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 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

THE HUGgers
HOOSIER USERS GROUP
People Helping People

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

Sept 1988 THE HUGgers NEWLETTER Volume 6, Number 8

TALKIN' SMART Part III

by

JIM ELLIS

(Cont'd from a previous issue) Last time I stopped with the mention of the 'S' registers, I now resume. There are registers, (places to store data), that may be changed, (depending on the modem that you may have,) to allow different settings for certain functions. Among these are how many seconds to wait for dial tone, etc. I have an Avatex 1200 modem that is only 90% Hayes (tm) compatible and its registers are NOT programmable. You take what they give you in that case. However, it is still a nice modem. The Desktalk II sports ram registers that may be be changed to suit your fancy. Most may be changed but not all may be stored in battery backed up memory. Much of the information you use to configure your modem may be saved in battery backed up memory so it will come up the way you like each time until you change it. However, there are limits to which items are stored. I'm not sure if the others specify which, but the manual for the Desktalk II doesn't tell you all that it saves. For instance, it stores one phone number, such as your favorite bbs, so that all you have to do to call it is type 'ATDS' and it will display the number in memory and dial it for you. Many are capable of returning an error code based on syntax. It can be either numeric or text based on the value you give a particular register. Included at the end of this article is the explanation of the pinout chart for RS-232C as used for most modems.

PIN	DESCRIPTION	DIRECTION
1	Protective Ground	N/A
2	Transmitted Data	TO modem
3	Received Data	FROM modem
5	Clear To Send	FROM modem
6	Data Set Ready	FROM modem
7	Signal Ground	N/A
8	Carrier Detect	FROM modem
20	Data Terminal Ready	TO modem
22	Ring Indicator	FROM modem

(to be cont'd)

WHAT'S NEW??

Two new data bases for the TI 99/4A.

TI-BASE - the following is a PRESS RELEASE, not a review.

TI-BASE. New from INSCEBOT, Inc.

TI-BASE is the place to collect and organize your data. Record Definitions are customized to your exact needs. Data can be freely interchanged between all types of records. Simple structured command language allows an easy implementation of needed functions. Address lists, checkbooks, business applications, all easily performed and maintained on diskette by TI-BASE.

DATA DEFINITION:
- Character
- Numerical
- Date
- Literals

MATH CAPABILITY:
- Arithmetic
- Trig
- Logical
- Boolean

DISK DIRECTIVES
- Initialization
- Catalog
- File Copy
- File Deletion

SORTED RECORDS:
- Nested command files
- Structured Directive

SORTED RECORDS:
- Sequential
- *Find*

(CONTINUED ON PAGE 5)

Reprint from MICROpendium

Learning the A-B-Cs

By REGENA

Summer vacation is here, and my children have been spending a lot of time playing computer games. My two-year-old also wants to use the computer, so the older children were trying to find programs he could use. He already knew the "S" key and "D" key that are used for TI-Invaders (the left and right arrow keys), but we decided educational programs would be better for him.

My favorite command modules for toddlers are Early Learning Fun from Texas Instruments and Early Reading and Addition and Subtraction from Scott, Foresman. If you have toddlers or preschoolers in your family, these modules are a "must" in your program collection.

Many of you may recall that my son Randy was born in 1980, the year I started doing a lot of work with the TI computer. Since he had an older sister who did everything for him, he didn't talk very early. One of the stories that I tell about him to computer user groups is that he loved to play with his "'puter" while I was working on my computer. We got the speech synthesizer and the Early Reading command module. This module has cute stories with the words written on the screen, and the computer talks. I noticed that Randy would repeat words the computer said. Some of the stories had "elephant," "tiger" and "astronaut." He soon learned the words of the computer stories. He still didn't say "Mom" or "Dad," but at least he was talking. I told the Scott, Foresman company they should have included a story about Mom and Dad.

