Centre Area Commodore Computer Club

Q: Can the C64/128 do 2400 baud?
A: It certainly can. The C64 can do 2400 baud in a not-so-reliable manner though. You will able to receive or transmit data at 2400 baud, but probably you won't be able to have any kind of file emulation or 80 column screen.

Currently, the only PD program I know of for the C64 that will do a more-or-less-reliable 2400-baud communication is CCGMS7.0.

The C128, with its RGB 80 Column output and 2Mhz mode, can go up to 9600 baud. (Though, some older C128, can only do 4800 baud)

Currently, a shareware program called Desterm will do 9600 baud and also has VERY nice features. (Shareware $25)

Q: Is there a VT100 emulator for the C64 that is 100% compatible?
A: Sure there is, for the C64 there are plenty available, like:

```
        v/vtermite      kermit          vt100.emulator
```

just to name a few. Some are more compatible than others. I have found no problems with "kermit." Most incompatibility problems with kermit arises when kermit starts using flow control and the host doesn't recognize that. They can be easily worked out by reading the kermit documentation.

For the C128, there are also several vt100 emulators, like vt100-128 and Desterm. Both claim to support 100% compatibility with VT100. Again, most compatibility problems with those programs can be fixed by reading the manual and setting the right parameters.

(NOTE Commodore 64 programs WILL run on a 128 in 64 mode!)

Q: Are there any electronic archive sites for the C64 or C128 software?
A: Yes, currently there are some FTP sites and a couple of mail servers.
The FTP sites are:

tolsun.oulu.fi	128.214.5.6
maxwell.physics.purdue.edu	128.46.135.3
milton.u.washington.edu	128.95.136.1
oswego.oswego.edu	129.3.1.1
tukky.jyu.fi	128.214.7.5

The Mail Servers are:
acliu@skat.usc.edu
corpane!disk specter@ms.uky.edu

For more information on how to use the server, send a message with
Subject: Mail-Archive-Request
then put "help" in the body of the message.
For those without FTP access, you can contact ftp sites by e-mail.
Send e-mail to "bitftp@pucc.bitnet" or "bitftp@pucc.princeton.edu".
Somewhere in the body of the message put the word "help" for more info.

Q: Where to get Desterm or Kermit?
A: You can get it from a BBS or from an archive site.

Q: What is the Punter Protocol?
A: Punter's C1 Protocol is a file transfer protocol similar to Xmodem, but a lot more reliable. It was written by Steve Punter, one of C64 Telecommunication's pioneers. You may be able to get PAL sources for his code from some of the finest archive sites.

Please, e-mail any suggestions or corrections

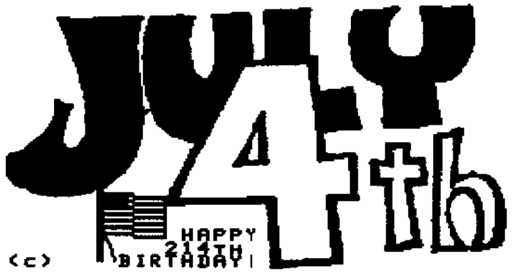
Alex C. Liu	INTERNET: acliu%skat@usc.edu
Voice: (213) 749-2730	BITNET: acliu@gamera
Q-Link: Alejandro	UUCP: ...!usc!acliu

N U T I   N E W S

* NITTANY USERS OF TEXAS INSTRUMENTS *

L. Chapin, Pres.   JULY 1990   M. Villano, Ed.

Send Exchange Newsletter
to NUTI 625 Wiltshire Dr.
State College, Pa. 16803
(Do NOT send to the MUG)
Materials may be copied.
Cite author & NUTI NEWS.



ARTICLES BEING FEATURED THIS ISSUE:

SUMMER MEETINGS SCHEDULE (REPEAT):

Due to anticipated inactivity, schedule posted in MUG for 3rd Tuesdays is valid for July and August except *no* organized meetings are planned. The dates are kept open for informal gatherings (group or individual), on any matters pertaining to computer use or club business (238-0396).

