



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

THE HUGgers
HOOSIER USERS GROUP
People Helping People

JULY 1992

The HUGgers Newsletter

Volume 11, Number 7

OFFICER'S CORNER

Finding things to write about for this column is sometimes rather difficult, fortunately this month there are several items of interest!

MYARC CORCOMP REPAIRS

First of all, the most recent issue of MICROpendium had a very interesting article concerning MYARC. It seems that Lou Phillips has finally closed Myarc down. However, there is some good news in that Cecure Electronics of Franklin, Wisconsin is in the process of negotiating with Lou Phillips to become an authorized Myarc repair center; and if the proper agreements can be reached may even produce some of Myarc's products. Cecure Electronics is owned by Don Walden, is president of the Milwaukee Users Group and vice president of the Chicago TI Users Group.

Along similar lines, in reading through recent messages on the Delphi on-line information service's TI Forum, I noticed two addresses of companies that do repair work on CorComp products. These are:

89 Computer Repair
2101 West Crescent Ave, Unit B
Anaheim, CA 92801
(718) 539-4834

International Diversified
Technologies, Inc.
2211 E. Winston Road, Suite G
Anaheim, CA 92806
(714) 956-4450

H.U.G. BBS

Sometime in the next few weeks, our BBs will be getting a new look. We have purchased the S&T BBS bulletin board software by Tim Tesch of S&T Software, Ltd. The new software is being customized to suit our specific needs and will support true ADM3A/ANSI graphics, on-line games, can have as many as 20 sub-directories, support Y-Modem protocol, 80-column users, will be menu driven, and include remote file handling by the sysops.

NEW IN OUR LIBRARY

It has been awhile since any really interesting new games or educational programs have surfaced on the networks such as Delphi or GENie. A new baseball game and a very slick educational astronomy program have recently appeared. These are by TI's John Phillips and are called: Baseball, Stargazer2 and Stargazer3. These are assembly language (EA5) programs and apparently were intended to be commercially released. The source code for all three is available, although I have not yet downloaded it. These programs will be added to our library.

ANNUAL HUGGER'S PICNIC

Bill Lucid is working on plans for our annual picnic. It is tentatively set for August 16, at Eagle Creek Park, and will be a pitch-in affair. This will be discussed further at this month's meeting. Any

(CONTINUED ON PAGE 9)

The Library Corner:
by Dan H. Eicher

The TI community has a great friend in Bruce Harrison. Not only has he produced many unique and innovative programs for us, he has constantly supported the programming efforts of others. In this installment of the Library Corner, I will review two disk of software and tutorials by Bruce.

If you own a TI and don't take Micropendium, you should. For the past year, Bruce has been writing a monthly assembly language tutorial for beginners and experts in Micropendium. On the first disk to be reviewed, he has gathered the text of his first eight tutorials. Also, included on this disk are "sidebars", sidebars are the assembly language source code that accompany his articles.

For anyone interested in learning assembly language, the best way to learn to write assembly language programs is by doing and study others source code. Bruce writes in a style that is easy to understand without being patronizing. Of the many assembly tutorials I have read (I have even written a couple), Bruce is by far one of the best at making this, at times, arcane subject matter understandable!

Disk number two, is a group of assembler subroutines that can be accessed from extended basic with CALL statements. While many assembler utilities demonstrate how to write a routine, they are small and of limited usefulness. This is not the case with this group of utilities. These utilities are used in Harrison's Software "Code Breaker" game!

Here is an extract from the ample documentation describing the utilities included and a bit about their use:

"On this disk there are six Assembly Language utilities for use with Extended Basic programs. Each is supplied as a D/F 80 object file suitable for merging into an XB program with ALSAVE (also supplied). The six object files are:

- DEVICE - places device name in string variable.
- NUMACC - special ACCEPT AT for numeric variables.
- STRACC - special ACCEPT AT for string variables.
- NOTEASG - assigns note values to an array variable.
- TRACK4 - replaces DSK1 in XB program.
- TRUINT - gives the integer part of a numeric variable.

Along with these are the source code files, each of which has an "S" added to the object file's name. (e. g. TRUINTS is the source for TRUINT.)

There is also an XB program which incorporates and demonstrates each utility. DEMONUM demonstrates NUMACC, for example. The only complicated situation is the demonstration for TRACK4, which has a menu generator called TESTMEN, and four short programs called PROG1, PROG2, etc.