Well, Brett Lynn (my present two-year-old) talked early and is now the one ready for the preschool programs. He likes the Early Learning Fun module and is learning the letters of the alphabet. However, I thought it would be better for the computer to say the name of each letter as it appears on the screen (Early Learning Fun was produced before the Speech Synthesizer was developed.)

This month's first program is written in TI Extended BASIC because speech is used. (Both programs required Extended BASIC and a speech synthesizer.) "Alphabet" is a simple adaptation that shows the capital letter on the screen and says the name of the letter. The child must then find the letter on the keyboard and press the proper key. When the key is pressed, the name of the letter is repeated. Sprites are used for the letters so they can move across the screen.

```
100 REM TI EXTENDED BASIC !0
74
110 REM SPEECH !114
120 REM!154
130 REM ALPHABET !251
140 REM!154
150 CALL CLEAR !209
160 CALL MAGNIFY(2)!223
170 FOR A=65 TO 90 !164
180 CALL SPRITE(#1,A,2,90,10
,0,30)!152
190 A$=CHR$(A)!167
200 CALL SAY(A$)!039
210 CALL POSITION(#1,ER,DC)!
201
220 IF DC<110 THEN 210 ELSE
CALL MOTION(#1,0,0)!240
230 CALL KEY(0,K,S)!187
240 IF S<1 THEN 230 !239
250 IF (K=A)+(K=A+32)THEN 27
0 !043
260 CALL SOUND(100,330,2)::
CALL SOUND(100,262,2):: GOTO
 230 !057
270 CALL SAY(A$)!039
280 FOR C=1 TO 20 :: CALL CO
LOR(#1,7):: CALL COLOR(#1,2)
:: NEXT C !190
290 CALL DELSPRITE(#1)!126
300 NEXT A !215
310 END !139
```

By the way, when Randy learned the letters from Early Learning Fun, he did not learn them in alphabetical order. The letters appear randomly in the program. I noticed he knew all the letters and he knew where they were on the keyboard — but he could not say the alphabet (and he learned to type before he could print).

This first program goes through the letters in alphabetical order. To modify the program so the letters appear randomly, change the following lines:

```
130 REM LETTERS (RANDOM)
170 RANDOMIZE :: A=(26*RND)+65
300 GOTO 170
```

You may SAVE this second program with a different name or on a different cassette. A random letter will appear on the screen and is named. The child needs to press the key with that letter, then the letter is said again. The screen clears, and another letter appears. The program continues indefinitely (press FCTN-4) to stop).

Line 150 clears the screen. Line 160 uses CALL MAGNIFY(2) to make the letters the large size sprite. Line 170 defines A for the character number. In the first program, there is a FOR-NEXT loop for all 26 letters of the alphabet in order. In the second program, a random letter is chosen.

Line 180 defines sprite #1 for the letter. The color 2 is black. The sprite starts in row 90 and column 10 and moves at a speed of 0 dot rows and 30 dot columns.

Line 190 defines the string A$ so line 200 can use CALL SAY to say the letter. Lines 210-220 check the position of the sprite and stop the letter when it is near the middle of the screen.

Lines 230-240 wait for the child to press a key. When a key is pressed, line 250 checks to see if it is the correct key matching the letter (either the shifted or unshifted key may be pressed). Line 260 sounds an "uh-oh" sound if the key pressed is incorrect, and the computer goes back to line 230. If the key pressed is correct, line 270 says the name of the letter, and line 280 blinks the letter by changing the color of the letter.

Line 290 deletes the sprite, and line 300 goes to the next letter. Line 310 ends the program.

LOWERCASE LETTERS

Most reading teachers say that as the child learns the letters he should learn to identify the lowercase letters as well as the capital letters, right from the beginning. The following program uses the same programming as the first program above but redefines characters so the lowercase letters are displayed. The child then matches the uppercase letter on the keyboard with the letter on the screen.