NUTI OFFICERS TO DEPART LOCAL AREA:

Long-time NUTI president (1986-90) *LuVern O. "Chip' Chapin* has taken a new Job in Washington, DC, and expects to make a permanent move there, sometime during the summer (read my comments elsewhere in this Issue). Another veteran member and until two years ago our software librarian, *Christopher A. "Chris" Stone* recently graduated from *SCHS* with honors. He received a *Merit Scholarship* and will study computer engineering or computer science at either at *Carnegie Mellon* in Pittsburgh, or at the *Washington University* in St. Louis. *Congratulations from NUTI, Chris!*

(c) Montage of ARTIST INSTANCES. From *Asgard Software*, Rockville, Md.

HAPPENINGS OF LOCAL INTEREST, ETC.:

Club prexy *Chip Chapin's* departure to a new job location leaves a void that's hard to fill, both for the *NUTI* organization and personally for me. *Chip* was the spiritual leader of our Group and my close computing buddy, and on both counts his loss to us will have significant effect. *Chip's* tenure as president, these past four years, saw us grow, if not in numbers then certainly in visibility and semblance of organization.

In early 1986, *NUTI* was a very loose assemblage of game-players, still reeling in the aftershocks of being orphaned by *TI*, with minimal hardware configurations of console-only, data cassette recorder, and TV as the standard. Not many members had the *Extended Basic* cartridge, or a speech synthesizer, and an intrepid few even wrote their own programs. Today, most of our members have expanded their *99/4A*'s with additional memory and drives, and use productivity software available for the *TI*.

Under *Chip's* stewardship, we published *almost 50 consecutive* issues of *NUTI NEWS*, and continued newsletter exchanges with *over 30* other users groups. We are listed in *COMPUTER SHOPPER, COMPUTER BUYER'S GUIDE* and the local *CENTRE DAILY TIMES*, and have manned booths at the Penn State Annual Computer Fair *(MIEC)* the past three years. Good relations have been maintained with regional user groups we met at the *TICOFF* and the *Central Pa.* fairs, thanks to *Chip's* outgoing personality and interest.

As a member, *Chip* was more than generous in making his talents and his resources available to us. Besides opening his house for meetings, he has spent countless hours helping others with their programming, or in re-pinning cables, installing drives, etc. and running up connect-time charges on the networks, downloading shareware and information for us. As the *Chez Geneve* columnist for our publication, Chip has been at the forefront of knowledge on the *MYARC 9640* and dabbles in *Advanced Basic* and *FORTRAN*. His articles have been reprinted in several newsletters.

And a personal note, *Chip* and your *Editor* have been kindred spirits in being the *only Geneve owners* in the Group. We have spent endless time in face-to-face meetings, voice phone or by modem hook-up debating the efficacies and limitations of the *9640*, its hardware, software and the more philosophical 'where-we-are-headed." We routinely troubleshooted each other's equipment, when a part was thought failing, by trial-and-error exchanges of our cards and peripherals. This diagnosis required no test bench to pinpoint problems and saved down-time and expense for shipping the item back to the manufacturer for repairs or replacement.

Chip literally taught me <u>all</u> I know about the Geneve and its operating system. As my mentor, I was always one step behind. When he bought a *9640* machine, I got one soon after. He installed half-height floppies and so did I, and so-on and so-on with upgraded monitors, modems, hard drives, added memory etc..(but note: this waiting period enabled me to benefit from his negative experiences. Fortunately his decisions have been mostly correct and I reaped the rewards). *Chip could have been a saleman for Lou Phillips!* As they say imitation is the *sincerest* form of flattery. *Chip*, best wishes for future success. Your contribution as a guest columnist will be most welcome. We shall miss you, but our loss will be *Mid-Atlantic 99'ers* or some other fortunate user groups's gain. We know you will remain a *loyal 99000'er*. Let's keep in touch!

