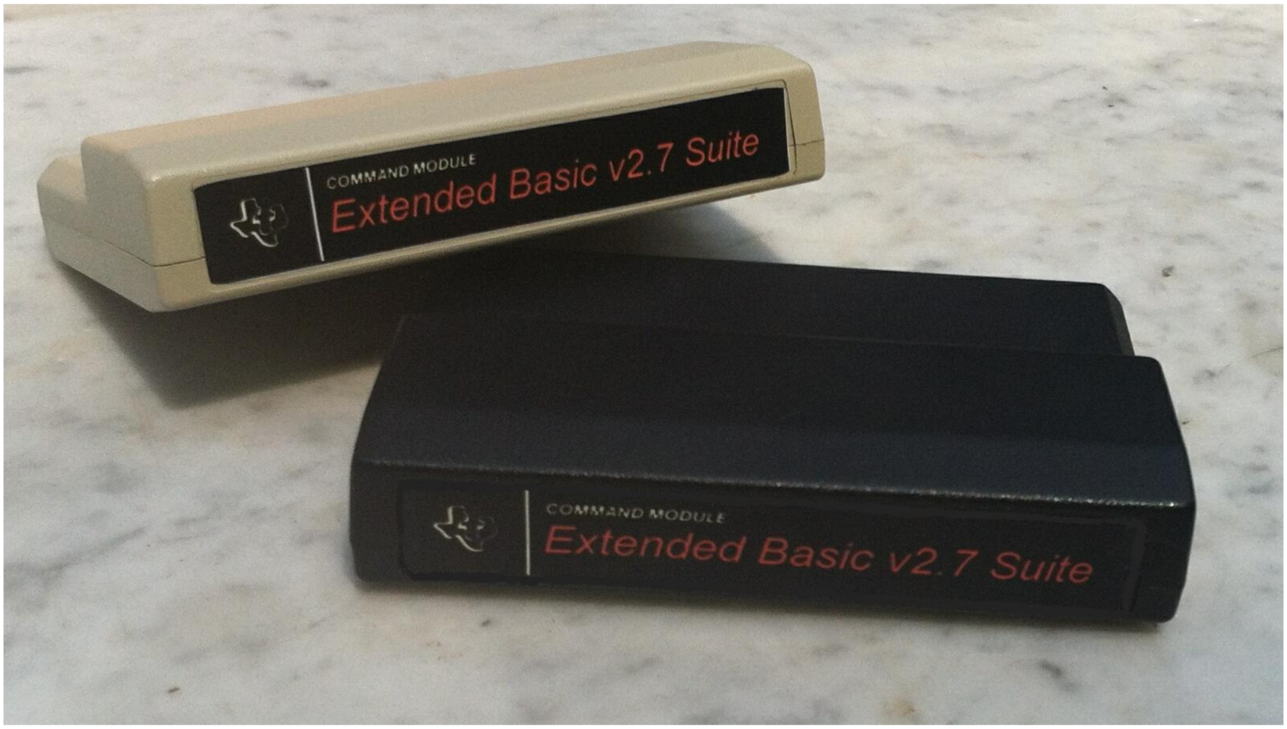
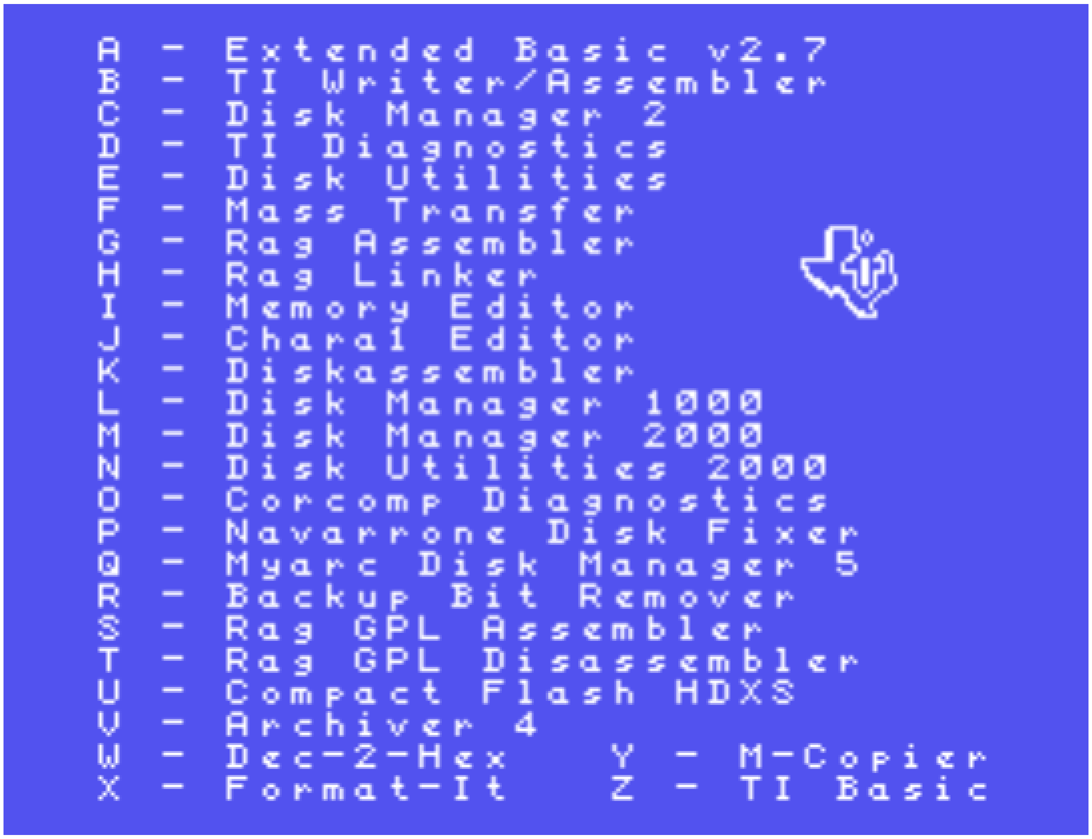