There are also two short XB programs to allow users who don't have E/A or TIW to print out this document and the source files. The one to print this is called PRINTHOW, and the one to print source code files is called PRINTSRC."

The source code is provided so you can learn from the examples and modify the programs to more closely match your needs. Full documentation on how to add these subroutines to your program is given along with the syntax for their usage. For anyone interested in adding assembler subroutines to their extended basic program this is a - "must have disk".

While Bruce Harrison has released these programs as freeware, I am sure a note explaining how you have found these program to be useful would be appreciated. Bruce is one of a kind. He really goes the extra mile to help people. A good example is this utility disk. We purchased it from him in Chicago at a cost that would barely cover the cost of the diskette it was written on. About a month latter free of charge he sent the TI HUGGERS his compendium of assembler tutorials. About a month after that he sent us, free of charge again, an update to fix a minor bug he found with the utility.

References:

Harrison Software
5705 40th Place
Hyattsville, MD 20781

MICROpendium
PO BOX 1343
Round Rock, TX 78680
One Years Subscription: 25 dollars.

(Reprint from newsletter from:
"The Windy City Times" issue July 92)

Chicago Users Group presents the 10th
annual TI INT'l World Faire for TI 99
/4A

HOLIDAY INN ELK GROVE
1000 BUSSE ROAD (ROUTE 83)
ELK GROVE VILLAGE, ILL

EASY ACCESS FROM I-90 AND
O-HARE AIRPORT

Exhibits and Siminares Start on
Sat OCT 31 92 at 9:am to 5 Pm
Admission cost \$4.00 Social Mixer-
Friday Oct 30 12Am Cost \$5.00 and
Banquet: Sat Oct 31 at 7:30-9:30Pm
Cost \$15.00 ea.

MORE INFROMATION: Write to:

Hal Shanafield, Jr.
2515 Marcy Ave
Evanston, Ill
60201-1111

Call Tel#1/708-8648644

Keep TI-ing!

In our search for articles from our members, about what interests them, we hear from Ricky Bottoms.

He has modified one of his Systems by putting memory expansion IN the Console.

He says, "The main reason I did this was because I am wanting to learn as much about the TI and associated electronics as I possibly can. Secondly, I am wanting to upgrade my system as much as I can by myself and I am wanting to make this particular unit more portable so that it is easier to bring to club meetings and to be able to use it in a variety of situations."

He goes on to say, "I am glad to be in the Hoosier User Group, because I have already learned a lot more than I knew at this time last year, but I know that their is much more to do and learn. I hope the HUG can keep on TI-ing and that I can give and receive knowhow about the TI in paticular and Computers in general."

Ricky adds a PS where he says, "The memory expansion will not allow the Myarc Randisk to work, so I can not use them together."

I would like to add that Ricky attends our meetings regularly and he drives up from Shelbyville.

BOB

Sorry I Goofed again

We recieved a letter and disket from Jim Peterson, reminding us that we didn't finish his series on making music the easy way. We printed part 1 in March and part 2 in April, so we will do our best to get the rest in coming issues.

BOB

PROGRAMMING MUSIC THE EASY WAY

Part 3

by Jim Peterson

In Part 1 of this series, I showed you the simple routine to set up a musical scale, and showed you how easy it was to merge in various routines to create different effects in single-note music. In Part 2 I showed you how to key in single-note melodies from sheet music. Now, we will get into 3-part harmony.

But first, there are a few more things I should have told you about reading music. You will often see curved lines arching over two or more notes. If the notes are not all the same, ignore those lines - they call for phrasing which you cannot really accomplish. But, if the line curves over two or three of the same note, you will get a better effect if you add all their duration values together and program them as a single note. For instance, if your chart gives a whole note a value of 8 and a half-note a value of 4, and the music has a curved line over a whole note followed by a half-note, just program one note with a duration of 12.

You may find a heavy black bar at the beginning of a measure, with a colon to its right, and somewhere later in the music will be a heavy bar with a colon at its left. This means that the notes between those bars are to be played through twice - and naturally you will want to save time by programming them in a GOSUB as I showed you in Part 2. It can get more complicated than that, but generally you can follow the lyrics to decipher what to do.

Rather rarely, you may find three

notes, usually joined together, with a 3 above them. These are called a triplet, and all three of them are to be played, with the same duration for each, in the length of time it would normally take to play one of them. These can create a problem under any method of music programming. The best method is to divide the duration of the note by three and write individual CALL SOUNDS in your music, rather than a GOSUB to a routine, to handle those notes.

