BL @BLROUT * LABELS
MOV @FULL,RO * IF TABLES FULL
JNE EADD1 * EXIT

```

```

LI   RTN,FNDARG      • FIND FIRST
BL   @BLROUT        • SOURCE ARG
C    R12,R10         • IF COMPLETE SIMT
JHE  EADD2          • SCANED, RETURN
LI   RTN,GARGS      • HANDLE
BL   @BLROUT        • ALL ARGS
MOV  @FULL,RO       • IF TABLES FULL
JNE  EADD1          • EXIT
JMP  EADD2
EADD1 LI R3,2        • STRING IN ARG2
LI   RTN,PARG       • FULL MSG
BL   @BLROUT        • TO BASIC
EADD2 B @RETSYS
ECLR LI RTN,RCLR    • CLEAR MEMORY FOR
BL   @BLROUT        • NEXT XREF
B    @RETSYS
EGET LI RTN,RGET    • GET NEXT LINE TO
BL   @BLROUT        • PRINT
LI   R3,1           • STRING IN ARG1
LI   RTN,PARG       • PUT PLINE
BL   @BLROUT        • IN ARG TO BASIC
B    @RETSYS        • RETURN TO SYSTEM

```

COPY "DSK2.SKELA3"

```

• GET BASIC ARGS • R(3)
BARG DATA BLCNTS+BARG# ADDR OF CNT EXECUTES
CLR RO
LI R1,1
BLWP @NUMREF      • GET LN CNT IN BAS
LI RTN,INTFIN    • CONVRT LINE CNT
BL @BLROUT        • TO WORD INTEGER
CLR RO
LI R1,2          • FOR 2ND BASIC ARG
LI R2,@LREC      • MAX LEN=255
LI R3,>FF00
MDUB R3,@LREC
BLWP @STRREF     • GET SOURCE LINE
B @BLRETN        • FROM BASIC.

```

```

• CLEAR MEMORY • R()
• FOR NEXT XREF •
RCLR DATA BLCNTS+RCLR# ADDR OF CNT EXECUTES
LI R1,TBL-BLCNTS
SRA R1,1
LI R2,BLCNTS
CLR *R2+
DEC R1
JGT $-4
LI R1,ENDDAT-TBL
DECT R1
LI R2,TBL
MOV @BLK,@TBL(R1)
DECT R1
JLT $+4
JMP $-10
LI R1,12
LI R2,MYREG
CLR *R2+
DEC R1
JGT $-4
B @BLRETN

```

```

• HANDLE LABELS • R(C)
RLAB DATA BLCNTS+RLAB# ADDR OF CNT EXECUTES
CLR R12
LI RTN,CLENT      CLEAR ENTRIES
BL @BLROUT

```

```

MOVB @LREC,RO
SRA RO,8
RLAB1 C R12,RO    CMPR IDX TO CNT
JHE RLAB2
CB @REC(R12),@BLK
JEQ RLAB2
MOVB @REC(R12),@CNT(R12)
INC R12
JMP RLAB1
RLAB2 MOV R12,R12
JEQ RLAB4
SLA R12,B
MOVB R12,@LENT CNT
SRA R12,8
LI RTN,TSINW      • TEST FOR NEW ENTRY
BL @BLROUT
RLAB4 B @BLRETN

```

```

• FIND FIRST ARG • R(AC)
• WITH R12.

```

```

FNDARG DATA BLCNTS+FNDARG# ADDR OF CNT EXECUTES
MOVB @LREC,R10
SRA R10,8
INC R12
C R12,R10
JHE FNDA1
CB @REC(R12),@BLK • GET TO
JEQ $-12          • OPCODE
INC R12
C R12,R10
JHE FNDA1
CB @REC(R12),@BLK • PASS
JNE $-12         • DPCODE
INC R12
C R12,R10
JHE FNDA1
CB @REC(R12),@BLK • GET TO
JEQ $-12        • FIRST ARG
FNDA1 B @BLRETN

```

```

• HANDLE THE ARGS • R(3AC)

```

```

GARGS DATA BLCNTS+GARGS# ADDR OF CNT EXECUTES
C R10,R12
JLT GARG6
GARG1 LI RTN,CLENT • CLEAR ENTRIES
BL @BLROUT
CLR @AGN
CLR R3
CB @REC(R12),@SQ • DROP IF
JEQ GARG6        • CHAR STRING
GARG2 C R12,R10    CMPR IDX TO CNT
JHE GARG4
CB @REC(R12),@BLK • BLANK
JEQ GARG4
CB @REC(R12),@COM • COMMA
JEQ GARG3
CB @REC(R12),@PLS • PLUS
JEQ GARG3
CB @REC(R12),@MNS • MINUS
JEQ GARG3
CB @REC(R12),@LPR • LEFT PAREN
JEQ GARG3
MOVB @REC(R12),@CNT(R3)
AB @ONE+1,@LENT CNT
INC R12
INC R3
JMP GARG2
GARG3 MDV @ONE,@AGN • LOOK AFTER COMMA
GARG4 MOVB @LENT,RO
JEQ GARG5
LI RTN,TSINW      • TEST FOR NEW ENTRY
BL @BLROUT        • (HAVE INDEX REG?)
GARG5 MDV @FULL,RO • EXIT IF FULL

```



DELAWARE VALLEY USERS GROUP: NOV. 1986