Cartridge Software for the TI99/4A Home Computer

EXTENDED BASIC VERSION 2.7 SUITE CARTRIDGE

This cartridge for the TI99/4A contains 120k of Grom and 512k of Rom memory. It's a combination cartridge with utilities, games, and of course Extended Basic v2.7.

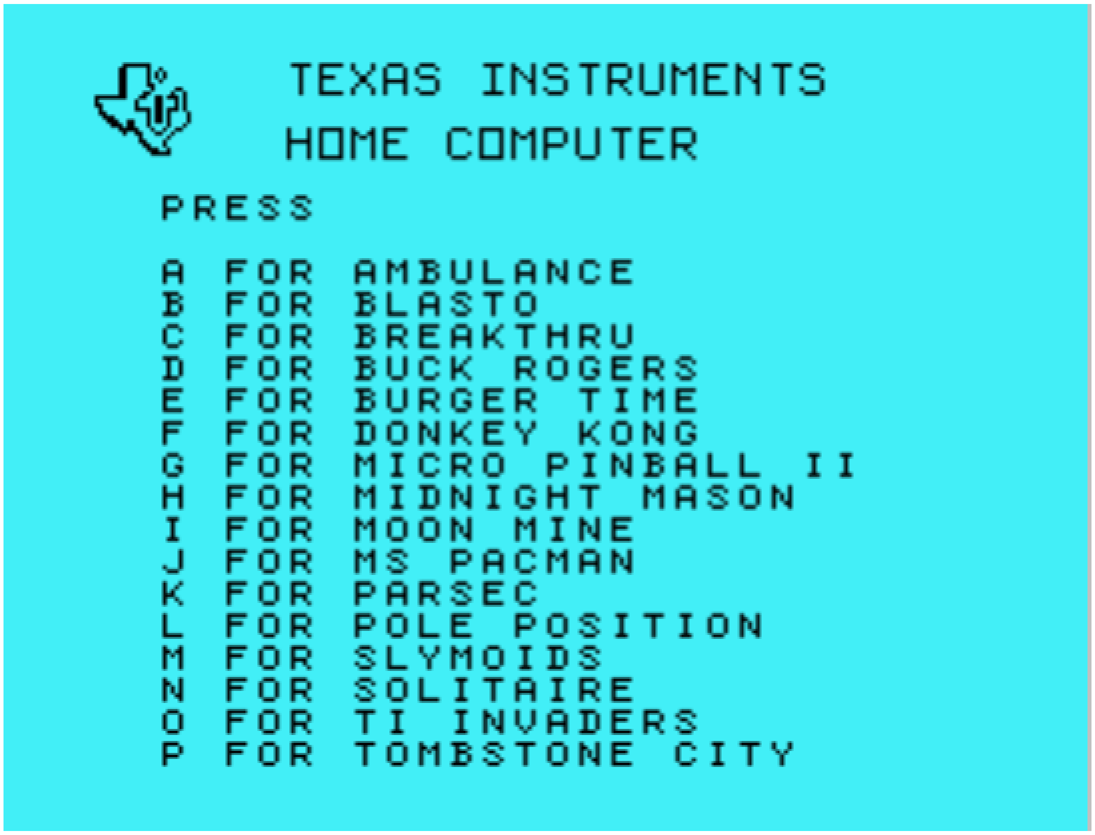
When the cartridge is inserted in the console and the console is powered up, It will automatically start and you will be presented with this startup menu screen:



```
A - Extended Basic v2.7
B - TI Writer/Assembler
C - Disk Manager 2
D - TI Diagnostics
E - Disk Utilities
F - Mass Transfer
G - Rag Assembler
H - Rag Linker
I - Memory Editor
J - Chara1 Editor
K - Diskassembler
L - Disk Manager 1000
M - Disk Manager 2000
N - Disk Utilities 2000
O - Corcomp Diagnostics
P - Navarrone Disk Fixer
Q - Myarc Disk Manager 5
R - Backup Bit Remover
S - Rag GPL Assembler
T - Rag GPL Disassembler
U - Compact Flash HDXS
V - Archiver 4
W - Dec-2-Hex      Y - M-Copier
X - Format-It      Z - TI Basic
```

