HOOSIER USERS GROUP HOOSIER USERS GROUP HOOSIER USERS GROUP HOOSIER USERS GROUP HOOSIER USERS GROUP
USERS GROUP HOOSIER USERS GROUP HOOSIER USERS GROUP HOOSIER USERS GROUP HOOSIER USERS GROUP HOOSIER
GROUP HOOSIER USERS GROUP HOOSIER USERS
HOOSIER USERS GROUP HOOSIER USERS GROUP
USERS GROUP HOOSIER USERS GROUP HOOSIER
GROUP HOOSIER USERS GROUP HOOSIER USERS
HOOSIER USERS GROUP HOOSIER USERS GROUP
USERS GROUP HOOSIER USERS GROUP HOOSIER USERS GROUP HOOSIER USERS GROUP HOOSIER USERS GROUP HOOSIER
GROUP HOOSIER USERS GROUP HOOSIER USERS GROUP HOOSIER USERS GROUP HOOSIER USERS GROUP HOOSIER USERS

THE HUGgers
HOOSIER USERS GROUP
People Helping People

May 1987 THE HUGger NEWSLETTER Volume 5, Number 2

OFFICER'S CORNER

The next HUG meeting is Sunday, May 17, 1987 at St. Ann's School. Map on the next to last page of each news letter shows the location of St. Ann's school.

HUG members in attendence at the annual April meeting elected Steve Sims, President and Treasurer (two offices). I was re-elected Vice-President. Dan Eicher was elected Secretary. The election comittee reported no absentee ballots were requested.

Steve Sims announced that the TI System that has been available for use to copy HUG library disks, will no longer be available. Early in the HUG history, the HUGger bbs was shut down for use at meetings. Steve has carried more than his load of reponsibities to the HUG. My TI System may or may not be @the HUG meetings. **Would someone volunteer** to bring a dual drive system for HUG library use?

St. Ann's School has TI 99 4/A consoles that are used in grade 1 thur grade 6, St. Ann's has allowed us to meet without a rental charge since Oct. 1986. St. Ann's School has a need for educational modules, HUG members interested in donatating modules please contact one of the HUG officers.

Members in attendence at the April meeting received their HUG newsletter, "expressed delivered" after lablels were attached and non-attending members newsletters had stamps attached for mailing. Will this become the **NEW** delivery schedule? If you have an item for the newsletter, it would be most appreciated to have the item no later than the weekend after the meeting.

HUG Constitution defines duties and responsiebities of the officers, and includes the bylaws of HUG. Dan Eicher has requested the office of Secretary be renamed Imformation Officer. Some items not covered in the Constitution are the HUGger newsletter, the HUGger bbs, and HUG assets. How many members are aware of the equipment owned by the HUG, the manufacture, the serial number, the equipment costs, and the date of aquistion, also the same of software that belongs to the HUG? We should have a written record of all HUG assets, in case of fire, theft, or whatever else could cause a loss.

Reprinted are the **1987 HUG MEETING** dates, please mark your calendars and plan to attend **HUG MEETINGS!**

William M. Lucid Vice-President

MEETING DATES

1987 Hoosier User's Group meeting dates

***May 17**
June 14
***July 19**
August 9
September 13
October 11
November 8
December 13

*** Indicates meeting will be third Sunday instead of the usual second Sunday of the month.**

SOUTH REGIONAL MEETING

The next South Regional meeting will be held on Wednesday, . For additional information call 881-5918.

PILOT/99 LANGUAGE REFERENCE SUMMARY

by
Sid Smart
MICRO/99 Users Group, Bloomington, IL

LANGUAGE ELEMENTS

Statement form:

```
    op-code[modifier]: operands
  where:
  - op-code is 1 or 2 characters telling what the statement does,
    e.g., CS for Compute String, DC for Draw Circle.
  - modifier is an optional true or false logical expression,
    e.g., Y for yes, N for no, (#A>0).
  - operands are data required by the op-code,
    e.g., sprite numbers, variable names, literals.
```

User variables:

```
    26 numeric variables,      #A thru #Z
    13 character variables,    $A thru $M
```

Answer buffer:

A system variable filled by Accept, Accept Single, or REad, and which is examined by any form of the Match statement.

Yes flag:

A system true/false variable set by any form of the Match statement or by Fire Button or Sprites Atouch statements. Modified statements will be executed only if the modifier matches the yes flag.

