

UPSTATE UPSTATE UPSTATE UPSTATE UPSTATE UPSTATE UPSTATE UPSTATE UPSTATE UPSTATE

OUR NEXT MEETING will be on Thursday

THE MAY Meeting will be

APRIL 17, 1986 at 7:30 pm

MAY 15, 1986 at 7:30 pm

PLACE: CAPITAL DISTRICT PSYCHIATRIC CENTER  
New Scotland Ave. Next to Albany Medical Center

The program for the APRIL meeting is as follows:  
Robert Katt will explore the inner secrets of TI-WRITER.  
A presentation of sort algorithms using TI Basic.  
John Chera will demonstrate the GE Printer.  
The Software Library Club will be present.

A NOTE to other Users Groups: The articles printed in the Upstate Newsletter may be reprinted if proper credit is given to the author and to the Upstate New York 99/4 Users Group.

ALBANY TEXAS INSTRUMENTS BULLETIN BOARD SERVICE (TIBBS)-out of service

UPSTATE 99/4A USERS GROUP  
P.O. BOX 13522  
ALBANY, N.Y. 12212

---

---

---

VOL IV, NO. 3

APRIL 1986

Allison Smith, EDITOR 439-4860

## MYARC's Extended BASIC II

By J. Feter Hoddie

Note: For a complete description of XBII please see the article I wrote after the 1985 TI Faire in Chicago. The following article is NOT a review but a look at this exciting new product, as distributed in February, 1986.

The purpose of this article is to explain the new Extended BASIC II language from MYARC. Please note that I did not use the word cartridge. Extended BASIC II (XBII) is much more than just a cartridge and, in fact, much more than just a language. To run XBII you need a MYARC 128K or 512K Memory Expansion card with a special XBII eeprom in it. If you currently own a MYARC card you will need a new eeprom which will be provided when you purchase XBII. You also get a disk and a cartridge. To run XBII you need all three pieces. The disk contains a series of files which make up the over 48K of assembly language code that make up XBII. The eeprom contains another 8K (I believe) and the cartridge contains nothing at all. That is not to say that you don't need the cartridge. The cartridge has 8K of RAM in it. What happens is that when you go to the title screen with the XBII cartridge in place, the Myarc memory expansion card writes the contents of the eeprom out to the cartridge. This happens in a blink of an eye so you never even know it happened. Now I'm not 100% sure why MYARC chose to put XBII together this way but I suspect it was to make upgrades easier. As you will find out, if you read on, XBII is not a completed product, and at least one more update will be required. By making the cartridge 'soft' MYARC only has to change the eeprom on the memory expansion card and the disk. They don't have to worry about the cartridge. Thus only two things to worry about instead of three. You may ask, if the cartridge is RAM, why do I need it at all? Why couldn't I just run XBII out of the Memory expansion card? The answer, as near as I can tell, is that you can't execute assembly code that is mapped into the cartridge space (>6000 to >7FFF) out of the PE box. But I'm not sure of this. It may be that Myarc needed that extra 8K of RAM as RAM and not ROM where they couldn't store data. I really don't know, this is only speculation. But now that I've explained the hardware aspect of XBII, I will now get into its features.

XBII is supposed to be 100% compatible with TI's Extended BASIC. You should be able to take any XB program you've written, load it into XBII and watch it run. This does work in many cases, however because XBII is not yet finished; it fails in just as many cases. XBII does not yet support DEF statements, user defined CALL/SUB statements, the MIN and MAX functions return erroneous data sometimes, you can not pass variables to assembly language in CALL LINK statements, and once in a great while the language will just lock up for no good reason. Now that is ALL the bad news. All of these problems will be fixed in the near future. Quite frankly, however, I am VERY impressed with XBII. It is a great product and has a lot of potential.