Now, let's get on to 3-part harmony. It is just the same as keying in single note music, except that you must also give frequency values to B and C - and, as before, you have to give those values only when they change.

So, load the SCALE routine from the first lesson, and key in this bit of music to experiment with. Notice that I found three repeating phrases and put them in subroutines in 500, 600 and 700 to make this shorter.

```
110 GOSUB 500 :: T=4 :: A=15
    :: B=11 :: C=9 :: GOSUB 100
0 :: T=8 :: A=18 :: GOSUB 10
00 :: T=2 :: A,B,C=0 :: GOSU
B 1000 :: T=2 :: A=23 :: B=1
8 :: C=15 :: GOSUB 1000 :: G
OSUB 600
120 T=2 :: A=21 :: B=16 :: C
=15 :: GOSUB 1000 :: A=23 ::
GOSUB 1000 :: T=12 :: A=20
:: B=16 :: C=11 :: GOSUB 100
0
130 T=2 :: A,B,C=0 :: GOSUB
1000 :: GOSUB 500 :: T=4 ::
A=21 :: B=16 :: C=13 :: GOSU
B 1000 :: T=10 :: A=25 :: GO
SUB 1000
140 T=2 :: A=28 :: GOSUB 100
0 :: GOSUB 600
150 T=2 :: A=27 :: B=23 :: C
=18 :: GOSUB 1000 :: A=30 ::
GOSUB 1000 :: T=10 :: A=28
:: B=23 :: C=20 :: GOSUB 100
0
160 T=2 :: A,B,C=0 :: GOSUB
1000 :: T=3 :: A=28 :: B=23
:: C=20 :: GOSUB 1000 :: T=1
:: A=27 :: GOSUB 1000 :: GO
SUB 700
170 T=8 :: A=25 :: B=21 :: C
=9 :: GOSUB 1000 :: T=2 :: A
=23 :: B=18 :: C=15 :: GOSUB
1000
```

```

180 T=10 :: A=20 :: B=16 ::
C=11 :: GOSUB 1000 :: T=2 ::
  A,B,C=0 :: GOSUB 1000
190 T=3 :: A=28 :: B=23 :: C
=20 :: GOSUB 1000 :: T=1 ::
A=27 :: GOSUB 1000 :: GOSUB
700
200 T=4 :: A=25 :: B=21 :: C
=16 :: GOSUB 1000 :: A=21 ::
  B=18 :: C=15 :: GOSUB 1000
210 T=14 :: A=20 :: B=16 ::
C=11 :: GOSUB 1000 :: T=2 ::
  A,B,C=0 :: GOSUB 1000 :: ST
OP
500 T=2 :: A=23 :: B=20 :: C
=16 :: GOSUB 1000 :: A=28 ::
  GOSUB 1000 :: A=27 :: GOSUB
  1000 :: A=28 :: GOSUB 1000
  :: A=27 :: GOSUB 1000
510 A=28 :: GOSUB 1000 :: A=
23 :: B=20 :: C=16 :: GOSUB
1000 :: A=20 :: B=16 :: C=11
  :: GOSUB 1000 :: A=16 :: B=
11 :: C=8 :: GOSUB 1000 :: R
ETURN
600 T=2 :: A=27 :: B=23 :: C
=18 :: GOSUB 1000 :: A=23 ::
  B=18 :: C=15 :: GOSUB 1000
  :: A=21 :: GOSUB 1000 :: A=2
3 :: GOSUB 1000
610 A=27 :: GOSUB 1000 :: A=
23 :: GOSUB 1000 :: RETURN
700 T=4 :: A=27 :: B=21 :: C
=16 :: GOSUB 1000 :: T=8 ::
A=25 :: GOSUB 1000 :: T=3 ::
  A=27 :: B=23 :: C=18 :: GOS
UB 1000
710 T=1 :: A=21 :: GOSUB 100
0 :: T=4 :: A=25 :: B=21 ::
C=16 :: GOSUB 1000 :: T=8 ::
  A=23 :: B=20 :: C=16 :: GOS
UB 1000
720 T=3 :: A=25 :: B=21 :: C
=16 :: GOSUB 1000 :: T=1 ::
A=23 :: GOSUB 1000 :: T=2 ::
  A=23 :: B=18 :: C=15 :: GOS
UB 1000
730 A=21 :: GOSUB 1000 :: A=
20 :: GOSUB 1000 :: A=21 ::
GOSUB 1000 :: RETURN