Pressing any key will run the corresponding menu entry. Execution is immediate in the case of a GPL program (A, B, C, D, & Z). Other entries, which are EA5 or XB programs, load very fast as they are stored in Rom and load with direct memory transfers. Program loading is even faster than when loading from a Ramdisk.

From the startup menu, you can also press the spacebar. You are then presented with a TI menu screen containing 16 games – Ambulance, Blasto, Breakthru, Buck Rogers, Burger Time, Donkey Kong, Micro Pinball II, Midnight Mason, Moon Mine, Ms Pacman, Parsec, Pole Position, Slymoids, Solitaire, TI Invaders, and Tombstone City. Press the spacebar again to go back to the startup menu.

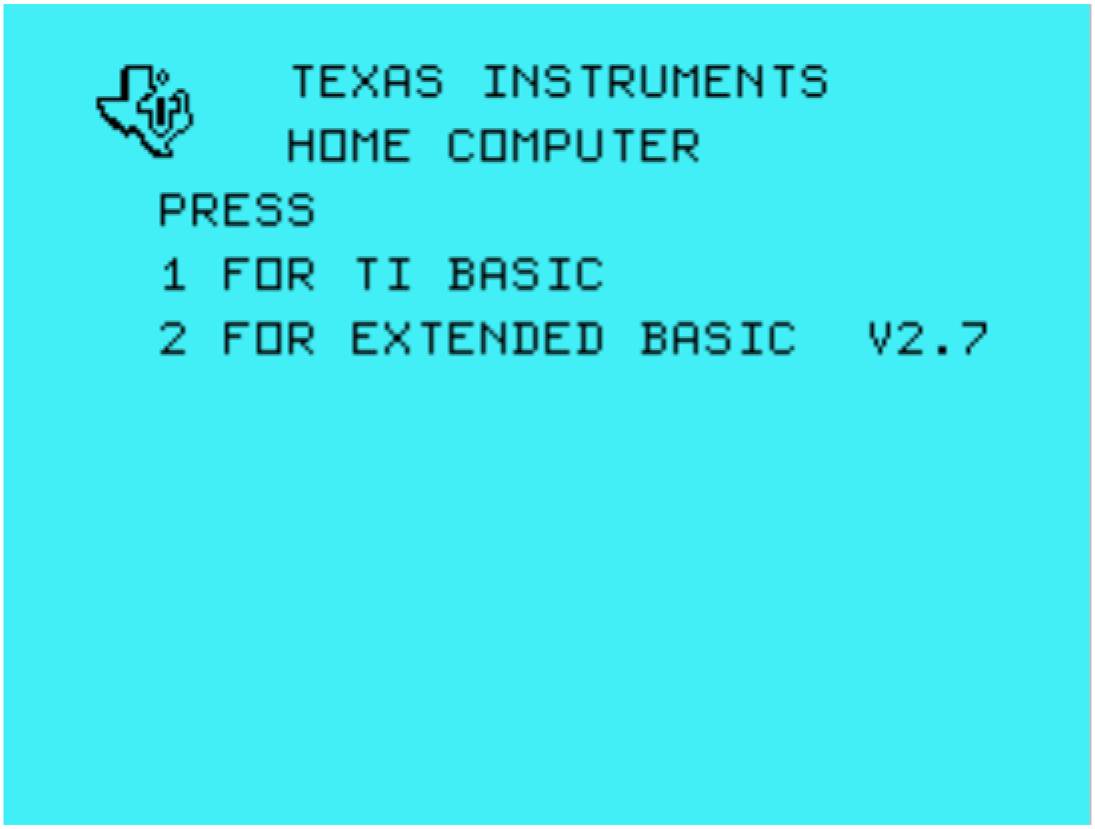


```
TEXAS INSTRUMENTS
HOME COMPUTER

PRESS

A FOR AMBULANCE
B FOR BLASTO
C FOR BREAKTHRU
D FOR BUCK ROGERS
E FOR BURGER TIME
F FOR DONKEY KONG
G FOR MICRO PINBALL II
H FOR MIDNIGHT MASON
I FOR MOON MINE
J FOR MS PACMAN
K FOR PARSEC
L FOR POLE POSITION
M FOR SLYMOIDS
N FOR SOLITAIRE
O FOR TI INVADERS
P FOR TOMBSTONE CITY
```