TIPS FROM THE TIGERCUB

#58.1

Tigercub Software
156 Collingwood Ave.
Columbus OH 43213

I am still offering over 120 original programs at $1 each, or on collection disks at $5 each. The five Tips From The Tigercub disks are reduced to $5 and the three Nuts & Bolts disks are now just $10 each.

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

****

TI-PD LIBRARY

I have selected public domain programs, by category, to fill over 300 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 11-page catalog listing all titles and authors. Be sure to specify TI-PD catalog.

****

In Tips #55 I published a CHARSUB routine to convert character patterns into assembly source code, and in Tips #55 and #56 I published several routines to manipulate hex codes into new character sets. Those patterns looked fine on my old TV, but when I demo'd them on a high-resolution monitor I could see too many missing pixels.

So I wrote this CHARFIX program which, when MERGEd into a program and CALLed after any character redefinition is completed, will permit any normal or re-identified character to be viewed on screen and edited and will then write the hex codes of any range of printable characters into an assembly source file which can be assembled, loaded and linked to instantly change character sets.

This routine also reidentifies the common punctuation into the same character sets as the letters, as described in Tips #55. If you do not want this feature, delete lines 29001-29003.

```
29000 SUB CHARFIX
29001 DATA 32,33,34,44,46
29002 RESTORE 29001 :: FOR J
=1 TO 5 :: READ CH :: CALL C
HARPAT(CH,CH$):: CALL CHAR(J
+90,CH$):: CALL CHAR(J+122,C
H$):: NEXT J
29003 CALL CHARPAT(63,CH$)::
 CALL CHAR(64,CH$):: CALL CH
AR(96,CH$)
29004 DISPLAY AT(1,1)ERASE A
LL:"1 2 3 4 5 6 7 8 9 0 : ;"
:" ":"@ A B C D E F G H I J
K L M":" ":"N O P Q R S T U
V W X Y Z [":" ":"\ ] ^ _ a
b c d e f g h i j"
29005 DISPLAY AT(9,1):"k l m
 n o p q r s t u v w x":" ":
"y z { | } ~"
29006 CALL CHAR(128,"FF"&RPT
$("81",6)&RPT$("FF",9)&"FFFF
"&RPT$("C3",4)&"FFFF"):: CAL
L COLOR(13,2,16)
29007 CALL CHARVIEW
29008 SUBEND
29009 SUB CHARVIEW
29010 DISPLAY AT(13,14):"CTR
L V TO VIEW" :: DISPLAY AT(1
4,14):" " :: DISPLAY AT(15,1
4):"CTRL E TO EDIT" :: DISPL
AY AT(17,14):"CTRL S TO SAVE
"
29011 DISPLAY AT(19,14):" "
 :: DISPLAY AT(20,14):" "
29012 CALL KEY(0,@,S):: IF S
=0 THEN 29012 ELSE IF @=150
THEN 29015 ELSE IF @=133 THE
N 29014 ELSE IF @=147 THEN 2
9013 ELSE 29012
29013 CALL DELSPRITE(#1):: C
ALL CHARSUB(HX$()):: DISPLAY
 BEEP :: STOP
29014 CALL EDIT(K):: GOTO 29
010
29015 DISPLAY AT(24,1)BEEP:"
"
29016 DISPLAY AT(24,1):"PRES
S A KEY" :: CALL KEY(0,K,S):
: IF S<1 OR K<32 OR K>143 TH
EN 29016
29017 DISPLAY AT(24,1):"" ::
 CALL CHARPAT(K,CH$)
29018 R=13 :: FOR J=1 TO 15
STEP 2
29019 H$=SEG$(CH$,J,1):: CAL
L HEX_BIN(H$,B$)
29020 H$=SEG$(CH$,J+1,1):: C
ALL HEX_BIN(H$,BB$):: FOR L=
1 TO 8 :: C$=C$&CHR$(ASC(SEG
$(B$&BB$,L,1))+80):: NEXT L
29021 DISPLAY AT(R,1):C$;::
DISPLAY AT(R,10):SEG$(CH$,J,
2);:: R=R+1 :: C$="" :: NEXT
 J :: DISPLAY AT(22,1):CH$;:
: GOTO 29012