The most notable feature of XBII is its graphics capabilities. In regular TI graphics mode (what you get in TI Extended BASIC) you now can define all 256 characters, use all 32 sprites, and define all color sets. This means that programs that worked in TI BASIC but would not run in TI Extended BASIC will work in MYARC XBII, not to mention the extra added characters beyond even what TI BASIC supplies. In graphics mode 2 you get text mode. True 40 columns. And the PRINT, DISPLAY AT, ACCEPT AT, and the rest of the screen display commands still work. You can even edit your program in 40 columns. This is really great. You can see so much more of what is going on while you are programming. Finally there is graphics mode 3 which is bit map. You can access every pixel on the screen individually. You can draw lines, points, circles, rectangles, check the color of a pixel, write text horizontally or vertically, and fill with a color, all using simple XBII CALL statements. Furthermore, although the documentation says it doesn't support it, you get automation of sprites in bit map mode, something that just isn't that easy to do. A future upgrade of XBII will allow you to fill with a character pattern as well as a color.

```

FORTH
FORTH
FORTH SCR #60
FORTH 0 ( TEXTPECT James H Posniewski 30Apr85 )
FORTH 1 HEX
FORTH 2 70 USER SPAN ( # of characters received by TEXTPECT )
FORTH 3 ; TEXTPECT ( addr count -- ) >R 0 0 SPAN !
FORTH 4 BEGIN KEY DUP OD ( cr ) = 0= WHILE
FORTH 5 ( while not a cr... )
FORTH 6 CASE OB OF DUP 0= IF 7 EMIT
FORTH 7 ELSE -1 CURPOS +! 1- ENDIF ENDOF
FORTH 8 09 OF DUP SPAN = IF 7 EMIT
FORTH 9 ELSE 1 CURPOS +! 1+ ENDIF ENDOF
FORTH 10 03 OF DUP SPAN = IF 7 EMIT
FORTH 11 ELSE -1 SPAN +! SPAN OVER - >R OVER OVER + DUP
FORTH 12 1+ OVER R CMOVE R> DISP. ENDIF ENDOF
FORTH 13 04 OF R SPAN = IF 7 EMIT
FORTH 14 ELSE 1 SPAN +! SPAN OVER - >R OVER OVER + DUP
FORTH 15 DUP 1+ R <CMOVE BL OVER C! R> DISP. ENDIF ENDOF -->

```

```

FORTH SCR #61
FORTH 0 ( TEXTPECT continued James H Posniewski 30Apr85 )
FORTH 1 OVER R =
FORTH 2 OVER 20 ( OR OVER 7E ) OR
FORTH 3 IF 7 EMIT
FORTH 4 ELSE OVER SPAN = IF 1 SPAN +! ENDIF
FORTH 5 >R OVER OVER + R>
FORTH 6 DUP EMIT SWAP C! 1+
FORTH 7 0 ( for endcase to drop )
FORTH 8 ENDIF
FORTH 9 ENDCASE
FORTH 0 REPEAT
FORTH 11 R> DROP DROP SPAN SWAP - CURPOS +! BL EMIT
FORTH 12 SPAN +
FORTH 13 0 SWAP OVER OVER C! 1+ C! ; ( store two nulls )
FORTH 14
FORTH 15 R->BASE

```

BUT THAT'S NOT ALL...

Wait! You aren't finished yet! now, it's necessary to install TEXTPECT in the place of EXPECT in QUERY. The FORTH interpreter uses QUERY to get your input. Do the following:

```

; TEST TIB 80 TEXTPECT 0 IN ! ; ( load TEXTPECT first! )
' QUERY 10 - 20 DUMP ( you must first load DUMP )

```

you will get output somewhat like this:

```

ACCC2 xxxx xxxx xxxx xxxx xxxxxxxx \
ACCA xx51 5545 52D9 8334 xQUER..4 | NOTE: These numbers
ACD2 A6C0 A574 A03E 0050 ...t.>.P | may differ!!
ACDA>>AC52<<A66B AB10 A592 .R.h... / xxxx means i don't care!

```

now, type

```

' TEST 10 - 20 DUMP

```

you will get something like the above as well. Now, examine the two lists. There should be one difference (besides the name) it should be the word with ">" and "<" signs around it. Take the number you find when you DUMPed TEST and poke it into the position in QUERY. In this case, xxxx (the number different) ACDA ! VOILA! Now, you should be able to cursor left and not delete characters.

TIPS FROM THE TIGERCUB

833

Copyright 1986

TIGERCUB SOFTWARE  
136 Collingwood Ave.  
Columbus, OH 43213

Distributed by Tigercub Software to T1-99/4A Users Groups for promotional purposes and in exchange for their newsletters. May be reprinted by non-profit users groups, with credit to Tigercub Software.

Over 130 original programs in Basic and Extended Basic, available on cassette or disk, only \$3.00 each plus \$1.50 per order for PPH. Entertainment, education, programmer's utilities. Descriptive catalog \$1.00, deductible from your first order.

