

FIRST CLASS MAIL

KANSAS CITY T199/4A COMPUTER USERS' GROUP  
POST OFFICE BOX 12591  
NORTH KANSAS CITY, MISSOURI 64116

The K.C. T199/4A Users' Group is in no way affiliated with or sponsored by Texas Instruments. Mention of a company or product is not considered an endorsement. Opinions expressed are those of the writers and not necessarily those of the Group, its officers, editor and the membership. Information from this Newsletter may be reproduced only if credit is given to both the author and this Newsletter.

: NEWSLETTER ONLY MEMBERSHIP \$12/YR :  
: USER GROUP MEMBERSHIP IS \$20/YR :

# KC 99'er CONNECTION

A KANSAS CITY PUBLICATION



\*\*\*\*\*  
: Volume 4 Issue 7 \*  
: << YOUR KANSAS CITY USER FRIENDLY GROUP >> \*  
\*\*\*\*\*

WE'LL LOOK FOR

YOU

AT THESE MEETINGS

- JULY 8 Regular business meeting
- JULY 22 Problem solving workshop
- AUGUST 12 Regular business meeting
- AUGUST 26 Problem solving workshop
- SEPTEMBER 9 FALL SWAP-N-SHOP
- SEPTEMBER 23 Annual Picnic
- OCTOBER 14 Regular business meeting

All meetings, workshops and the Swap-N-Shop begin at 2:00 PM and run two to three hours.

The gatherings are held at the ARTHUR MAB CENTER located behind the Midwest Research Institute off Cherry St. South of Volker Blvd. and Rockhill Road.

The annual picnic will be at the same location as last year: Macon Park, North Kansas City, MO. Time is not yet set but we will probably start at 3:00 PM. Bring a main dish and either a salad, a vegetable, or a dessert. Each family provides its own tableware as well as its own beverage.

INVITE A FRIEND WHO HAS A TI BUT HAS YET JOINED!

The BBS is no longer operating due to hardware problems. At our next meeting we will discuss whether to underwrite a membership to the Delphi network. We would use the BBS funds to finance the membership.

## TIPS FROM THE TIGERCUB

#56

TigerCub Software  
156 Collingwood Ave.  
Columbus OH 43213

I am still offering over 120 original and unique entertainment, educational and utility programs at just \$1.00 each, or in collection disks at \$5.00 per disk.

The contents of the first 52 issues of this newsletter are available as ready-to-run programs on 5 tips disks at \$10 each.

And my three Nuts & Bolts Disk, \$15 each, each contain over 100 subprograms for you to merge into your own programs to do all kinds of wonderful things.

My catalog is available for \$1, deductible from your first order (specify TIGERCUB catalog).

\*\*\*\*\*

## II-PU LIBRARY

I have selected public domain programs, by category, to fill over 200 disks, as full as possible if I had enough programs of the category, with all the Basic-only programs converted to Xbasic, with an E/A loader provided for assembly programs if possible, instructions added and any obvious bugs corrected, and with an autoloader by full program name on each disk. These are available as a copying service for just \$1.50 postpaid in U.S. and Canada. No fairware will be offered without the author's permission. Send SASE for list or \$1, refundable, for 7-page catalog listing all titles and authors. Be sure to specify II-PU catalog.

\*\*\*\*\*

In Tip #55, I showed you some quick and easy ways to create new character sets. Since folks nowadays don't like to key in long programs, let's continue with "tinygram" programming, and at the same time show you how to manipulate strings, and teach you the value of using MERGE format.

First, let's make a screen to display our new characters. Some of them will have to be double-spaced horizontally or vertically, so -

```
100 CALL CLEAR : X=1 : FOR CH=48
101 159 : PRINT CHR$(CH)&" " : X=X
+2 : IF X<29 THEN 110 ELSE PRINT "
" : X=X : GOTO 100
110 NEXT CH
```

Save it- SAVE DSK1.100.MERGE

Now, you might like to move the common punctuation marks into the same character sets as the characters, so that you will not have to reidentify so many sets, also so you can color them easier.

```
120 DATA 32,33,34,44,46
130 FOR J=1 TO 5 : READ CH : CALL
CHARPAT(CH,CH%) : CALL CHAR(J+90,C
H%) : CALL CHAR(J+122,CH%)
140 NEXT J : CALL CHARPAT(65,CH%) :
CALL CHAR(64,CH%) : CALL CHAR(
96,CH%)
```

If you want to program in Basic, or use BX8 with characters all the way up to ASCII 159, add CALL CHAR(J+154,CH%) to the end of line 130 and CALL CHAR(128,CH%) to the end of line 140.