29022 SUBEND
29023 SUB HEX_BIN(H$,B$):: H
X$="0123456789ABCDEF" :: BN$
="0000X0001X0010X0011X0100X0
101X0110X0111X1000X1001X1010
X1011X1100X1101X1110X1111"
29024 FOR J=LEN(H$)TO 1 STEP
 -1 :: X$=SEG$(H$,J,1)
29025 X=POS(HX$,X$,1)-1 :: T
$=SEG$(BN$,X*5+1,4)&T$ :: NE
XT J :: B$=T$ :: T$="" :: SU
BEND
29026 SUB CHARSUB(HX$())
29027 DISPLAY AT(12,1)ERASE
ALL:"Source code filename?":
"DSK" :: ACCEPT AT(13,4)SIZE
(12)BEEP:F$ :: OPEN #1:"DSK"
&F$,OUTPUT
29028 DISPLAY AT(15,1):"LINK
ABLE program name?" :: ACCEP
T AT(16,1)SIZE(6):P$
29029 DISPLAY AT(18,1):"Rede
fine characters from    ASCI
I      to ASCII"
29030 ACCEPT AT(19,7)VALIDAT
E(DIGIT)SIZE(3):F
29031 ACCEPT AT(19,21)VALI
DATE(DIGIT)SIZE(3):T
29032 PRINT #1:TAB(8);"DEF";
TAB(13);P$ :: PRINT #1:"VMBW
   EQU  >2024" :: PRINT #1:"
STATUS EQU  >837C"
29033 NB=(T-F)*8 :: CALL DEC
_HEX(NB,H$):: A=768+F*8 :: C
ALL DEC_HEX(A,A$)
29034 FOR CH=F TO T :: IF CH
<144 THEN CALL CHARPAT(CH,CH
$)ELSE CH$=HX$(CH)
29035 IF FLAG=0 THEN PRINT #
1:"FONT";:: FLAG=1
29036 FOR J=1 TO 13 STEP 4 :
: M$=M$&">"&SEG$(CH$,J,4)&",
" :: NEXT J :: M$=SEG$(M$,1,
23)&"  *"&CHR$(CH)
29037 PRINT #1:TAB(8);"DATA
"&M$ :: M$="" :: NEXT CH
29038 PRINT #1:P$;TAB(8);"LI
   R1,FONT" :: PRINT #1:TAB(
8);"LI   R0,>"&A$ :: PRINT #
1:TAB(8);"LI   R2,>"&H$
29039 PRINT #1:TAB(8);"BLWP
@VMBW":TAB(8);"CLR  @STATUS"
:TAB(8);"RT":TAB(8);"END" ::
 CLOSE #1
29040 SUBEND
29041 SUB DEC_HEX(D,H$)
29042 X$="0123456789ABCDEF"
:: A=D+65536*(D>32767)
29043 H$=SEG$(X$,(INT(A/4096
)AND 15)+1,1)&SEG$(X$,(INT(A
/256)AND 15)+1,1)&SEG$(X$,(I
NT(A/16)AND 15)+1,1)&SEG$(X$
,(A AND 15)+1,1):: SUBEND
29044 SUB EDIT(CH)
29045 DISPLAY AT(13,14):"1 T
O TOGGLE" :: DISPLAY AT(14,1
5):"CURSOR" :: DISPLAY AT(15
,14):"E S D X TO MOVE" :: DI
SPLAY AT(17,14):"CTRL A TO A
BORT"
29046 DISPLAY AT(19,14):"CTR
L R TO" :: DISPLAY AT(20,15)
:"REIDENTIFY"
29047 R=13 :: C=3 :: X=128 :
: CALL SPRITE(#1,130,11,R*8-
7,C*8-7):: X$=CHR$(129)&CHR$
(146)
29048 CALL KEY(0,K,S):: IF S
<1 THEN 29048 ELSE ON POS("1
EeSsDdXx"&X$,CHR$(K),1)+1 GO
TO 29048,29049,29050,29050,2
9051,29051,29052,29052,29053
,29053,29055,29056
29049 X=X+1+(X=129)*2 :: GOT
```

```
0 29054
29050 R=R-1-(R=13):: GOTO 29
054
29051 C=C-1-(C=3):: GOTO 290
54
29052 C=C+1+(C=10):: GOTO 29
054
29053 R=R+1+(R=20)
29054 CALL LOCATE(#1,R*8-7,C
*8-7):: CALL HCHAR(R,C,X)::
GOTO 29048
29055 CALL DELSPRITE(#1):: S
UBEXIT
29056 FOR R=13 TO 20 :: FOR
C=3 TO 10 :: CALL GCHAR(R,C,
GH):: CALL LOCATE(#1,R*8-7,C
*8-7):: B$=B$&CHR$(GH-80)::
NEXT C
29057 CALL BIN_HEX(B$,H$)::
DISPLAY AT(R,10):H$ :: B$=""
 :: HEX$=HEX$&H$ :: NEXT R :
: DISPLAY AT(22,1):HEX$;:: C
ALL CHAR(CH,HEX$):: HEX$=""
29058 CALL DELSPRITE(#1):: F
OR R=13 TO 20 :: DISPLAY AT(
R,14):"" :: NEXT R :: SUBEND
29059 SUB BIN_HEX(B$,H$):: H
X$="0123456789ABCDEF" :: BN$
="0000X0001X0010X0011X0100X0
101X0110X0111X1000X1001X1010
X1011X1100X1101X1110X1111"
29060 L=LEN(B$):: IF L/4<>IN
T(L/4)THEN B$="0"&B$ :: GOTO
 29060
29061 FOR J=L-3 TO 1 STEP -4
 :: X$=SEG$(B$,J,4)
29062 X=(POS(BN$,X$,1)-1)/5
:: T$=SEG$(HX$,X+1,1)&T$ ::
NEXT J :: H$=T$ :: T$="" ::
SUBEND
```