Tips from the Tigercub, a full disk containing the complete contents of this newsletter Nos. 1 through 14, 50 original programs and files, just \$15 postpaid.

Tips from the Tigercub Vol. 2, another diskfull, complete contents of Nos. 15 through 24, over 60 files and programs, also just \$15 postpaid. Or, both for \$27 postpaid.

Muts & Bolts (No. 1), a full disk of 100 Extended Basic utility subprograms in merge format, ready to merge into your own programs. Plus the Tigercub Menuloader, a tutorial on using subprograms, and 5 pages of documentation with an example of the use of each subprogram. All for just \$19.95 postpaid.

Muts & Bolts No. 2, another full disk of 100 utility subprograms in merge format, all new and fully compatible with the last, and with 10 pages of documentation and examples. Also \$19.95

postpaid, or both Muts Bolts disks for \$37 postpaid.

Tigercub Full Disk Collections, just \$12 postpaid! Each of these contains either 5 or 6 of my regular 93 catalog programs, and the remaining disk space has been filled with some of the best public domain programs of the same category. I am NOT selling public domain programs - my own programs on these disks are greatly discounted from their usual price, and the public domain is a FREE bonus! TIGERCUB'S BEST PROGRAMMING TUTOR PROGRAMMER'S UTILITIES BRAIN GAMES BRRIN TEASERS BRAIN BUSTERS! MANEUVERING GAMES ACTION GAMES REFLEX AND CONCENTRATION TWO-PLAYER GAMES KID'S GAMES MOPE GAMES WORD GAMES ELEMENTARY MATH VOCABULARY AND PEADING MUSICAL EDUCATION KALEIDOSCOPIES AND DISPLAYS

For descriptions of these send a dollar for my catalog!

I found a bug in Muts & Bolts #2 which prevents using HIGHCHAR after HEAVYCHAR. To fix it, remove the write-protect tab, MERGE OSK1.HEAVYCHAR RES 21000,1 SAVE OSK1.HEAVYCHAR, MERGE Replace write-protect tab.

While they last, and the supply is limited, I will sell a single Texas Instr. cassette interface cable for \$2.00 with any order for cassette software.

Did you ever wonder how a computer sort actually worked? This program will let you actually see it in

action. It will also show you the value being held in the temporary variable T\$, and the total number of swaps and comparisons made.