```

Save that under the filename ROSES, clear the memory with NEW, and key this in -

```

1000 CALL SOUND(D*T,N(A),V1,
N(B),V2,N(C),V3):: RETURN

```

Save that by SAVE DSK1.PLAIN3, MERGE. Load ROSES again and merge it in by MERGE DSK1.PLAIN3. Add a line -

105 D=200 and RUN it.

Sounds rather raw and harsh, doesn't it? Try changing that line 105 to -
105 D=200 :: V2=5 :: V3=8

Try it again. Sound better? The first time, all 3 voices were being played at the loudest volume. Usually computer music will sound better if the harmony notes are given a lower volume.

Experiment and find the volumes you like best. Is the music too slow for you? Just change the value of D. Is it not in your singing key? Just change the value of F in line 100, as I showed you before.

```

But, does the music still have too strong a beat for your taste? Clear the memory again and key this in -
1000 CALL SOUND(-4250,N(A+Z)
,V1,N(B+Z),V2,N(C+Z),V3):: G
OSUB 1010 :: RETURN
1010 FOR W=1 TO T*D :: NEXT
W :: RETURN

```

Save that as NEG3, MERGE because it uses negative duration for 3 voices. Then load ROSES again and merge it in. This time, try line 105 with D=50 and with V2 and V3 as you wish. Sound smoother?

In lines 110, 130, 160, 180 and 210 of ROSES, you will find A,B,C=0. That makes all three voices silent, because in line 100 N(0) is given a frequency of 40000 which is above the range of human hearing. This is how I programmed those silent pauses, the "rests" which were written in the music.

On a piano or guitar, the strings continue to vibrate during a rest, so that the sound gradually fades out. However, the electronically generated tones of a computer stop very suddenly. That is why I often add the duration of the rest to the duration of the preceding note, and play it right on through. Some people think that doesn't sound right, so here is another solution. Clear memory again and key this in -

```

2000 FOR W=2 TO 8 STEP 8 ::
CALL SOUND(-999,N(A+Z),V1+W,
N(B+Z),V2+W,N(C+Z),V3+W):: G
OSUB 2010 :: NEXT W :: RETUR
N

```

```

2010 FOR Y=1 TO T*D/4 :: NEX
T Y :: RETURN

```

Save that as REST, MERGE. Load ROSES again, merge in SCALE and NEG3 (this

will not work well with PLAIN3) and merge in REST. Now go to lines 110, 130, 160, 180 and 210, delete the A,B,C=0 :: and change the GOSUB 1000 after it to GOSUB 2000. Add line 105, run it and see if you like that better. Anyway, keep it for now because we will use it again.

You will probably want to have the music play through more than once. Just add :: FOR J=1 TO 4 to the end of line 105 (if you want it to play 4 times) and change the end of line 210 to read NEXT J :: STOP .

I said that you could change the key of the music just by changing the value of F in line 100. There is also a way to change it while the music is playing. After the FOR J=1 TO 4 in 105 put :: Z=Z-(J=2)*3-(J=3)*1+(J=4)*4 That is somewhat complicated but it just means to play the second time three whole keys higher, the third time one key higher still (I know the *1 is unnecessary!) and drop back 4 keys for the 4th time, so you can take it from there and modify it as you wish.

If you want to use that routine with silent rests, change the GOSUB after each rest to 3000 instead of 1000, and add this line -
 3000 CALL SOUND(-4250,N(A),V
 1,N(B),V2,N(C),V3):: GOSUB 1
 010 :: RETURN

This tune happens to end in a rest, which is unusual. If you key in another tune and it seems to end too abruptly, just after that NEXT J and before the STOP, put in a long duration such as T=12 and a GOSUB 2000 to that REST routine to fade out more slowly.

Now, when you are keying in your own tunes, the notes on your sheet music will usually have two or three of those little eggs on the stem. It is best to use the upper one for A, the next one for B, and the lower one for C; the computer could care less, but you will find it easier to keep track of what you are doing. If there are less than three, just go directly below to the bass clef and find a note there. If you still don't have enough, you can always use 0 to make that voice silent. Or, you can usually just let the previous note continue. If your sheet music has guitar chords - those little square grids with dots on them - above the

staff, they will give you some help - if there is no guitar chord above the note you are working on, the chord has not changed and it is safe to use the previous harmony notes.