****

I think programs, at least non-commercial ones, should be open for anyone to modify for their own use. For that reason, I would not normally publish the following routine. However, I recently received a large number of programs, originally in the IUG library, and found that the author's name had been erased from the title screen or REM of ALL of them. I know, for I already had many original versions, including some that I wrote myself.

Now, that is inexcusable. If a programmer is willing to share his work, he does deserve credit for it. And if people are going to play dirty, maybe there is a good reason to protect programs.

So here is how to do it. Ken Woodcock wrote this ingenious routine and published it in the Tidewater newsletter. I have modified it so that it can be deleted after it has done its work. It is to be MERGEd into any XBasic program(32k required) and RUN, and will change the line length byte of each line to zero, so that the program cannot be LISTed, altho it can load and run:

```
1 CALL INIT :: CALL PEEK(-31
952,A,B,C,D):: SL=C*256+D-65
539 :: EL=A*256+B-65536 :: F
OR X=SL TO EL STEP -4
2 CALL PEEK(X,E,F,G,H):: ADD
=G*256+H-65536 :: J=J+1 :: I
F J<4 THEN 3 :: CALL LOAD(AD
D-1,0)
3 NEXT X :: STOP :: !@P-
```

Save that as FIX in MERGE format. Merge it into any program (RESequence first if it has line numbers less than 4) and RUN. Then type 1, FCTN X and FCTN 3 to delete line 1. Delete lines 2 and 3 in the same way. Then SAVE. Now try LISTing it and watch the fireworks.

Ken wrote an even more ingenious UNFIX routine to unprotect the program, but I'm not passing that on!

Now, suppose you have a party game program that you don't want the kids playing with. So, RESequence it to some odd number, such as RES 797. Put in a line just before that 796 STOP . Then merge in FIX, run it, and delete those first 3 lines.