Labels:

*name where name is 1 to 10 upper case alphabetic characters.
@A targets the most recently executed Accept statement.
@M targets the next Match statement.
@P targets the next PRoblem statement.
A label occupies a line by itself, and is a target for jumps.

STATEMENT FORMATS

Primary statements:

A: [variable]	Accept input data from keyboard
AS: [variable]	Accept Single character from keyboard
C: #A <- expression	Compute numeric variable
CH:	Clear screen, Home cursor
CS: $A <- expression	Compute String
E:	End program or subroutine
J: label	Jump
JM: label[,label,...]	Jump on Match to corresponding label
M: string[,string,...]	Match answer buffer to strings
MJ: string[,string,...]	Match or Jump to next match statement
PR:	PRoblem starting point
R: remark	Remark
T: data	Type to the screen
TH: data	Type and Hang cursor
TP: data	Type to Printer
U: label	User subroutine invocation

Character graphics statements:

CC: charset,color,color	Character Color
CP: code,pattern	Character Pattern
HC: row,col,code,repeat	Horizontal Character output to screen
IT:	Initialize Text mode
SN: color	ScreeN color
TC: row,col	Text Cursor positioning
VC: row,col,code,repeat	Vertical Character output to screen

Sprite control statements:

GP: sprite,pattern	Graphic Pattern
SA:	Sprites Atouch
SC: sprite,color	Sprite Color
SD: sprite	Sprite Delete
SG:	Sprites Gone
SH: sprite,sprite	Sprite Hit
SL: sprite,row,col	Sprite Location
SM: sprite,row-vel,col-vel	Sprite Motion
SP: sprite,code	Sprite Pattern
SS: size	Sprite Size (1-4)

Bit map graphics statements:

DC: row,col,radius	Draw Circle
DL: row1,col1,row2,col2	Draw Line
DR: row1,col1,row2,col2	Draw Rectangle
GC: color,color	Graphic Color
IG:	Initialize Graphics mode
PP: row,col	Plot Point
TG: row,col,characters	Type in Graphic mode
UP: row,col	Unplot Point

File control statements:

CF: filename	Close File
OF: filename	Open File
RE: [variable]	REad file
RF: [rec#]	Restore File
WA:	Write Answer buffer to file
WR: data	WRite data to file

Miscellaneous statements:

LP: count	LooP
EL:	End Loop
BW:	Begin While control loop
WH: expression	WHile
JS: #,x,y	Joy Stick
FB: #	Fire Button
S: duration,freq,vol,voice	Sound

PILOT -

Here are three short Demo progerams created by Bill Harms and myself. They are meant to demonstrate what the various commands do, and some suggestions as to how to apply them. Chick

```
R: ON DISK AS: BITMAP
*START                                 ! Actually a User Subprogram
IG:                                    ! Initialize Graphics
T: HERE'S A CIRCLE AT 100,100 WITH A RADIUS OF 10
C: #N<-10                              ! gives #N the value of 10
LP: 8                                  ! starts a LooP of 8
DC: 100,100,#N                         ! DrawsCircle size 10
C: #N<-#N-1                            ! " #N=#N-1 "
EL:                                    ! End of Loop
T: NOW FOR A TRIANGLE
T: LINE FROM 10,30 TO 100,30 THEN TO 100,200 AND BACK
DL: 10,30,100,30                       ! DL: = DrawLine
DL: 100,30,100,200
DL: 100,200,10,30
T: NOW FOR A RECTANGLE. 5,130 TO 90,250
DR: 5,130,90,250                       ! DR: = DrawRectangle
DR: 7,137,95,252
LP: 100                                ! (these 2 lines create a
EL:                                       delay )
IT:                                    ! Initilize Text
*REENTER                               ! A 2nd User subprogram
T: PRESS >A< FOR AGAIN
T: PRESS >S< TO STOP
AS:                                    ! asks for a Single Answer
MJ: A                                  ! if "A" is Matched ...
J: *START                              ! Jump to subprogram *START
MJ: S                                  ! if "S" is Matched ...
T: Well its been fun. Bye              ! Adios
E:                                     ! END
J: *REENTER                            ! take care of any other key
```

NOTE: To keep final message visable long enough to read add A: just before the last "E:". (A: = input)

* * * * * * * * * * * * * * * * * * *

```
R: ON DISK AS: P46AVGS
*START
CH:
T: This program will find the average of entered numbers.
T: Enter "9999" to stop entering numbers.
C: #S<-0
C: #N<-0
*MORE
T: NEXT NUMBER PLEASE
T: (Don't just hit enter)
```