When powering up or on a reset, you can hold down the spacebar and get the standard TI menu. Only XB v2.7 is on the menu, but it will work without the 32k memory expansion being present if need be.

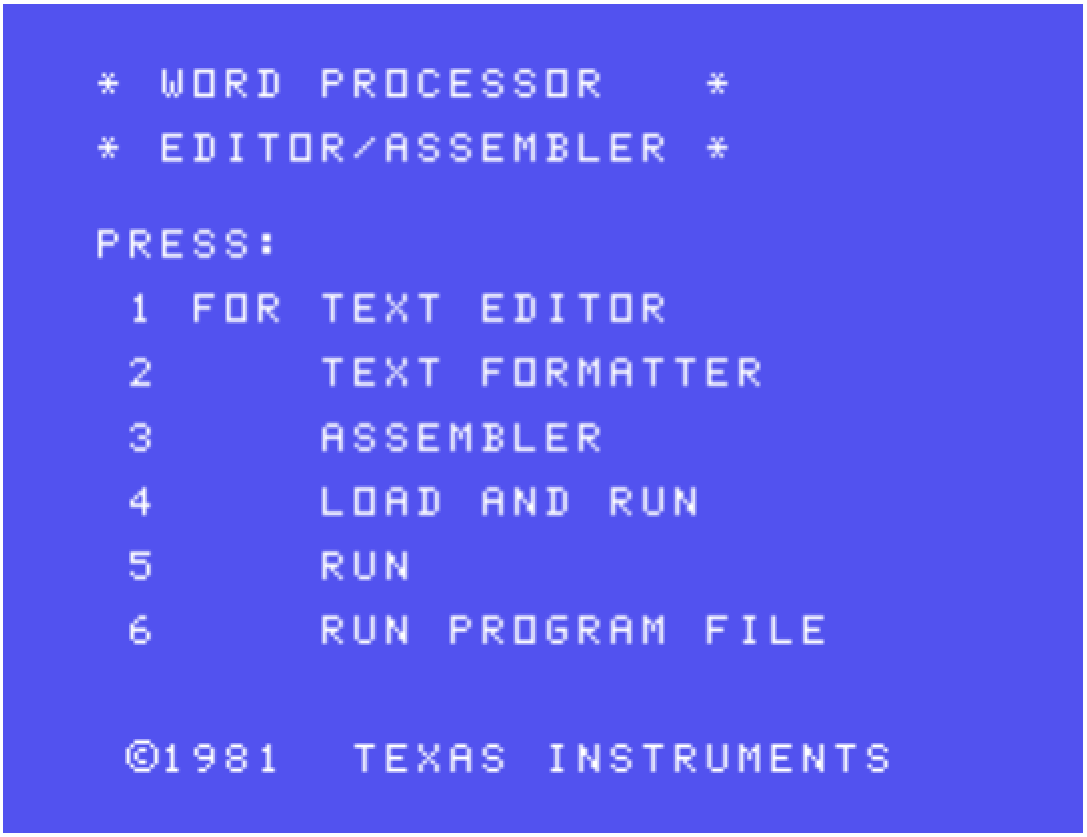


```
        TEXAS INSTRUMENTS
        HOME COMPUTER
  PRESS
  1 FOR TI BASIC
  2 FOR EXTENDED BASIC  V2.7
```