I hope you remember what line number you resequenced it to start from, because now you can only run it by RUN 797 !

****

In Tips #57 I reported the discovery that printing to the disk from the TI- Writer Formatter, with the C option, really converted the carriage returns to trailing blank ASCII 32's, and I published a routine to strip them. I have found an easier way. First PF and C DSK... to convert the CRs to blanks. LF DSK... and SF DSK... to strip out those blanks, but that leaves the pestiferous tab line, so LF DSK... and PF DSK... again!

The first few disks of Tips #58 that I sent out had a poor version of this program. This is the corrected version. First key this in:

```
1 DISPLAY AT(12,1)ERASE ALL:
"SKIP INSTRUCTIONS? Y" :: AC
CEPT AT(12,20)SIZE(-1)VALIDA
TE("YNyn"):@Q$ :: IF @Q$="Y"
 OR @Q$="y" THEN 8
2 DISPLAY AT(24,5)ERASE ALL:
"PRESS ANY KEY"
3 RESTORE 30721
4 REM
5 FOR J@=1 TO T@ :: READ @$
:: DISPLAY AT(J@,1):@$:" "
6 CALL KEY(0,K@,S@):: IF S@=
0 THEN 6
7 NEXT J@
8 DATA 0
9 RESTORE 8 :: READ N
10 REM
```

Save it- SAVE DSK1.D/MERGE, MERGE - Then key this in:

```
100 OPEN #1:"DSK1.D/MERGE",V
ARIABLE 163,INPUT :: OPEN #2
:"DSK1.D/MERGE2",VARIABLE 16
3,OUTPUT :: L=129 :: FOR J=1
 TO 10
110 LINPUT #1:M$ :: PRINT #2
:CHR$(0)&CHR$(L+J)&CHR$(156)
&CHR$(253)&CHR$(200)&CHR$(1)
&"1"&CHR$(181)&CHR$(199)&CHR
$(LEN(M$))&M$&CHR$(0):: NEXT
 J
120 CLOSE #1 :: PRINT #2:CHR
$(255)&CHR$(255): CLOSE #2
```

Run it to convert D/MERGE into a merge format file D/MERGE2 on DSK1. Then key this in. Don't change line numbers:

```
100 CALL CLEAR :: OPEN #1:"D
SK1.@DATA",VARIABLE 163,OUTP
UT :: DEF L$(X)=CHR$(120)&CH
R$(X)
105 PRINT #1:L$(X)&CHR$(161)
&CHR$(200)&CHR$(6)&"@DUMMY"&
CHR$(0)
110 L=L+1 :: X=X+1 :: ACCEPT
 AT(L,0):M$ :: IF L=24 THEN
CALL CLEAR :: L=0
120 IF M$<>"END" AND M$<>"en
d" THEN PRINT #1:L$(X)&CHR$(
147)&CHR$(199)&CHR$(LEN(M$))
&M$&CHR$(0):: GOTO 110
130 REM
140 PRINT #1:CHR$(0)&CHR$(4)
&"T@"&CHR$(190)&CHR$(200)&CH
R$(LEN(STR$(X-1)))&STR$(X-1)
&CHR$(0)
141 PRINT #1:L$(X)&CHR$(168)
&CHR$(0)
150 PRINT #1:CHR$(255)&CHR$(
255):: CLOSE #1
```

Enter MERGE DSK1.D/MERGE2 to merge in that file. SAVE the program as DATAWRITER. Then RUN it and try it out by using it to write itself some instructions. Answer the prompts with:

DATAWRITER V1.2
by Jim Peterson

To be used to add instructions to programs.

Type the instructions and format them, centered or hyphenated or right-adjusted just as you want them to appear on screen, and enter each line. They will be written to a D/V163 file named @DATA. When finished, enter END.

Then enter NEW, then MERGE DSK1.@DATA, and RUN to see if everything is OK. If so, load the program needing instructions, make sure its lowest line number is more than 10 and the highest is less than 30721, and enter MERGE DSK1.@DATA.