There are many other CALL SOUND routines you can use for different effects. This is similar to the one that Bill Knecht used for his hymns - I call it VIBRA.

```
105 D=1 :: V1=1 :: V2=5 :: V
3=11
1000 FOR J=1 TO T*D :: CALL
SOUND(-99,N(A),V1,N(B),V2,N(
C),V3):: CALL SOUND(-99,N(A)
*1.01,V1,N(B),V2,N(C),V3)::
NEXT J :: RETURN
```

This one I call WUBBA, for no good reason -

```
105 D=1 :: V1=1 :: V2=5 :: V
3=11
1000 FOR J=1 TO T*D :: CALL
SOUND(-99,N(A),V1,N(B),V2,N(
C),V3):: CALL SOUND(-99,N(A)
*1.01,V1,N(B),V3,N(C),V2)::
NEXT J :: RETURN
```

And this one I call TREM -

```
105 D=1 :: V1=1 :: V2=5 :: V
3=11
1000 FOR J=1 TO T*D :: CALL
SOUND(-999,N(A),V2,N(B),V2,N
(C)*1.01,V3):: CALL SOUND(-9
99,N(A)*1.01,V1,N(B),V2,N(C)
,V3):: NEXT J :: RETURN
```

I included line 105 in those, to merge in the duration and volumes along with the sound routine. Change the value of D to suit yourself, even in decimal increments such as D=1.5 .

It is easy to play a song repeatedly but with a different effect each time. Merge in VIBRA and change its line number to 1010. You can do this by typing 1000 and FCTN X, Enter, FCTN 8 to bring it back, type over the line number, and Enter. Merge in WUBBA and change it to line 1020 in the same way, then TREM and change it to line 1030. Add :: FOR R=1 TO 3 to the end of line 105. Put in a new line 1000 -
 1000 ON R GOSUB 1010,1020,10
 30 :: RETURN

And change the end of line 210 to NEXT R :: STOP.

Next time - more different effects, and autochording.

The TI and the IBM

Some BASIC comparisons

By BARRY A. TRAVER

©1992 B. Traver

This article is the last in a series of three, comparing the TI-99/4A and the IBM, with an emphasis upon programming in BASIC on the two machines. The previous installments were published in the March and April editions of MICROpendium.

The first article was an introduction to some of the similarities and differences between the two. There we noted that sometimes the differences in the BASICs are related to differences in hardware. For example, there is no CALL SPRITE in QuickBASIC, because genuine sprites as we know them on the TI are not supported on the IBM. Likewise, there is no CALL SAY in QuickBASIC and the CALL SOUND in QuickBASIC does not support three-voice harmony, since most IBMs do not support speech or multi-voice music, whereas the typical TI system is capable of both. (If you want those capabilities on an IBM, you have to purchase a special card, such as the SoundBlaster or AdLib card, and people who own such are in the minority.)

If, however, you stay away from full-color fancy graphics with animation, speech, and complex music, it is possible to bring many BASIC programs from one machine to the other without much difficulty. The second article suggested some hints for converting TI Extended BASIC programs to run in QuickBASIC. One useful device is to write QuickBASIC functions and subprograms that emulate TI XB statements that do not have a present exact counterpart in QuickBASIC. (I myself have written QuickBASIC emulations for many TI XB statements, including ACCEPT AT, DISPLAY AT, CALL GCHAR, CALL HCHAR, LINPUT, MAX, MIN, SEG\$, RPT\$, CALL VCHAR, and more. Simplified versions of some of these were included in the second article; for more information on the full package, write to Barry Traver, 835 Green Valley Drive, Philadelphia, PA 19128, enclosing a SASE.)

In this third article, we turn our attention

to going in the opposite direction, i.e., converting QuickBASIC programs (as well as GW-BASIC or BASICA programs) to run in TI Extended BASIC. Important: many of the observations in the previous article have application here (e.g., that TI XB LINPUT is equivalent to QuickBASIC LINE INPUT), so I will not be repeating all that I have said before. That is, this article is dependent upon the preceding one, so do not presume that this article is sufficient by itself. It isn't! (Perhaps I should also mention that even when you put the three articles together, you still have only an introduction to these topics. I do not pretend to be exhaustive in my treatment; my intention is to provide enough specific suggestions to get you on your way.)