Then you can change any of the variables and resort. Try AAA in the last position or ZZZ in the first. You will find that some of the fastest sorts are not so fast when a list is already almost in sequence.

```
100 CALL CLEAR :: CALL SCREE
N(16):: FOR SET=2 TO 9 :: CA
LL COLOR(SET),5,16):: NEXT SE
T :: ON WARNING NEXT :: RAND
OMITE
```

```
110 DISPLAY AT(21,1)ERASE AL
L:)):TIGERCUB SORT WATCHER(
<<: "Wait, please - genera
ting a random array..." ::
DIM A$(101),B$(1::),ST(25,2)
120 FOR J=1 TO 100 :: FOR L=
1 TO 3 :: B$(J)=B$(J)+CHR$(1
NT(2+RND*65)):: NEXT L :: X
=RJ :: A$(2)=B$(X):: GOSUB 32
767 :: NEXT J
```

```
130 DISPLAY AT(3,1)ERASE ALL
:"(1) BUBBLE SORT:" :: "(2) SH
AKER SORT:" :: "(3) SWAP SORT"
:: "(4) SHUTTLE SORT:" :: "(5)
EASY SORT"
140 DISPLAY AT(13,1)::"(6) QU
ICK SORT:" :: "(7) RESORT SORT
": :: "(8) SHELL SORT:" :: "(9)
RESERVED": :: "Type number of
choice"
```

```
150 ACCEPT AT(21,23)VALIDATE
(DIGIT)$I$(2)BELP:K :: IF K
<1 OR K>10 THEN 150
160 DISPLAY AT(24,1)::"Size o
f array? (10-100)" :: ACCEPT
AT(24,25)VALIDATE(DIGIT)$I$(
E(3):G :: IF G<1 OR G>100 TH
EN 160
170 ON K GOSUB 230,300,430,5
00,550,650,850,910,2500 ::
DISPLAY AT(22,1)::M;"SWAPS":C
"COMPARISONS" :: C,M#
```

```
180 DISPLAY AT(24,1)::"Choose
(1)Menu or (2)Resort" :: AC
CEPT AT(24,7)VALIDATE(I$)S
I$(E(1):G :: IF G=1 THEN 130
190 DISPLAY AT(24,1)::"Change
which position? B" :: ACCEP
T AT(24,24)VALIDATE(DIGIT)$I
$(E(1)-3):P :: IF P=0 THEN 210
ELSE IF P<1 OR P>6 THEN 190
```

```
200 DISPLAY AT(24,1)::"Change
to" :: ACCEPT AT(24,12)SIZ
E(3):A$(P):: W=P :: GOSUB 10
20 :: GOTO 190
210 DISPLAY AT(22,1):: " " ::
" " :: GOSUB 1010 :: M=G :: ON
K GOSUB 200,310,440,510,560,
660,660,920,2500 :: DISPLAY
AT(22,1)::M;"SWAPS":C;"COMPA
RISONS" :: C,M# :: GOTO 100
220 REM #BUBBLESORT#
```

```
230 CALL CLEAR :: GOSUB 900
240 FOR J=2 TO N :: C=C+1 ::
IF A$(J)>A$(J-1)THEN 260
250 T=A$(J):: GOSUB 1050 ::
A$(J)=A$(J-1):: T=T+1 :: GOSU
B 1020 :: A$(J-1)=T :: T=T-
1 :: GOSUB 1020 :: M=M+1 ::
F=1
```

```
260 NEXT J :: C=C+1 :: IF F=
0 THEN 200
270 M=M+1 :: F=0 :: M=M+1 ::
M=M-1 :: GOTO 240
280 RETURN
290 REM #SHAKERSORT#
```

```
300 CALL CLEAR :: GOSUB 900
310 M=M+1 :: L=1 :: M=M+1 ::
R=M
320 M=M+1 :: F=0 :: FOR J=L
TO R-1 :: C=C+1 :: IF A$(J)<
A$(J+1)THEN 340
330 T=A$(J):: GOSUB 1050 ::
A$(J)=A$(J+1):: T=T+1 :: GOSU
B 1020 :: A$(J+1)=T :: T=T+
1 :: GOSUB 1020 :: M=M+1 ::
F=1
```

```
340 NEXT J :: C=C+1 :: IF F=
0 THEN 410
350 M=M+1 :: R=R-1 :: C=C+1
:: IF R=L THEN 410
360 M=M+1 :: F=0 :: FOR J=R
TO L STEP -1 :: C=C+1 :: I
F A$(J)>A$(J-1)THEN 380
370 T=A$(J):: GOSUB 1050 ::
A$(J)=A$(J-1):: T=T+1 :: GOSU
B 1020 :: A$(J-1)=T :: T=T-
1 :: GOSUB 1020 :: M=M+1 ::
F=1
```

```
380 NEXT J :: C=C+1 :: IF F=
0 THEN 410
390 M=M+1 :: L=L+1 :: C=C+1
:: IF L=R THEN 410
400 GOTO 320
410 RETURN
420 REM #SWAPSORT#
```

```
430 CALL CLEAR :: GOSUB 900
440 FOR J=1 TO N-1 :: M=M+1
:: R=J :: FOR JJ=J+1 TO N ::
C=C+1 :: IF A$(R)<A$(JJ)IH
EN 460
```

```

450 M=M+1 : R=J
460 NEXT JJ : C=C+1 : IF R
  WJ THEN 480
470 T=AR(J) : GOSUB 1850 :
  AR(J)=AR(R) : X=X+1 : GOSUB
1820 : A=AR(T) : I=I+R : I=I+6
480 NEXT J : RETURN
490 REM *****SHUTTLE SORT*****
500 CALL CLEAR : GOSUB 980
510 FOR J=1 TO N-1 : FOR JJ
  WJ TO I STEP -1 : C=C+1 :
  IF AR(JJ)<AR(J+1) THEN 530
  : T=AR(JJ) : GOSUB 1850 :
  : AR(JJ)=AR(J+1) : X=X+1 : GOSUB
1820
520 AR(JJ+1)=T : X=JJ+1 :
  GOSUB 1820 : NEXT JJ
530 NEXT J : RETURN
540 REM *****EASY SORT*****
550 CALL CLEAR : GOSUB 980
560 M=M+1 : D=1
570 M=M+1 : D=2*8 : C=C+1
  : IF D=C THEN 570
580 M=M+1 : D=INT(D/2) : C=C
+1 : IF D=D THEN 630
590 FOR J=1 TO N-D : M=M+1
  : Y=J
600 M=M+1 : Z=Y+D : C=C+1
  : IF AR(Y)<AR(Z) THEN 620 :
  : T=AR(Y) : GOSUB 1850 : A
  Y=Y+AR(Z) : Y=Y : GOSUB 19
20 : AR(Z)=T : X=X+1 : GOS
UB 1820
610 M=M+1 : Y=Y-D : C=C+1
  : IF Y>0 THEN 600
620 NEXT J : GOTO 580
630 RETURN
640 REM *****QUICKSORT*****
650 CALL CLEAR : GOSUB 980
660 M=M+1 : L=1 : M=M+1 :
  R=N : M=M+1 : T=0
670 T=AR(INT((L+R)/2)) : GOS
SUB 1850 : M=M+1 : J=L :
M=M+1 : JJ=R
680 C=C+1 : IF AR(J)>T THEN
  710
690 M=M+1 : J=J+1
700 GOTO 680
710 C=C+1 : IF AR(JJ)<T THEN
  730
720 M=M+1 : JJ=JJ-1 : GOTO
710
730 C=C+1 : IF AR(JJ)<AR(JJ
) THEN 760
740 C=C+1 : IF JJ=J THEN 7
60
750 M=M+1 : J=J+1 : GOTO 7
30
760 C=C+1 : IF JJ=J THEN 7