And enter END, then OLD DSK1.DATAWRITER, then MERGE DSK1.@DATA.

Jim Peterson

STUMBLING BLOCKS

By Chip Chapin

This article includes a short program (2 pages) which demonstrates the 9640 FORTRAN Character Redefinition capability. Basic and Extended Basic programers will recognize the commands and (hopefully) the method used to modify and display the redefined characters. There are advantages in MDOS, but more about that later.

For those not familiar with this subject, the idea is to be able to create screen graphics characters or pieces of characters by modifying the shapes assigned to the ASCII values associated with the keys on the keyboard. For example, the ASCII value 65 is associated with the capital letter A. We know that if we press the key under our left little finger, and the caps lock is on, the capital letter 'A' will appear on the screen. The computer, of course, does things differently, and I am not going to try and spell all of that out. Suffice to say that, in a program, you can refer to an ASCII value and to a hexidecimal value associated with that ASCII value, and, if you modify the hexidecimal value, you will change the shape that is displayed on the screen.

Confused? Let me try and rephrase that. All of the characters on the keyboard have an ASCII value mapped to that specific key. The actual shape which will be displayed on the screen when you press that key is flexible. A mathematical value 16 hex digits long determines that shape. A default value exists for all of the keys, which gives us our standard character set (font). To make your own character set, you have to redesign each of the characters. This is done by changing the hex value associated with the ASCII number. Of course, you don't have to create a new character set. You might just like to have something a little different than the asterisk to set things off on one of your screen displays. You can redesign the asterisk and use it. Sprites are designed using this same concept, but this article and program don't really get into that.

One of the problems associated with this type of graphics design is that the shape reverts to the default shape just when you don't want it to - there are rules about that! But, under most circumstances, the standard keyboard characters (ASCII values 32 - 127) do not get reloaded during program operation unless you reset the screen with a CALL SETnn command. For this reason, it is sometimes awkward to have used a character from the standard set - sometimes it is needed in its original form but you also need the new design. With MDOS there is an advantage for this circumstance. In TI Basic you could use ASCII codes 32 - 156. In TI Extended Basic, only codes 32 - 143 were available. Using FORTRAN 9640, you can redefine ASCII codes 1 - 255. Which means you have lots of codes to redefine for graphics purposes and still have the standard character set available.

The program listed below is a nice toy. It lets you input ASCII codes and it will show you the hex value as well as displaying the standard shape onto the screen. It will also let you modify the hex value and see the new shape. No provisions have been made to save any of the data, so sharpen up your pencil and write down those hex numbers when you make some neat shapes. You can do a print screen and get the hex values, but your printer will only print the standard ASCII shapes - it doesn't know about your screen redefinition. Maybe someone will take this program idea and turn it into a really good utility for designing sprites or character sets, such as was done in XBasic years ago.