Okay. Here's the outline of the rest of this article:

1. Some general comments and remarks about TI BASIC, GW-BASIC (or BASICA), TI Extended BASIC, and QuickBASIC (and QBASIC);
2. Some advice on physically getting a BASIC program from the TI to the IBM;
3. A few thoughts on specific differences in the languages with an eye toward finding TI XB equivalents of statements in an IBM BASIC;
4. Some miscellaneous final comments.

GENERAL COMMENTS

1. Essentially, GW-BASIC (or BASICA; for our purposes the two can be considered to be equivalent) is to QuickBASIC what TI BASIC is to TI Extended BASIC. GW-BASIC is much more limited than QuickBASIC and TI Extended BASIC. One of the great strengths of TI XB (though often not effectively utilized by TI XB programmers) is the ability of having genuine user-defined subprograms with local variables and passed parameters. That feature is lacking from both GW-BASIC and TI BASIC, but is present in both QuickBASIC and TI Extended BASIC.

One thing that this means is that your BASIC vocabulary is "extensible," i.e., you can add new words to TI XB and QuickBASIC, such as CALL DELAY(X).

In addition to subprograms, QuickBASIC also allows user-defined functions. (True, with the DEF statement, TI XB also allows this, but in a much more limited way.) That TI XB does not support multi-variable, multi-statement user-defined functions isn't really a large handicap, because (even though it is slightly more awkward) whatever can be done by calling a function can also be done by calling a subprogram as long as you add one appropriate additional parameter.

Suppose, for example, that you want to define a function RAND which provides a random integer from A to B. Here's a simple program that shows how it might be done in QuickBASIC:

```
INPUT A
INPUT B
CHOICE = RAND(A, B)
PRINT CHOICE
STOP
```

```
FUNCTION RAND (A, B)
RAND = INT(RND * (B - A + 1)) + A
END FUNCTION
```

Since TI XB does not support multi-variable user-defined functions but does support multi-variable user-defined subprograms, here's how you might accomplish the same thing in TI XB:

```
100 INPUT A
110 INPUT B
120 CALL RAND(A, B, CHOICE)
130 PRINT CHOICE
140 STOP
150 SUB RAND (A, B, C)
160 C = INT(RND * (B - A + 1)) + A
170 SUBEND
```

Incidentally, for our purposes, we have not made any distinction between QBASIC and QuickBASIC, but perhaps I should mention that the two are not exactly the same. QBASIC is included with DOS 5.0 for the IBM; QuickBASIC is a separate package. 98 percent of the commands are identical, so don't worry about the differences. QuickBASIC, however, does have some advantages over QBASIC: QuickBASIC includes a compiler, more sophis-

ticated debugging tools, and half a dozen commands (e.g., CALL INTERRUPT) not supported in QBASIC. If you're going from QB to TI XB, the differences are almost never important; if you're going from TI XB to QB, then (in my opinion) QuickBASIC (at \$60-\$70) is a very worthwhile purchase.

ADVICE ON TRANSFERS

2. Physically getting the BASIC program between the IBM and the TI was covered in detail in the April installment. What I want to emphasize here is that what you want to transfer is normal ASCII text, not the tokenized BASIC code as it exists in memory for a running (or a runnable) program.

Example: some of you may know that CHR\$(156) in TI XB is the "token" (or "shorthand") for "PRINT." But that CHR\$(156) on the TI will mean nothing to an IBM. Thus we do not want to transfer to the IBM the actual TI XB program (with its shorthand tokens) but a text LISTING that is more understandable for the IBM. Likewise, IBM BASIC programs exist in two different formats. The "binary" or "fast load" format corresponds to our regular (tokenized) TI XB programs. That format isn't very understandable to a TI. What you need to transfer over to the TI is the IBM BASIC program in (ASCII) text format.

If the IBM BASIC program is in the wrong format, here's how to get it into the right format before transferring it. If you're using GW-BASIC, load in the program and re-save it in ASCII format. Here's the appropriate syntax: SAVE "MYPROG.BAS",A (the "A" at the end is the key). If you're using QuickBASIC, likewise load in the program and re-save it in ASCII format (just designate "Text" rather than "Fast Load" on the menu). Then use any of the procedures mentioned in the preceding article to transfer the text file between the machines.