Save by SAVE DSK1.120.MERGE

If you are using that trans-iteration, you must remember that with upper case characters the ? is @, space is !, ! is \, \* is ], comma is , period is . With the lower case they are FCIN keys C, F, A, G, W and V and for the 3rd set (ASCII 129 to 154) they are CTRL comma, period, !, =, \* and (.

You can transfer upper case to lower by - CALL CHARPAT(CH,CH%) and then CALL LCHAR(CH+32,CH%) or the opposite by CH-32 and if you have BX8 merged in you can create a 3rd set by CH+64.

The following are all incompatible with each other, so give them all line number 150 and save them in merge format as 150A, 150B, etc.

The numerals and the upper case letters all have the topmost pixel row blank to provide spacing between lines of text. We can make taller letters by deleting the top row and doubling the 7th row -

```
150 FOR CH=48 TO 126 : CALL CHARPA
T(CH,CH%) : CALL CHAR(CH,SEG$(CH%,3
,12)&SEG$(CH%,13,4)) : NEXT CH
151 REM
```

Or, you can double the 3rd row -

```
150 FOR CH=48 TO 95 : CALL CHARPAT
(CH,CH%) : CALL CHAR(CH,SEG$(CH%,3,
4)&SEG$(CH%,5,12)) : NEXT CH
151 REM
```

The lower case letters are really small upper case with the upper 3 rows blank. All their vertical bars are in the 4th, 6th and 8th rows, so let's drop the first 3 rows and quadruple the 7th.

```
150 FOR CH=97 TO 127 : CALL CHARPA
T(CH,CH%) : CALL CHAR(CH,SEG$(CH%,7
,6)&RPT$(SEG$(CH%,13,2),4)&SEG$(CH%
,15,2)) : NEXT CH
151 REM
```

Or, for topheavy letters, quadruple the 5th row -

```
150 FOR CH=97 TO 127 : CALL CHARPA
T(CH,CH%) : CALL CHAR(CH,SEG$(CH%,7
,2)&RPT$(SEG$(CH%,9,2),4)&SEG$(CH%
,11,6)) : NEXT CH
151 REM
```

Or, if you want line spacing -

```
150 FOR CH=97 TO 122 : CALL CHARPA
T(CH,CH%) : CH=SEG$(CH%,5,8)&RPT$(
SEG$(CH%,13,2),3)&SEG$(CH%,15,2) :
CALL CHAR(CH,CH%) : NEXT CH
151 REM
```

Or, for something silly -

```
150 FOR CH=48 TO 90 : CALL CHARPAT
(CH,CH%) : CALL CHAR(CH,SEG$(CH%,3,
2)&RPT$(SEG$(CH%,5,2),4)&SEG$(CH%,9
,4)&SEG$(CH%,15,2)) : NEXT CH
151 REM
```

For some good blocky characters -

```
150 FOR CH=48 TO 90 : CALL CHARPAT
(CH,CH%) : CALL CHAR(CH,RPT$(SEG$(C
H%,3,2),2)&SEG$(CH%,5,8)&RPT$(SEG$(
CH%,15,2),2)) : NEXT CH
151 REM
```

Or, if you would prefer them shorter for single-line spacing -

```
150 FOR CH=48 TO 90 : CALL CHARPAT
(CH,CH%) : CALL CHAR(CH,"00"&RPT$(S
EG$(CH%,3,2),2)&SEG$(CH%,7,6)&RPT$(
SEG$(CH%,15,2),2)) : NEXT CH
151 REM
```

If you would like numerals the same size as lower case.

```
150 FOR CH=48 TO 57 : CALL CHARPAT
(CH,CH%) : CALL CHAR(CH,"0000"&SEG$
(CH%,1,6)&SEG$(CH%,9,4)&SEG$(CH%,15
,2)) : NEXT CH
151 REM
```

You can even shrink the lower case to only 4 rows high, although some letters are not very legible -

```
150 FOR CH=97 TO 122 : CALL CHARPA
T(CH,CH%) : CALL CHAR(CH,SEG$(CH%,1
,6)&SEG$(CH%,5,4)&SEG$(CH%,11,6)) :
NEXT CH
151 REM
```