```

```

80
770 M=M+1 : H=AR(J) : AR(J
)=AR(JJ) : X=X+1 : GOSUB 1820
  : AR(JJ)=H : X=JJ : GOS
UB 1820 : GOTO 680
780 M=M+1 : J=J+1 : M=M+1
  : JJ=JJ-1 : C=C+1 : IF JJ
  =R THEN 800
790 M=M+1 : T=T+1 : M=M+1
  : ST(I,0)=J : M=M+1 : ST(
  T,1)=R
800 M=M+1 : R=JJ : C=C+1 :
  : IF L<R THEN 670
810 C=C+1 : IF T=0 THEN 830
820 M=M+1 : L=ST(I,0) : M=M
+1 : R=ST(I,1) : M=M+1 : T
  =T+1 : GOTO 670
830 RETURN
840 REM *****RESORT SORT*****
850 CALL CLEAR : GOSUB 980
860 FOR J=2 TO N : C=C+1 :
  IF AR(J)>AR(J-1) THEN 900
870 T=AR(J) : GOSUB 1850 :
  FOR L=J-1 TO I STEP -1 : A
  (L+1)=AR(L) : X=L+1 : GOSU
B 1820
880 C=C+1 : IF AR(L-1)>T
  THEN 890 : AR(L)=T : X=L
  : GOSUB 1820 : GOTO 900
890 NEXT L
900 NEXT J : RETURN
910 REM *****SHELLSORT*****
920 CALL CLEAR : GOSUB 980
930 M=M+1 : M=N
940 M=M+1 : M=INT(M/3)+1
950 FOR J=1 TO N-M : FOR JJ
  WJ TO I STEP -M : C=C+1 :
  IF AR(JJ)<AR(JJ+M) THEN 970
  : T=AR(JJ) : GOSUB 1850
960 AR(JJ)=AR(JJ+M) : J=JJ :
  GOSUB 1820 : AR(JJ+M)=T
  : X=JJ+M : GOSUB 1820 : N
  EXT JJ
970 NEXT J : C=C+1 : IF M>
  1 THEN 940 : RETURN
980 REM *****AREM ARRAY*****
990 FOR J=1 TO G : AR(J)=0
  (J) : J=J : M=AR(J) : GOSU
B 1820
1000 NEXT J : M=6
1010 DISPLAY AT(24,1) "A to
  abort P to pause" : RETUR
N
1020 RR=X
1030 IF RR>20 THEN RR=RR-20
  : GOTO 1030
1040 CC=1-(X/20)*5-(X/40)*5-
  (X/60)*5-(X/80)*5 : DISPLAY
  AT(RR,C):AR(I) : M=M+1 :
  GOSUB 1860 : RETURN