(Continued from Page 10)

Line 160 uses CALL MAGNIFY(4), which will be a sprite four characters enlarged. In Extended BASIC, four characters may be defined in one CALL CHAR statement. Each letter is made up of four possible characters, so each item in the DATA statements represents one letter. Each DATA statement (lines 210-450) defines two letters. The REMark statements are inserted to make the listing easier to read. Lines 170-200 define the graphics characters to print the lowercase letters.

When the program is RUN, the screen clears and is blank for about seven seconds while the characters are being defined.

Line 470 uses the loop for the 26 letters of the alphabet. The character number for the sprite will be A*4+36 because the characters are defined starting with character 40, and each letter involves four characters. The letters for the string A$ will still be the characters from 65 to 90, or A+64.

Feel free to use the basic programs and customize them by adding your own sounds and graphics for positive reinforcement for your children. You may wish to use the idea in this last program to redraw the capital letters — make better looking capital letters rather than using the built-in computer letters. And, of course, you may change the program to do random letters rather than the alphabet in order.

If you want to save typing effort, you may have a copy of these program by sending $3 plus a blank cassette or diskette and a stamped, self-addressed mailer to REGENA, P.O. Box 1502, Cedar City, UT 84720. Specify the title and that you need the TI version.

Lowercase letters

```
100 REM TI EXTENDED BASIC !0
74
110 REM SPEECH !114
120 REM!154
130 REM LOWERCASE LETTERS !1
62
140 REM!154
150 CALL CLEAR !209
160 CALL MAGNIFY(4)!225
 170 FOR C=40 TO 143 STEP 4 !
126
180 READ C$ !254
190 CALL CHAR(C,C$)!091
200 NEXT C !217
210 DATA 00000000000000000003D4
38181818181433D0000000000000000
00,0000808080808080BCC2818 18
181C2BC000000000000000000 !19
2
220 REM!154
230 DATA 00000000000000000003C4
2808080804 23C000000000000000000
00,0101010101013D4381818181 4
33D000000000000000000000 !08
6
240 REM!154
250 DATA 00000000000000000003C4
281FF8080423C000000000000000000
00,0609080808080 83E0808080 80
8080800000000000000000 !16
4
260 REM!154
270 DATA 3D4381818181433D0101
0101014122 1C000000000000000000
00,0000808080808080BCC2818 18
 1818181000000000000000000 !18
0
280 REM!154
290 DATA 00000078000000000080
8080808080808000000000000000000
00,0800000008080808080800
8088870000000000000000000 !07
9
300 REM!154
310 DATA 0000808080808080889
0A0C0A0908884000000000000000000
00,0808080808080808080808080
8080800000000000000000 !18
7
320 REM!154
330 DATA BCC2818181818181000
0000000000007884020202020200
20,000000000000000BCC2818 18
1818181000000000000000000 !15
9
340 REM!154
350 DATA 00000000000000003C4
2818181814 23C000000000000000000
00,BCC2818181 81C2BC8080808080
8000000000000000000000000 !18
8
360 REM!154
370 DATA 3D4381818181433D0101
0101010101000000000000000000000
00,0000000000000000BCC2818 08
0808080000000000000000000 !10
5
380 REM!154
390 DATA 00000000000000003C4
2403C0202423C000000000000000000
00,0000080808087F0808080808080
8080800000000000000000 !11
5
400 REM!154
410 DATA 00000000000000008 18
1818181814 33D000000000000000000
```

```
00,0000000000000000414122221
4140808000000000000000000 !02
4
420 REM!154
430 DATA 4141222214140808000
0000000000000004048888505020 2
0,0000000000000000824428 1028
448282000000000000000000 !012
440 REM!154
450 DATA 4141222214140808 101
0202040400000000000000000000000
00,0000000000000007F020408 1
0204070000000000000000000 !03
3
460 REM!154
470 FOR A=1 TO 26 !104
480 CALL SPRITE(#1,A*4+36,2,
80,10,0,30)!075
490 A$=CHR$(A+64)!156
500 CALL SAY(A$)!039
510 CALL POSITION(#1,DR,DC)!
201
520 IF DC<100 THEN 510 ELSE
CALL MOTION(#1,0,0)!028
530 CALL KEY(0,K,S)!187
540 IF S<1 THEN 530 !029
550 IF (K=A+64)+(K=A+96)THEN
 570 !087
560 CALL SOUND(100,330,2)::
CALL SOUND(100,262,2):: GOTO
 530 !103
570 CALL SAY(A$)!039
580 FOR C=1 TO 20 :: CALL CO
LOR(#1,16):: CALL COLOR(#1,2
):: NEXT C !239
590 CALL DELSPRITE(#1)!126
600 NEXT A !215
610 END !139
```