Something modernistic -

```
150 A$="00" :: FOR CH=48 TO 90 :: C
ALL CHARPAT(CH,CH$):: CALL CHAR(CH,
SEG$(CH$,1,4)&A$&SEG$(CH$,7,6)&A$&S
EG$(CH$,15,2)):: NEXT LH
151 REM
```

Or perhaps even better -

```
150 A$="00" :: FOR CH=48 TO 90 :: C
ALL CHARPAT(CH,CH$):: CH$=SEG$(CH$,
3,10)&RPT$(SEG$(CH$,13,2),2)&SEG$(C
H$,15,2)
151 CALL CHAR(CH,SEG$(CH$,1,4)&A$&S
EG$(CH$,7,2)&A$&SEG$(CH$,11,2)&A$&S
EG$(CH$,15,2)):: NEXT CH
```

I call this one "Spooky".

```
150 FOR CH=48 TO 122 :: CALL CHARPA
T(CH,CH$):: CH$=SEG$(CH$,3,14)&SEG$(
CH$,1,2):: X$=SEG$(CH$,1,1)&"O"
151 FOR J=3 TO 15 STEP 2 :: X$=1*%S
EG$(CH$,J,1)&SEG$(CH$,J-1,1):: NEXT
J :: CALL CHAR(CH,X$):: X$="" :: N
EXT CH
```

And "Spooky" backward -

```
150 FOR CH=48 TO 122 :: CALL CHARPA
T(CH,CH$):: FOR J=1 TO 15 STEP 2 ::
CH2$=CH2$&SEG$(CH$,J,1)&SEG$(CH$,J
+3,1):: NEXT J :: CALL CHAR(CH,CH2$
):: CH2$="" :: NEXT CH
151 REM
```

Now, clear the memory with NEW, then -

```
MERGE DSK1.100
MERGE DSK1.120
```

Add a line 500 GOTO 500

And start MERGEing in your series of "150" routines and running them to see what you have created.

Then, save these next routines in MERGE format as 160A, 160B, etc.

All normal characters have the leftmost column of pixels and the two right-most columns blank, for spacing between letters. We can widen the character into the left column -

```
160 FOR CH=48 TO 122 :: CALL CHARPA
T(CH,CH$):: FOR J=1 TO 15 STEP 2
161 CH2$=CH2$&SEG$(CH$,J,1,1,1)&SEG$(
CH$,J+1,1):: NEXT J :: CALL CHAR(CH
,CH2$):: CH2$="" :: NEXT CH
162 REM
163 REM
```

Or widen it both left and right -

```
160 FOR CH=48 TO 122 :: CALL CHARPA
T(CH,CH$):: FOR J=1 TO 15 STEP 2
161 CH2$=CH2$&SEG$(CH$,J-1,1,1)&SEG$(
"01234567",SEG$(CH$,J,1,1,1)&SEG$(
"02BA",POS("048C",SEG$(CH$,J+1,1,1
),1)
162 NEXT J :: CALL CHAR(CH,CH2$)::
CH2$="" :: NEXT CH
163 REM
```

Or even a full 8 columns wide by just changing the "02BA" in line 161 to "0129"

For darker characters, we can shade them into the 7th column -

```
160 FOR CH=48 TO 122 :: CALL CHARPA
T(CH,CH$):: FOR J=2 TO 16 STEP 2 ::
IF SEG$(CH$,J-1,1)="1" THEN CH2$=C
H2$%18" :: GOTO 163
161 IF CH=67 OR CH=71 OR CH=99 OR C
H=103 THEN 162 :: IF SEG$(CH$,J-1,1
)="3" AND SEG$(CH$,J,1)="0" THEN CH
2$=CH2$%60" :: GOTO 163
162 CH2$=CH2$&SEG$(CH$,J-1,1)&SEG$(
"0367CBEF",POS("02468ACE",SEG$(CH$,
J,1,1,1)
163 NEXT J :: CALL CHAR(CH,CH2$)::
CH2$="" :: NEXT CH
```