The 3 Grom Carts In Addition To Extended Basic v2.7

TI Writer / Assembler

I have combined the TI-WRITER and EDITOR/ASSEMBLER cartridges, you will have instant access to all the features of both cartridges without the need to load anything from a disk. The Editor used for this module is TI-Writer v4.3, which will take you back to the menu screen with the normal (Q)uit and (E)xit, or the double (QQ)uit works also. The Formatter and Assembler return to the menu with a FCTN-9. You may have to hit FCTN-9 twice to exit the Formatter. Loading of E/A option 5 files (now option 6 on the new menu,) should now be bug-free, as I have cleared more VDP memory upon entry to the module.



```
* WORD PROCESSOR    *
* EDITOR/ASSEMBLER *

PRESS:
 1 FOR TEXT EDITOR
 2     TEXT FORMATTER
 3     ASSEMBLER
 4     LOAD AND RUN
 5     RUN
 6     RUN PROGRAM FILE


 ©1981  TEXAS INSTRUMENTS
```

Disk Manager 2

This cart is included mainly for users of the TI Disk Controller 80 track upgrade. It's the best choice for formatting 80 track, Double Sided, Single Density disks. Maximum drive number has also been increased to 'Z' for users of Horizon Ramdisks.

DISK

MANAGER

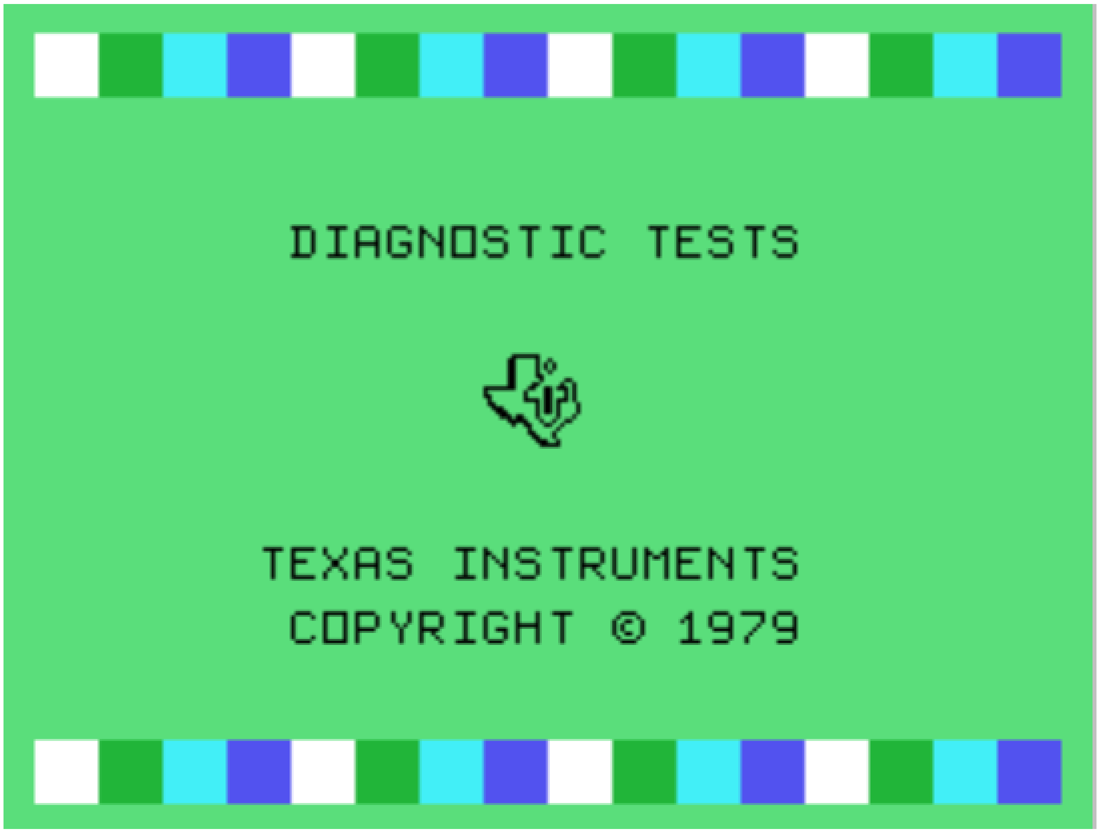
TEXAS INSTRUMENTS

COPYRIGHT ©1982

2.1

Diagnostic Tests

This cart tests the various functions of the TI console.