REPRINT from QB-99'er NEWSLETTER

TURNING PRINTERS INTO TYPEWRITERS

by Ed Machonis

There are often times when we just want to type a short note or letter and rather than load in a full blown word processing program, we settle for writing it out with such low tech implements as pens and pencils.

It is very easy to turn your printer into an electric typewriter. Four lines of Basic code will do it.

```
1 OPEN #1:"PIO"
2 INPUT A$
3 PRINT #1:A$
4 GO TO 2
```

This program enables the user to type a line of text, edit it as desired, and then print it by hitting the enter key.

Whenever a line of text is to be indented or contains a comma, that line must begin and end with a quotation mark ("). The quotes will not be printed nor will they be counted in the width of the line of text.

To skip a line, just hit enter.

This program allows sending of print codes directly to an Epson RX-80 printer provided they are in the same form as in the previously described RX-80 program. (i.e., CHR$(27)= CONTROL PERIOD) By pressing CONTROL PERIOD, then SHIFT E, and then <ENTER>, the print control code for emphasized type is sent to the RX-80 printer. Other codes, of course, can be sent in the same manner.

By adding a few more lines, the program can be made more useful. We can require an input as to the maximum line width to be printed and use this information to set equal right and left margins, A check has been added to insure that the maximum line width is not exceeded and it includes a prompt to display what a overly long line can be shortened to. User instructions have also been added. The expanded 10 Line Basic program looks like this.

PRINTALINE

```
1 PRINT ::::"TO INDENT TEXT
OR TO USE A COMMA, BEGIN &
END THAT LINEWITH QUOTATION
MARKS"::
2 INPUT "PRESS ENTER TO SKIP
 A LINE.
        HOW WIDE?(80 CHARAC
TERS MAX)":WIDTH
3 MARGIN-INT((80-WIDTH)/2)
4 OPEN #1:"PIO"
5 INPUT "
        INPUT LINE A LINE O
F TEXT:
         ":TEXT$
6 IF LEN(TEXT$)>WIDTH THEN 7
 ELSE 9
7 PRINT :"LINE TOO LONG! SH
ORTEN TO"::WIDTH;"CHARACTERS
 MAX."::SEG$(TEXT$,1,WIDTH):
:
8 GOTO 5
9 PRINT #1:TAB(MARGIN);TEXT$
10 GOTO 5
```

When typing notes, etc., where it is desireable to start printing at column one, input a line width of 80 and monitor the line width on the screen.

A simple way to use this program for correspondence is to use a line width of 56. This will fill exactly two lines of the TI screen. Right margin justification can be accomplished by inserting spaces between words until the second line of text is completely filled.

The OPEN statement in line 4 should be changed as required for the particular printer in use. The line width feature is designed for PICA print. Line 3 can be changed to accomodate ELITE or CONDENSED type styles.

(CONTINUED FROM PAGE 1)

LOCAL VARIABLES, ON LINE HELP, CHARACTER MANIPULATIONS

$24.95. Florida Residents add 6%, $1.50 postage and handling. Quantity prices on request.