Or shade them both left and right -

```
160 FOR CH=48 TO 122 :: CALL CHARPA
T(CH,CH$):: FOR J=1 TO 15 STEP 2 ::
A$=SEG$(CH$,J,1):: P=POS("01234567
89ABCDEF",A$,1)
161 A$=SEG$(CH$,J+1,1):: P=POS("0246
8ACE",B$,1):: B$=SEG$(CH$,J+1,1,1)
162 NEXT J :: CALL CHAR(CH,CH2$)::
CH2$="" :: NEXT CH
163 CALL CHAR(74,"000000000000000000
0000000000000000000000000000000000
")
```

Or shaded into both of the rightmost columns -

```
160 FOR CH=48 TO 122 :: CALL CHARPA
T(CH,CH$):: FOR J=2 TO 16 STEP 2 ::
CH2$=CH2$&SEG$(CH$,J-1,1)&SEG$(CH$,
J,1,1,1):: NEXT J :: CALL CHAR(CH,CH
2$):: CH2$="" :: NEXT CH
161 REM
162 REM
163 REM
```

Or into all 8 columns -

```
150 FOR CH=48 TO 122 :: CALL CHARPA
T(CH,CH$):: FOR J=1 TO 15 STEP 2 ::
P=POS("0123456789ABCDEF",SEG$(CH$,
J,1,1)
151 A$=SEG$(CH$,J,1,1,1,1,1,1):: P=POS("02468ACE",SEG$(CH$,J+1,1,1,1,1,1,1)
152 NEXT J :: CALL CHAR(CH,CH2$)::
CH2$="" :: NEXT CH
153 REM
```

More neatly, shaded inward at right -

```
160 FOR CH=48 TO 122 :: CALL CHARPA
T(CH,CH$)
161 FOR J=1 TO 15 STEP 2 :: CH2$=CH
2$&SEG$(CH$,J,1)&SEG$(CH$,J,1,1,1,1,1)
162 CALL CHAR(CH,CH2$):: CH2$="" ::
NEXT CH
163 REM
```

Or inward at right, outward at left -

```
160 FOR CH=48 TO 122 :: CALL CHARPA
T(CH,CH$):: FOR J=1 TO 15 STEP 2
161 CH2$=CH2$&SEG$(CH$,J,1,1,1)&SEG$(
"01234567",SEG$(CH$,J+1,1,1,1)&SEG$(
"0CBC",POS("048C",SEG$(CH$,J+1,1,1
),1)):: NEXT J
162 CALL CHAR(CH,CH2$):: CH2$="" ::
NEXT CH
163 REM
```

Here's a weirdo -

```
160 FOR CH=48 TO 122 :: CALL CHARPA
T(CH,CH$):: FOR J=9 TO 15 STEP 2
161 CH2$=CH2$&SEG$(CH$,J,1,1,1)&SEG$(
"01234567",SEG$(CH$,J,1,1,1)&SEG$(
"02BA",POS("048C",SEG$(CH$,J+1,1,1
),1)
162 NEXT J :: CALL CHAR(CH,SEG$(CH$,
1,8)&CH2$):: CH2$="" :: NEXT CH
163 REM
```

Try changing that to FOR J=1 TO 7 and CALL CHAR(CH,CH2\$&SEG\$(CH\$,9,8))

And one more -

```
160 FOR CH=48 TO 122 :: CALL CHARPA
T(CH,CH$):: FOR J=1 TO 7 STEP 2
161 A$=SEG$(CH$,J,1,1,1,1,1,1):: B$=SEG$(CH$,J+1,1,1,1,1,1,1)
162 CALL CHAR(CH,CH2$&A$&B$):: CH2$="" :: NEXT CH
163 REM
```

Now, clear the memory, MERGE in 100 and 120, out in a holding line 500 GOTO 500 and start MERGEing in all of the different combinations of the 150 and 160 lines and see how many different character sets you can make!

Memory full.