```

JNE GARG6
C @AGN,@ONE
JNE GARG6
INC R12
C R12,R10
JL GARG1
GARG6 B @BLRETN
*
* CLEAR ENTRY ENT * R()
* AND UENT *
*
CLENT DATA BLCNTS+CLENT# ADDR OF CNT EXECUTES
MOVB @ZERO,@LENT
MOVB @ZERO,@LUENT
LI RO,UENT
LI R1,ENT
LI R2,S
CLR *RO+
CLR *R1+
DEC R2
JGT S-6
B @BLRETN
*
* TEST FOR NEW ENTRY * R()
*
TSINW DATA BLCNTS+TSINW# ADDR OF CNT EXECUTES
MOV @FULL,RO
JNE TSIN4
LI RTN,LDUENT * LD USE ENT (SHORT)
BL @BLROUT
MOV @DROP,RO
JNE TSIN4
MOV @TM,RO CNT
JGT TSIN2
LI RTN,ADNW * ADD NEW ENTRY
BL @BLROUT
LI RTN,ADTSU * ADD AT TSU
BL @BLROUT
JMP TSIN4
TSIN2 LI RTN,SRCH * SEARCH FOR UENT
BL @BLROUT
MOV @FND,RO
JEQ TSIN3
LI RTN,ADTSU FOUND, ADD AT TSU
BL @BLROUT
JMP TSIN4
TSIN3 C @TM,@TMAX
JHE TSIN5
LI RTN,ADNW NOT FOUND, ADD NEW
BL @BLROUT
LI RTN,ADTSU * ADD AT TSU
BL @BLROUT
TSIN4 B @BLRETN
TSIN5 LI RTN,PFULM * FULL MSG
BL @BLROUT * TO PLINE
JMP TSIN4
*
* FULL MSG AND LAST LINE * R()
* NUMBER TO PLINE *
*
PFULM DATA BLCNTS+PFULM# ADDR OF CNT EXECUTES
MOV @ONE,@FULL * SIGNAL FULL TABLES
INC @I * TO CNT FOR FULL MSG
LI RTN,HX2DC * LAST LINE #
BL @BLROUT * TO MSG2 LINE
DATA 3,I,PLND,4
LI RO,>1000
MOVB RO,@LPLINE
SRA RO,B
LI R1,MSG2 * MOVE MSG2
LI R2,PLINE * TO PLINE
MOVB *R1+,*R2+
DEC RO

```

```

JGT S-4
B @BLRETN
*
* ADD NEW TBL ENTRY * R(567)
*
ADNW DATA BLCNTS+ADNW# ADDR OF CNT EXECUTES
MOV @FULL,RO IF TABLES FULL
JNE ADNW2 EXIT
MOVB @LUENT,R7 CNT
SRA R7,B
DEC R7 CNT TO IDX
INC @TM NEW CNT
C @TM,@TMAX
JGT ADNW3
MOV @TM,R5 CNT
MOV R5,@TSU CNT
DEC R5 CNT TO IDX
MPY @MLN,R5 R6=MLN*R5, R5=0
ADNW1 MOVB @UENT(R5),@TBL(R6) MOVE UENT
INC R6 TO TBL
INC R5
C R5,R7 IDX TO IDX
JLE ADNW1
ADNW2 B @BLRETN
ADNW3 LI RTN,PFULM
BL @BLROUT
JMP ADNW2
*
* ADD AT TSU INDEX * R(567)
*
ADTSU DATA BLCNTS+ADTSU# ADDR OF CNT EXECUTES
MOV @FULL,RO IF TABLES FULL
JNE ADTS2 EXIT
MOV @TSU,R5 CNT
DEC R5 CNT TO IDX
MOVB @MLNS(R5),R7 CNT
AB @ONE+1,R7 UP BY 1
CB R7,@AMLS+1 CMPR CNT TO CNT
JLE ADTS1
LI RTN,ADNW * ADD NEW ENTRY
BL @BLROUT
JMP ADTSU+2
ADTS1 MOVB R7,@MLNS(R5) SAVE NEW CNT
SRA R7,B
MPY @AMLS,R5 R6=AMLS*R5
A R7,R6 R6=AMLS*R5+R7
DEC R6 IDX
SLA R6,1 (AMLS*R5+R7)*2
MOV @I,@LNS(R6) IDX
INC @LNS(R6) IDX TO CNT
ADTS2 B @BLRETN
*
* LOAD USE ENT (UENT)* R(3)
*
LDUENT DATA BLCNTS+LDUENT# ADDR OF CNT EXECUTES
MOVB @LENT,RO CNT
SRA RO,B
CLR R1
LI R2,ENT
LI R3,UENT
CB *R2,@AT * DROP '@'
JEQ LDUE1 * DROP '*'
CB *R2,@AST * DROP '*'
JEQ LDUE2 * KEEP ALPHA
CB *R2,@ACHR
JL LDUE4
JMP LDUE3
LDUE1 INC R2
DEC RO
CB *R2,@GT * DROP '>'
JEQ LDUE4
JMP LDUE3
LDUE2 INC R2
DEC RO

```