INSCEBOT, Inc. P O Box 291610, Port Orange, FL 32029

Requires 32K memory expansion, a minimum of one disk drive and either Extended Basic, Editor Assembler or Mini/memory modules for the TI-99/4A

++++++++++++++++++++++++++++++++++++
++++++++++++++++++++++++++++++++++++

FIRSTBASE - Warren Agee's long worked upon and long awaited data base. The following is from a publicity release and is NOT a review:

-IBM style query commands

-Batch processing with four-function floating point math

-Multiple Keyword searches

-Extremely large capacity

-Handy Macros

With FIRSTBASE, the new data base manager from Olympys Technologies, you can organize, manipulate, and report data like never before. We listened to what users missed in other database managers, and what we've created is quite impressive. We increased record size to 3000 bytes, allowing you to store much more information than ever before. We allowed you to search for data using powerful query commands, similar to ones used on IBM compatible programs, so you can find exactly what you are looking for with a single command. We added a batch-processing capability, so you can perform "search and replace" functions or delete a group of records at once. You can ever perform MATH on any field. We added a MACRO feature so you can store frequently used commands on disk for quick recall.

```
####################################
#      THIS MONTH'S PROGRAM        #
#                IS                #
#      TI-BASE,HOW,WHY,WHAT        #
#       !!              !!         #
#       !!DON'T MISS IT!!          #
####################################
```

OFFICERS

PRESIDENT....JOHN POWELL 786-3270
VICE-PRES.....CARL CLARK 1-398-6226

SPECIFICATIONS

720 bytes/fields
3000 bytes/record
75 fields/record
32,767 records/databases
100 megabytes/file

-Search on multiple fields using AND and OR, or on Keywords

-Search on multiple Keys

-Quiries sent to screen, printer, new database, existing database

-Flexible report generator

-Written in c99

-Requires 32K, disk and either XB, E/A, or TI-WRITER Cartridge.

-Myarc Geneve 9640 Compatable

SUGGESTED RETAIL PRICE: $49.95

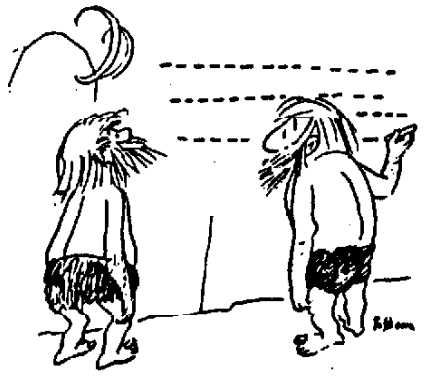
Available exclusively through GENIAL COMPUTERWARE, P.O. Box 183, Grafton Ma. 01519

Developed by Olympys Technologies, Advancing software technologies.

Scheduled release date: Late June or early July, 1988.

FIRSTBASE written by Warren Agee. c99 language by Clint Pulley.
XXXXXXXXXXXXXXXXXEOFXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

```
*********************************
*                               *
*    SOUTH SIDERS MEETING       *
*     SECOND-----THURSDAY       *
*      AFTER THE MEETING        *
*           MONTHLY             *
*        CALL 786-3270          *
*         FOR LOCATION          *
*                               *
*********************************
```



INSURANCE SALES
"I call it 'linear data processing'."

TIME DATED
SEPTEMBER 11 1988
MATERIAL

88/12 77255

Steven Sims
212 South Hancock
Indianapolis, IN 46222

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

Forwarding and Address
Correction Requested

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 doted line)

Check One:

Active Member

New: $20 ____
Renewal: 15 ____

Subscribing Member

New: $10 ____
Renewal: 7.50 ____

Amount Enclosed: $ ____

____ D ____
S ____

Name:_____ Today's Date:_____

Address:_____ Apt. # ____

City:_____ State:____ Zip:_____

Phone: (____) ____-____

Interests/Comments:_____