Jim Peterson

\* HINTS AND TIPS ASSEMBLY TUTORIAL #2  
 \* FROM VASI 99 NEWSLETTER  
 \* PRINTS (A) ACROSS THE SCREEN

```
*
DEF A
REF V8BW
A LI R0,0
A1 LI R1,24100    LOADING THE LETTER A TO SCREEN IN LEFT CORNER
BLWP @V8BW
LI R1,1000      START OF TIME DELAY LOOP
DEC R1          DECREASE CONTENTS OF R1 BY 1
```

```
*
* THIS INSTRUCTION TAKES TWO BYTES
JNE $-2        JNE MEANS JUMP IF NOT EQUAL.
* LOOKING AT THE PREVIOUS INSTRUCTION CHECKS TO SEE IF IT IS EQUAL
* IF NOT IT GOES BACK TWO BYTES ($-2) UNTIL IT REACHES EQUAL BEING (0)
* HERE
```

```
*
LI R1,2000     PLACES SPACE CHARACTER INTO R1
BLWP @V8BW     PRINTS TO SCREEN IN LOCATION R0
LIMI 2        ALLOWS INTERRUPT SO CAN USE QUIT KEY
LIMI 0        DISABLES INTERRUPT
LI R0,76/     COMPARE IMMEDIATE THE CONTENTS OF R0 TO 76/
* CHECKS TO SEE IF WRITTEN TO LAST SCREEN LOCATION
```

```
*
JEW A         IF C1 IS EQUAL THEN JUMP TO LABEL A WHICH
* STARTS THE PROGRAM OVER
```

```
*
INL R0        IF C1 NOT EQUAL INCREASE VALUE IN R0 BY 1
* GIVEN THE NEXT SCREEN LOCATION
```

```
*
JMP A1        JUMP TO THE INSTRUCTION A1 WHICH LOADS (A)
* (INIU R1
```

```
*
END A         THE END DIRECTIVE FOLLOWED BY PROGRAM NAME
* LOADING THIS WILL AUTO-START PROGRAM
```

#### II BASIC WORDS

WORD	LAST LETTER
ERABK	--- K
IENLLIND	--- E
SILL	--- I
UNIALMLUIL	--- N
EPKSI	--- E
UTIALLIN	--- I
KACHB	--- K
UNPE	--- N
EVAS	--- E
ENLQASLREC	--- N

#### LISTING WITH COMMENTS (reprinted from Punn Newsletter)

Have you ever wanted to place comments on a catalog listing, especially when passing a diskette on to a friend. If you do not have Disk Utility 4 by John Birdwell, then try the following. Go crank up your Funnelweb disk and load up DM 1000. Now do these steps:

1. Go into DM-1000 and change the output device by pressing FCTN 3 at the main menu. You will need to change from 'PIQ' to 'Dskn.filename'. As you don't want printer codes and you don't wish to save permanently, press 'N' at the option.
2. Go through the normal procedure to catalog a disk. Once the catalog is displayed, press FCTN 7 to print. However, instead of going to your printer. It goes to your Disk Drive, creating a DVBC file with the file-name you give it in step 1.
3. Go back to Funnelweb and load in TI-Writer. Now you can add all the comments you wish to each line that is displayed in the listing. Be sure to turn word-wrap off (CTRL O).

By using this method you can make your listing more informative to the person who will be receiving it. For example, programs related together can be grouped together regardless to their names.

THE NEXT GENERATION by Ted Holt, Wa. Carey College

The rapid-fire change of the field that I'm in, while welcome, still sometimes confuses. When I'm just settled in, to my great chagrin, Big Blue changes the words that it uses. These wondrous machines that we use every day I call personal COMPUTERS. Do you? That word's had its day. It's clearly passe. Now they're personal SYSTEMS (slash two). We've hoarded our data (such money we've spent) Since Holerith's days of the card file. To tape it then went, on to floppies was sent. Then to Winchester disk, now called HARD FILE. A new way of moving the cursor came through. "A mouse!" we exclaimed. "Oh how nice!" "But wait," said Big Blue. "A rodent won't do. From now on, it's POINTING DEVICE." Think of the many fine dots on a screen. They're pixels, we all know quite well. Not so! Whether green, blue, or somewhere between. From now on each one is a PEL. Uh tv celebs, since Korea unseen, the whole truth tell, in plain English please: that new on the scene is not merely machine but the next generation in computerese!

(from BAHAMA'ILUN, August 1, 1987)

#### II BASIC WORDS

##### CORRECT WORD

BREAK  
 CONTINUE  
 LIST  
 CALLMOTION  
 SPRITE  
 CALLINIT  
 GCHAR  
 OPEN  
 SAVE  
 CALLSCREEN

Computerized translation of languages has a way to go. Bell Laboratories tested new software. They entered the saying - "Out of sight, out of mind" into the computer for translation into Chinese and then back to English. And what came out? "invisible idiot."

Industry Week 8/15/88