[Continued next page]

```
A: #A
M:
JY: #A
M: 9999
JY: *AVERAGE
C: #S<-#S+#A
C: #N<-#N+1
J: *MORE
*AVERAGE
C: #A<-#S/#N
T: The average of #N
T: entries is: #A
T: Press M for More!
T: Press any key but Enter to Stop
AS:
M: M
JY: *START
```

☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆

```
R: *** SCREEN LOCATIONS in the          ! Same as REM
R: *   graphic mode, using DC:
R: *
IG:                                     ! Initialize Graphocs
GC: 7,8                                 ! color Graphics RED on CYAN
T: at  30,40 -  30,120 -  30,200        ! TYPE 1st line...
DC: 30,40,10                            ! DrawCircle..Top left of screen
DC: 30,120,10                           ! Another...Top center of screen
DC: 30,200,10                           ! And one...Top  right of screen
A:                                      ! Used to wait for a key (ENTER)
GC: 13,8                                ! Graphics now GREEN on CYAN
T:     70,40 -  70,120 -  70,200        ! TYPE 2nd line...
DC: 70,40,10                            ! DrawCircles at center of screen
DC: 70,120,10
DC: 70,200,10
A:                                      ! Graphics to MAGENTA on CYAN
GC: 14,8                                ! Wait again
T:    110,40 - 110,120 - 110,200        ! TYPE 3rd line...
DC: 110,40,10                           ! DrawCircles near the bottom
DC: 110,120,10
DC: 110,200,10
T:                                      ! TYPE a blank line
T: Press <ENTER> to quit                ! TYPE "quit" instructions
A:                                      ! Wait for <ENTER> to be hit...
E:                                      ! and END program.
```

NOTE: At the "PROGRAM IS DONE" message, to run another program enter PILOT. If this doesn't work , enter -PILOT. (Because you are in FORTH you can also type MON to leave.

Chick

will be returned to your picture.

(7) Now use the joystick and get to the upper left-hand corner of your character. When you start to increase the size of the Instance box, the top line of the box should cross over the top-most pixel (s) of your character, same for the left-most pixel (s). Remember, whatever is under the box line will be included!

(8) The box will increase by 8 pixels at a time. Increase the box so that it covers the entire character. (If you find that the box also covers a part of another character, then go back and move it so it doesn't.) Use the least amount of space as possiable to cover the letter!

(9) When you have covered the entire letter, hit the fire button, the letter will automatically be saved under the entered name.

(10) REPEAT STEPS 6-9 until you have saved all the characters that you want to use in your font style!

****

ASSEMBLING THE FONT

(1) You will now start to assemble your font file, Use the EDIT SECTION OF TI-WRITER. Place the character of whatever one you are going to load in first, (probably "A") on line 0001, note the line you are at.

(2) Hit Function 9 (BACK), so that you now are back in the command mode of the editor.

(3) Type the LF command (Load File), then using this format below load the first character Instance:

```
              [xxx DSKx.A_I]
                 |       |
                 |       Instance file
   line number     you put the font  char
on
```

This will load the character instance into memory after line xxx. Using the LF command this way will allow you to load D/V 80 files (Instances!) without disturbing the data that you have in the editor already!

(4) After you have loaded in the instance at the end of your file (line xxx), you will need to add a number to the line that has only 2 numbers on it! It will look something like this:

2,3

After the second number in that line type a comma and the WIDTH number that you wrote down for the character you are working on. Say the number is 9, it will look like this:

2,3,9

(5) Go back to step (2) until you have finished loading and altering all your characters for the font style.

WARNING*WARNING: If you are doing it right, you are working WITHOUT carriage returns. YOU DON'T WANT THEM! So for Pete's sake or your own.. DON'T EVER HIT REFORMAT!

NOTE: It is a good ideal to ALWAYS include a space character in every font you do. To do this put a blank line at the end of your assembled file (don't erase this one!) and load any character Instance after that blank line. Take the width of the widest character and add it to the 2 number line (like we did before). Now change those other lines that loaded in to all 0's. Keep the same number of numbers, but change them to 0's.

(6) You will then go through your file and make sure that there are NO BLANK LINES (except the space character), or C/Rs at the end of lines. Also, double check that those lines that had 2 numbers now have 3!