```

```

1050 DISPLAY AT(22,14) "T4="
  : T4 : M=M+1 : GOSUB 1860 :
  : RETURN
1060 CALL KEY(1,K1,S) : IF
  S=0 THEN 1090
1070 IF K1=65 THEN 130
1080 CALL KEY(1,K2,S) : IF
  S<1 THEN 1080
1090 RETURN

Don't try timing these
sorts, because the screen
display distorts the speed.
Option 9 has been left open
so that you can add your own
favorite sort routine, in
the same format, starting in
line 25800.

These routines may not be
the most efficient foras,
and their names may not be
correct. If you know better
ones, let us know!

100 'BASKET WEAVING by Jim P
eterson
110 CALL CLEAR : M=1 : T=
2 : CH="ASASASASASASAS5FF
  #FF#18FF#FF" : CALL CHAP
  (142,CH) : CALL COLOR(14,2,
  M,13,2,M) : CALL SCREEN(M)
120 CALL HCHAR(1,1,143,760)
  : CALL CHAR(134,CH) : CH=14
  2
130 FOR C=1 TO 31 STEP T :
  FOR R=1 TO 23 STEP T : CALL
  HCHAR(R,C,CH) : NEXT R : F
  OR R=24 TO 2 STEP -T : CALL
  HCHAR(R,C,CH) : NEXT R :
  NEXT C
140 CH=ABS((CH+142)*135*(CH+
  134)*143) : RANDOMIZE : T=I
  NT(3*RDND+2)
150 FOR R=1 TO 23 STEP T :
  FOR C=2 TO 32 STEP T : CALL
  HCHAR(R,C,CH) : NEXT C
160 FOR C=31 TO 1 STEP -T :
  CALL HCHAR(R,C,CH) : NEXT
  C : NEXT R : CH=CH-1 : M=
  INT(14*RDND+3) : T=INT(3*RD
  ND+2)
170 IF CH=134 THEN CALL COLD
  R(13,2,M) : GOTO 130 ELSE CA
  LL COLD R(14,2,M) : GOTO 130

```

```

bottom. Arrow keys can then
be used to create a line
graph of asterisks or what-
ever, annotated with text as
desired.
180 OPEN B1:"OSKI.GRAPHPAGE"
  : OUTPUT #1 PRINT B1:TAB(4);R
  P1(" ",75) : FOR J=1 TO 57
  : J=57*J
185 IF J<10 THEN J=" " * 6J
110 PRINT B1:J;R;R1(" ",30
  16" " : NEXT J
120 FOR T=1 TO 2 : PRINT B1
  : " : FOR J=1 TO 77 : J1
  =57*J1) " : PRINT B1:56
  (126,T,1) : NEXT J : PRINT
  B1 : NEXT T : CLOSE B1

1 'TO PRINT A HANDY REFERENC
  E CHART OF ASCII TO HEX CODE
  - MODIFIED FROM READING-BERK
  S AUG 85
98 OPEN B1:"PIO" : PRINT B1
  :CHR$(27);CHR$(77);CHR$(5)
100 FOR I=32 TO 63 : FOR Y=
  X TO X+64 STEP 32 : CALL CH
  ARPAT(Y,Y) : PRINT B1;Y "
  :CHR$(Y) " ; Y; : NEXT Y :
  PRINT B1:" " : NEXT X

100 CALL CLEAR : CALL MAGNIF
  Y(2) : RANDOMIZE : BISPLAY
  AT(3,2) "TIGERCUB SPEED TYP
  ING TEST" : TAB(12) "SPEED"
  : T=10
110 DISPLAY AT(5,10) : I=0-T
  : X=INT(26*RDND+65) : CALL SP
  RITE(0,1,X,2,96,128) : FOR D=
  1 TO T : CALL KEY(1,K,S) :
  ON (K+1)+2 GOTO 120,130
120 T=T-1 : GOTO 110
130 NEXT B : T=T+1 : GOTO
110

The UG newsletters are
full of good editorials,
remining people that they
had better pay for their
freeware or there won't be
any more. I totally agree
with that - but I can't help
thinking that if there had
been as much emphasis on
paying for commercial
software instead of pirating
it, there would still be a
lot more good programmers
supporting the TI!

```

MEMORY FULL  
Jim Peterson



Another of the exciting features of XBII is the CALL MARGINS command. This CALL lets you set up windows on your screen. You just do a CALL MARGINS and give the screen boundaries you want to use (example: CALL MARGINS(10,20,1,20) would set up a window using rows 10 through 20 and columns 1 through 20). All statements which access the screen in graphics modes 1 and 2 then act only on that window. You can even do a CALL MARGINS when entering a program if your TV cuts off part of the picture. This command makes adding status lines, help areas, and all sorts of neat program features a breeze.