Once it has reached the TI, you can edit it using the TI-Writer editor (or some reasonable equivalent, such as Funnelweb). (Just be sure to use "PF" rather than "SF" when saving it to disk.) You will, of course, need a utility to convert the text (D/V80) file to a TI XB MERGE (D/V163) file, but many such utilities exist, such as 80MERG, TEXTTOPROG, XLATE, or TEXTLOADER (the last, by Curtis Provance, is a favorite of many). Finally, when you've converted it to a MERGE

file, you need to MERGE it into memory (syntax: MERG DSK1.MYPROG) and then to save it as a normal TI XB program.

LANGUAGE DIFFERENCES

3. In April you saw that there is a lot of overlap between TI XB and QB. Both include ABS, ASC, ATN, CALL, CHR\$, CLOSE, COS, DATA, DIM, END, EOF, ERR, EXP, FOR...NEXT, GOSUB, GOTO, IF...THEN...ELSE, INPUT, INT, LEN, LET, LOG, ON ERROR, ON...GOSUB, ON...GOTO, OPEN, OPTION BASE, PRINT USING, RANDOMIZE, READ, REM, RESTORE, RETURN, RUN, SGN, SIN, SOUND, SQR, STOP, STR\$, TAN, and VAL. Sometimes there are subtle differences (e.g., STR\$ in QB puts a blank space in front of a positive number, and RANDOMIZE in QB needs a "seed" as in RANDOMIZE TIMER), but TI XB and QB have much in common.

Occasionally different words are used for the same function; for instance, INSTR in QB corresponds to POS in TI XB and LOCATE(ROW,COL): PRINT MESSAGE\$ in QB corresponds to DISPLAY AT(ROW,COL):MESSAGE in TI XB, but we also spoke about that last time. What remains to be done is to find TI XB counterparts for words that are unique to QB. Although I don't have enough room remaining here in this final article to do justice to this subject, I do expect to return to the topic in my regular Extended BASIC column as I add these QB features to TI XB as new TI XB subprograms.

For now, however, I'll just describe a few more of the commands, and leave it to you to write equivalents. Many are nice features that may be of value to any TI XB programmer. For instance, LTRIM\$ and RTRIM\$ trim off leading and trailing blanks from a string, and LEFT\$ and RIGHT\$ make it easy to extract a substring at the left or right of a string.

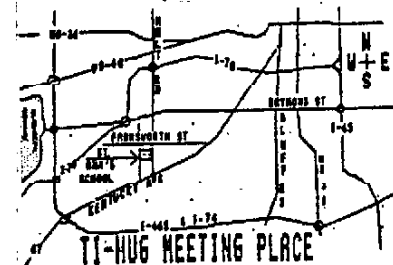
One nice thing about QB is the presence of DO...LOOPS, WHILE...WEND statements, and SELECT CASE options, which make it simpler to write structured programs in BASIC. If you analyze the logic involved, however, any of these can also be performed by using IF...THEN...ELSE variations. (Again, I hope to explain more about that later, when I talk about structured programming in my Extended BASIC column, D.V.)

WORKING WITH 80 COLUMNS

4. Well, once again I'm "out of time and space," but I do hope that I've at least got-

ten you started in getting the TI and the IBM to be on speaking terms with one another, especially in the area of BASIC programs. One of the biggest problems that you may have in converting from QB to TI is that many IBM programs take advantage of an 80-column screen. Doing the same thing in normal graphics mode on a TI (with only 28 or 32 columns to work with) can be quite a challenge. (Maybe enough so to get you to save up for an appropriate 80-column card for your TI, such as a "TINY T-I-M" from OPA?) On the other hand, working with numbers on a TI is much easier than on an IBM. (There you have to be concerned about four types — integers and long integers as well as single- and double-precision decimals — whereas on the TI everything is done with double-precision accuracy, in a more friendly way than is true on the IBM.)

"The proof of the pudding is in the eating." I know that it is possible to convert programs between TI Extended BASIC and QuickBASIC for the IBM because I've done it, both ways. If you have access to both computers and are interested in programming, I challenge you to give it a try as well. And I suspect that in the process you'll come to appreciate your TI all the more, as you recognize that each machine has (and continues to have) advantages that the other does not have. Keep on computin'!