(7) Now that you've double checked everything, hit Function 9 (BACK). Type the PF command. (Print File) Type the filename that you would like to call your font. Use this format:

```
DSKx.nnnnnn_F
```

(Remember, you do not want to SAVE FILE, you want to PRINT FILE to disk.) You now have created your very own font! Now go into the Enhancement part of TI-Artist and load your font, and see how good it looks, you may need to alter some of your characters. If everything doesn't look satisfactory, then continue to step 8.

(8) Go into the Enhancement section of TI-Artist, and load your font. Get all your characters onto the screen. There should be 1 pixel spacing between your characters touch the one on the left may have been defined incorrectly, when you saved it as an Instance.

(9) Go into the [S] Slides section and resave the character, making sure that the left side of the box goes over the left most pixel of the character.

(10) If your characters are not level, then you may need to see which characters are too high. Write down all the characters that need to be lowwered.

(11) Go back into the Enhancement, and re-save those characters as Instances, and make sure that you start 1 (or more) pixel higher than last time when you re-save it! Keep doing this until you've corrected all the faulty characters.

(12) After you have re-saved all the characters that were not right, go back to the Editor in TI-Writer. Load your Font file and Scan through the file until you find the character that you want to correct.

(13) Delete the definition lines below the 3 number line. Note the line that the 3 number line is at! Load your saved Instance using the same format as before:

```
xxxx DSKn.xxxxxxx_I
```

(14) Delete the line that has been loaded that only has 2 numbers on it.

You have now (hopefully!) corrected that character, if not, do it again. Lowwer case characters and symbols can cause you problems on centering, etc., so a little experience may be necessary to get things right, but a little common sense will prevail.

Learning to do this could open the door to a lot of possibilities... who says a font has to look like an alphabet?

(Editors note: Re-typed from NEWJUG NORTH March 1987, newsletter with credit to WEST NY 99ers newsletter.)

✡✡✡✡✡✡✡✡✡✡✡✡✡✡
✡✡✡✡✡✡✡✡✡✡✡✡✡✡

Hoosier User's Group has an offer on consignment a **PACKED** TI 99/4A deal, to sell, consisting of the following items. Terms of the consignment are a minimum system price has to be guaranteed before items will be sold seperately.

<u>HARDWARE</u>

2 TI-99/4A Consoles
1 TI P-Box (No Flex cable card)
1 TI P-Box Flex cable card (No P-Box)
1 TI RS232 (PHP1220)
1 Foundation 128K memory
1 Corcomp Disk Controller
1 TI Disk Controller (w/DM II)
1 TI P-Code card (TI p-code software)
2 Toshiba DS/DD half height disk drives
1 Speech Synthesizer
1 Wico Joystick adapter
1 Navarone cartridge expander
1 Direct Video/Audio cable
1 Cassette tape recorder cable
  (and several 5/10 minute tapes)
1 TI Replacement keyboard
1 TI Replacement power supply

<u>SOFTWARE</u>

2 TI Extended Basic cartridges
1 TI Writer cartridge & disk
1 TI Terminal Emulator II
1 TI Editor/Assembler
1 Microsoft Multiplan
1 SXB (Super Extended Basic) disks
1 TI Forth disk
1 Futura General Ledger disk
1 Programing Aids II disk
1 Programing Aids III disk
1 Engineering library disk

Send us an offer for item/items you are interested in to the Hoosier User's Group, P.O. Box 2222, Indianapolis, IN 46206-2222. Include a stamped self-addressed envelope for a reply.

Your offer maybe mailed via **GENIE** mail to SPEEDWAY.500 (HUGger Newsletter ED).

HOOSIER USERS GROUP DIRECTORY

OFFICERS

President........	Steve Sims	631-7255
Vice-President...	Bill Lucid	291-3995
Secretary........	Dan Eicher	787-4184
Treasurer........	Steve Sims	631-7255
Regional Center:		
South..........	Dennis Sherfy	881-5918