WARNING - locks up if you don't type a number when asked for an ASCII code.

```
      PROGRAM CHARACTER
      IMPLICIT INTEGER(A-Z)
      DOUBLE PRECISION DCHAR
      CHARNUM=65
C
      CALL SET80
      CALL SCREEN(16,2)
C
      WRITE (6,100)
100   FORMAT('+',M4.17,'C H A R A C T E R   R E D E F I N I T I O N')
      WRITE (6,110)
110   FORMAT('+',M7.5,'This utility displays the Hexadecimal character patterns
     1associated',M8.5,'with ASCII character codes.  Several repetitions of the
     2character',M9.5,'[shape] are displayed at the same time.')
      WRITE (6,120)
120   FORMAT('+',M11.5,'You may also modify the pattern of a character and view
     1the new shape.',M12.5,'Several repetitions of the modified character are d
     2isplayed.')
      WRITE (6,130)
130   FORMAT('+',M14.5,'Hard copy of the Hex pattern can be obtained by printing
     1 the screen.',M15.5,'Since the pattern is not downloaded, your printer wil
     2l print the',M16.5,'standard version of the character.')
      CALL CHARPA ( 65, DCHAR )
      WRITE (6,140) CHARNUM,DCHAR
140   FORMAT('+',M18.12,'Example: ',M20.14,'Char',I4,': ',Z16)
      CALL HCHAR ( 21, 42, 65, 16 )
C
      WRITE (6,150)
150   FORMAT('+',M22.10,'Press <Enter> To Continue...')
160   CALL KEY (0,K,S)
      IF ( S .EQ. 0 ) GOTO 150
      IF ( K .NE. 13 ) GOTO 160
C
170   CALL SET80
      WRITE (6,200)
200   FORMAT('+',M5.5,'To See HEX Pattern:',M7.8,'1  Enter ASCII Number And Pres
     1s <Enter>',M8.20,'(Range Is 1 - 255)')
C
      WRITE (6,205)
205   FORMAT('+',M10.5,'')
      READ (6,210) CHARNUM
210   FORMAT(I4)
      IF ((CHARNUM .LT. 1) .OR. (CHARNUM .GT. 255)) GOTO 170
      CALL CHARPA (CHARNUM,DCHAR)
      WRITE (6,220) CHARNUM,DCHAR
220   FORMAT('+',M12.10,'Char:',I4,'    ',Z16)
      CALL HCHAR (13,48,CHARNUM,16)
      CALL HCHAR (16,40,CHARNUM,8)
      CALL HCHAR (23,40,CHARNUM,8)
      CALL VCHAR (16,39,CHARNUM,8)
      CALL VCHAR (16,48,CHARNUM,8)
C
      WRITE (6,230)
230   FORMAT('+',M24.5,'<C>ontinue, <M>odify, <E>xit...')
```

```
240   CALL KEY (0,K,S)
      IF ( S .EQ. 0 ) GOTO 240
      IF ( K .EQ. 67 ) THEN
        GOTO 170
      ELSEIF ( K .EQ. 77 ) THEN
        GOTO 300
      ELSEIF ( K .EQ. 69 ) THEN
        GOTO 1000
      ELSE
        GOTO 240
      ENDIF
C
300   CALL SET80
      WRITE (6,310)
310   FORMAT('+',M1.4,'Character shapes are modified by changing the Hexidecimal
     1 Pattern',M2.4,'values. Enter the ASCII number of the character to be modi
     2fied. After')
      WRITE (6,315)
315   FORMAT('+',M3.4,'the initial display of data, you may modify the Hexidecima
     1al values.',M4.4,'Press <Enter> to see the affect of your changes.')
C
      WRITE (6,318)
318   FORMAT('+',M6.10,'Hex Digits include: 0 1 2 3 4 5 6 7 8 9 A B C D E F')
      WRITE (6,320)
320   FORMAT('+',M8.5,'')
      READ (6,330) CHARNUM
330   FORMAT(I4)
      IF ((CHARNUM .LT. 1) .OR. (CHARNUM .GT. 255)) GOTO 300
      CALL CHARPA (CHARNUM,DCHAR)
      WRITE (6,340) CHARNUM,DCHAR
340   FORMAT('+',M10.10,'Char:',I4,'    ',Z16)
      CALL HCHAR (11,48,CHARNUM,16)
      WRITE (6,350)
350   FORMAT('+',M12.22,'')
      READ (6,360) DCHAR
360   FORMAT(Z16)
      CALL CHAR(CHARNUM,DCHAR)
      CALL HCHAR(13,48,CHARNUM,16)
      WRITE (6,370)
370   FORMAT('+',M22.40,'<C>ontinue, <E>xit... )
C
380   CALL KEY (0,K,S)
      IF ( S .EQ. 0 ) GOTO 380
      IF ( K .EQ. 67 ) THEN
        GOTO 300
      ELSEIF ( K .EQ. 69 ) THEN
        GOTO 1000
      ELSE
        GOTO 380
      ENDIF
1000  END
```

* * * * * * *