MONTHLY MEETING LOCATION
LITTLE HOUSE NEXT TO
ST. ANN'S SCHOOL
2839 S. MCCLURE
INDIANAPOLIS, IN

MEETING STARTS
AT 2:00 P.M.
JULY 19 1992

Wisconsin company near agreement to produce Myarc products

A Wisconsin firm, Cecure Electronics, expects to become an authorized Myarc repair center by the end of June or the first part of July.

Company spokesman Don Walden says he has been unofficially making repairs to Myarc hardware for approximately two years. He says he can currently do repairs which do not require the gate array, PAL chips or circuit boards, which are proprietary, but he has been in contact with Lou Phillips of Myarc in order to be able to provide service with the proprietary chips and schematics also.

Walden says the company has manufacturing capability, and he is looking to make available Myarc parts and kits to individuals, as well as producing complete boards for Myarc distributors, depending on the type of agreement he reaches with Phillips.

Walden says Phillips has been "overwhelmed." Walden notes, "I have a staff. It's not a business of one person. I look at it as a triple win. Myarc wins because its

name is not going to be maligned. People who have the products win and I win. It's a nice situation all the way around."

Walden says his company has a "niche" in contract manufacturing for research and development products whereby he is able to produce short runs of 10-25 boards.

"We've even done one or two boards for companies. Most larger houses don't want to deal with you unless you want to manufacture thousands of boards," he says.

Walden says he would deal with the hardware portion of the business and Beery Miller (See April 1992 MICROpendium) with the software portion "instead of one person trying to do it all."

He notes, "The demand for Geneves and Lou's products is great. There's a waiting list."

He says he is capable of adding a 32K memory upgrade and a 64K video upgrade to the Geneve and converting a 512 card for use as a Geneve 512K card, and has done these for a number of individuals.

Walden says voltage problems have caused symptoms of "whiteout" or lockup on a number of Geneves sent to him for repair.

He says a lot of the earlier TI PEBs have a higher voltage on the buffer which can cause a Geneve to malfunction. He has found pushbutton PEBs with as high as 15 or 16 volts and rocker type PEBs as high as 12 to 14 volts. He says the extra voltage creates heat, causing the computer to shut down and stop working because it gets too hot.

He says some individuals have pulled out or put cards in with the PEB on, and this usually damages the PAL.

Walden is president of the Milwaukee Users Group and the Wisconsin Council of TI Users Groups and vice president of the Chicago TI Users Group.

For further information, write Cecure Electronics, 7759 So. Scepter Dr. 7, Franklin, WI 53132-2201 or call (414) 529-2173.

HUG OFFICERS

President	Gregory Larson	783-4575
Vice Pres	Bryant Pedigo	255-7381
Secretary	Dan H Eicher	241-994A
Treasurer	Fred Edstrom Jr	898-7300
Librarian	Bryant Pedigo	255-7381

(CONTINUED FROM PAGE 1)

suggestions as to alternative locations or arrangements would be welcome.

A final reminder about dues, remember that our dues are now due in the Spring rather than the anniversary of when you originally joined. Anyone that hasn't already renewed please take a few minutes to write out a check for your dues and mail it to our P.O. Box. Thanks!

- B.C.P.

```

=====
"
"
"           BBS
"   Hoosier Users Group
"   Baud rate 300,1200 & 2400
"   On Line 24 Hours Daily
"           782-994A
"
=====

```

DISCLAIMER

This newsletter is brought to you through the efforts of the officers and members of the HOOSIER USERS GROUP. Every member is encouraged to submit articles.

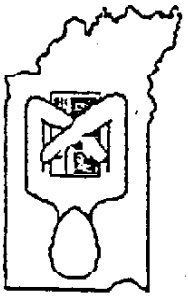
If you have an article you would like to share with the other members mail it to:

Bryant Pedigo
6461 N. Sherman Drive
Indianapolis, IN 46220

Opinions expressed are those of the author and not necessarily those of the HOOSIER USERS GROUP.

MATERIAL
TIME DATED
 July 19, 1992

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.

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

please print	cut on line
Check One:	
Active Member	TODAY'S DATE _____
New: \$20 _____	NAME _____
Renewal: \$18 _____	ADDRESS _____ APT # _____
Dues will be due in May of each year. New members subtract \$1.50 for each month from May to Oct. New member minimum \$10.00	CITY _____ STATE _____ ZIP _____
Amount Enclosed _____	PHONE (____) _____ - _____
# _____ D _____	INTERESTS/
S _____ O _____	COMMENTS _____