XBII also has CALL PEEKV and CALL POKEV commands to let you directly access memory in VDP (screen) memory. Previously this was only available in TI BASIC with the Editor/Assembler cartridge in place. There is a new VALHEX function which does hexadecimal to decimal conversions. For example "A=VALHEX("FF") would set A equal to 255. There is a FREESPACE variable which returns how many bytes of space are left but right now it always returns a zero. You could use it to check if your program is running out of memory like: IF FREESPACE<100 THEN PRINT "Running low on space . . .".

One pleasant surprise is that the LIST command is now at least 50% faster. Listings zoom by at lightning speed. You can pause the listing with the space bar but wow!!

Now the biggest and best feature of XBII was supposed to be that it was FASTER than TI's XB. It is. But not that much. But there is a VERY good reason for this. XBII is supposed to support integer variables. This means that the variable is only capable of storing numbers from -32767 to +32768 and no decimal values. Furthermore integer variables take up only 2 bytes of memory whereas floating point (regular) numerical variables take up 8 bytes. Also the computer can understand integer variables easily since that is what it uses internally. Thus integer variables are faster and more efficient. Unfortunately they have not yet been implemented. When they are, that's when I think XBII will really shine. Then it will really be fast.

Now another thing people expect from XBII is the ability to write longer programs. They figure if you got a 128K card in the PE box then you should be able to use some of that, right? The answer is yes . . . sort of. XBII gives you 24K of program space. If that sounds like all you got with TI XB, you're right. Except that in TI XB that 24K also had to hold all numerical variables and all sorts of other data. IN XBII it holds ONLY program. You have an additional 24K for numeric variables and another 24K for string variables. So with a little creative programming you can fit TONS of stuff in there. You can also (although it doesn't seem to work in this version) define how much assembly space you want by doing a CALL INIT with a byte count. So to reserve 9000 bytes of assembly space you do a CALL INIT(9000). When you do a SIZE command, you are told how many bytes of PROGRAM, STRING, and VARIABLE space you have left.

XBII comes with a very complete manual. It is more of a reference guide than a tutorial but it is very clear. It is very similar to TI's manual for their XB except that the new commands are listed and additions to existing commands are explained. The examples are clear, complete, and easy to follow. The only flaw is that the manual doesn't list which commands don't work. I guess this means that they will be finished real soon now.

That just about covers my first impressions of XBII. Overall I think MYARC has done an excellent job on this product although it is definitely not finished. The good news is that it will be very soon as the story goes that this is almost identical to the version of BASIC that will be in the new computer so it has to be finished soon. It is a great programming environment. I wrote a draw program in bit map mode complete with fill, circle, line, box, and color commands in about 2 hours. Small things have been considered. You can now say RUN A\$ where A\$ is the name of the program you want to RUN. You can say OLD FRANK and XBII will look for a file on DSK1 called FRANK. There has been a great deal of attention to detail and quality in XBII, its just a matter of finishing it up now. XBII is here FINALLY and it looks like it'll do everything MYARC promised.

## PRESIDENT'S COMPUTER CORNER

One of the functions that a computer can be called on to do is SORT. However, there are several methods or algorithms that could be implemented on a computer to perform a sort. For April, all five sort techniques will be demonstrated and compared. This time I will bring a backup diskette for the program. As a remainder the techniques are:

- (1) Selection Sort - April
- (2) Bubble Sort - April
- (3) Heap Sort - April
- (4) Shell sort - April
- (5) Quick Sort - April

At the April meeting, all 5 sorts will be compared for speed relative to the number of records to be sorted.

Also for April, two rookie speakers will appear: John Chera with his GE Printer and Bob Katt with tips on TI-WRITER. The Software Demonstration Club will also be on hand.

A new feature with this newsletter, Shannon Posniewski will be contributing articles under the title, THE PROCESSOR. Please read his FORTH article.

### FOR SALE:

Still available - Multiplan at \$30.00  
Many TI modules at \$5 and up.  
Call Nick at 372-1178 or see Art at meeting.

If you wish to advertise any TI items for sale talk to me at the meeting or write to the editor.

Arthur F. Payeur