HUGGER NEWSLETTER EDITOR

Bill Lucid	(317)-291-3995
GENIE MAIL	SPEEDWAY.500

MONTHLV MEETING LOCATION

ST. Ann's School
2839 S. McClure
Indianapolis, IN

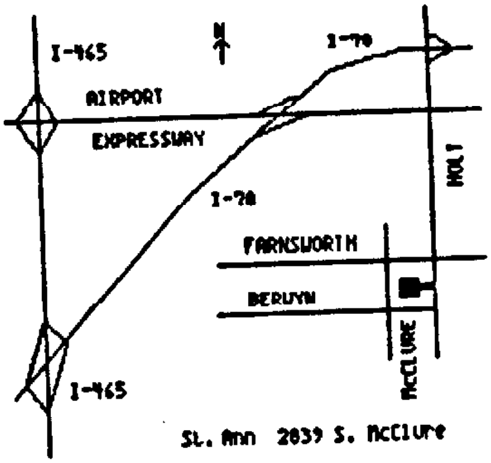
Meetings open at 2:00 PM

NEWSLETTER EXCHANGE

The HOOSIER USERS GROUP is participating in a Newsletter Exchange program with other TI Users Groups. This offer is made with the understanding that, with proper credit, your Users Group can reprint articles from the HOOSIER USERS GROUP Newsletter, and with proper credit, we can reprint articles from other TI Users Groups Newsletters.

PRINTOUTS

Library listings can be ordered for $.25 6 X 9 self addressed envelope with $.66 postage. The HUGbbs Reference Guide can be ordered for $.50 and a 4 X 9 self addressed envelope with $.22 postage. Please send orders to our P.O. Box. SORRY, PRINTOUTS WILL BE SENT TO ACTIVE MEMBERS ONLY.



St. Ann 2839 S. McClure

HUGbbs INFORMATION

317-631-994A 300 baud only
The HUGbbs is on-line 24 hours a day.

SPONSOR THE HUGbbs: Any member or retail business can sponsor the HUGbbs. For $5.00 donation, you get 5 (40 column) lines on the Log-on Title Screen for a week (or for a $10.00 donation, you get 10 (40 column) lines plus a 24 line by 40 character ad in the Sales option of the file module. To sponsor the HUGbbs, send a check or money order to our P.O. Box (or turn in at our Monthly meeting) specifying how many weeks (and how many lines) you want to sponsor, your name (or company name), address, phone, what you want to say, and the week (and an alternate week) you want the ad to appear.*

BACK ISSUES

Back Issues purchased at the monthly meeting are $1.00 each. Mail order price is $1.50 per Newsletter (postage included). Orders will be filled within 3 weeks of receipt.

ADVERTISING POLICIES

There will be no charge for advertisements submitted to the HUGger Newsletter by members (for private sale only). Format for the advertisements is 45 characters wide by 10 lines long. The Ad should be typed or hand printed exactly how it is to appear in the Newsletter. Deadline for an Ad to appear in next month's Newsletter is the 2nd Saturaday of the month.*

For companies who wish to advertise in the HUGger Newsletter, our rates are as follows:

Pre-Printed Inserts (one page):	**$20.00**
One Full Page (one sided) Ad:	**$25.00**
One Half Page Ad:	**$13.00**
One Quarter Page Ad:	**$ 7.00**

All Ads must be in ready to print condition. Advertisements must be in our P.O. Box before the 2nd Saturaday of the month to appear in the following month's Newsletter.*

*NOTE: The officers of the HOOSIER USERS GROUP reserve final approval on all advertizements submitted for the HUGger Newsletter and the HUGbbs. The officers and the Newsletter committee are not responsible for typographical errors due to illegiable advertisements. All proceeds are accepted as donation to the **HOOSIER USERS GROUP.**

HOOSIER USERS GROUP
P.O. Box 2222
Indianapolis, IN 46206-2222

Forwarding and Address
Correction Requested

TIME DATED
May 17, 1987
MATERIAL

88/03   733DE

Dan Eicher
4410 Cardinal Drive
Indianapolis, IN 46237

APPLICATION FOR MEMBERSHIP

Below you will find an application for membership to the Hoosier Users Group. Active membership entitles you to the Newsletter, up and download on the HUGbbs, attendance and voting rights at regular club meetings, access to the HUGger Library of Programs, special club activities and special guest speakers for one year. Subscribing members will receive the **NEWSLETTER** only.

Make check or money order payable to **Hoosier Users Group.** Send completed application to:

HOOSIER USERS GROUP
P.O. Box 2222
Indianapolis, IN 46206-2222

(Cut on dotted line)

Check One:

Active Member

New: $20 ______
Renewal: 15 ______

Subscribing Member

New: $10 ______
Renewal: 7.50 ____

Amount Enclosed: $ ______

________ D ________
S ________ O ________

Name:______________________________Today's Date:____________

Address: ____________________________________Apt. # ______

City:______________________________State:______Zip:____________

Phone: (______) ______-__________

Interests/Comments: ____

____