DIAGNOSTIC TESTS

TEXAS INSTRUMENTS
COPYRIGHT © 1979

CALLS

Type CALL and anything on the list below and it goes there from TI Basic or XB v2.7. The same calls work as DSR entries from TIW/EA option 6, 'Run Program File'. You can use them in programs, too. To access the CALL in a program, use the format RUN “callname” in Extended Basic v2.7. For example, the program:

```
10 RUN “ARC4”
```

when run will load and run Archiver 4.

CALL	RUNS
AMBLNC	Ambulance
ARC4	Archiver 4
BBRMOV	Backup Bit Remover
BKROGR	Buck Rogers
BKTHRU	Breakthru
BLASTO	Blasto
BURGER	Burgertime
CFHDXS	Compact Flash HDXS
CHREDT	Chara1 Editor
CORDIA	Corcomp Diagnostics
DEC2HX	Decimal To Hex To Binary Converter
DM1000	Disk Manager 1000
DM2	Disk Manager 2 cartridge
DM2K	Disk Manager 2000
DONKEY	Donkey Kong
DSKASM	Diskassembler
DSKFIX	Disk Fixer
DSKU	Disk Utilities
DU2K	Disk Utilities 2000
EA	Combined Editor/Assembler and TI Writer
FORMIT	Format-It
GAMES	Game Menu
INVADE	TI Invaders
MCOPY	M-Copier
MASXFR	Mass Transfer
MDM5	Myarc Disk Manager 5
MEMEDT	Memory Editor
MENU	White on blue main cartridge menu
MMASON	Midnight Mason
MONMIN	Moon Mine
MSPAC	Ms Pacman
PARSEC	Parsec
PINBAL	Micro Pinball 2
POLPOS	Pole Position
RAGASS	Rag Assembler
RAGLNK	Rag Linker
RGPLAS	Rag GPL Assembler
RGPLDS	Rag GPL Disassembler
SOLTAR	Solitaire
SLYMOD	Slymoids
TESTS	Diagnostics Tests
TIW	Combined Editor/Assembler and TI Writer
TSCITY	Tombstone City
XB	Extended Basic v2.7

EXTENDED BASIC VERSION 2.7 Documentation

There are several new commands in Extended Basic v2.7 which support floppy disk formatting from the command line of Extended Basic. It is only a limited implementation, as I have very few bytes of program space left in the gram banks. The new commands are: CALL SSSD, CALL DSSD, CALL DSDD, and CALL DSQD. They have been tested with the TI, Corcomp, and Myarc Floppy disk controllers and the Myarc Hard and floppy disk controller. The disk is formatted as it would be using Disk Manager 1000 or Disk Utilities and choosing N(o) when prompted to verify. The only sectors checked are 0 and 1 as these sectors are written to at the end of the formatting process.

Extended Basic v2.7 is compatible with TI Extended Basic with the following modifications:

The foreground and background colors of edit mode are white on dark blue. When in "RUN" mode the colors revert to the standard black on cyan.

The resident character set is one with true lower case.

The command CALL CHARSET has been modified to restore both upper and lower case characters to the original TI character set and also resets screen, foreground, and background colors to Black on Cyan.

The CALL INIT command executes a little faster.

By holding down any key after selection of XB v2.7, the user can bypass the auto-search for DSK1.LOAD and go directly to edit mode.

The following commands are resident in version 2.7:

CALL ALLSET resets the characters 32 through 126 to their original TI definitions. Works like CALL CHARSET but does not affect screen, foreground, or background colors.

CALL BYE will reset to the title screen from a running program.

CALL BEEP will sound the familiar accept tone.

CALL CAT("pathname.") will catalog a device while in command mode. Has been tested with TI, Corcomp, and Myarc floppy controllers as well as the Myarc HFDC and Horizon Ramdisk. The pathname must end in a period. Directories on the HFDC of up to a total length of 11 characters are acceptable. Example - CALL CAT("WDS1.ABCDE.") is the longest pathname that will work.

CALL CHARA1 will restore the resident XB v2.7 character set if it has been re-defined.

CALL CHIME will generate a chime sound.

CALL CLSALL will close all open files.

CALL COLR13 sets all foreground, background, and screen colors to Black on Light Green.

CALL COLR15 sets all foreground, background, and screen colors to Black on Light Blue.

CALL COLR17 sets all foreground, background, and screen colors to Black on Cyan.

CALL COLR18 sets all foreground, background, and screen colors to Black on Medium Red.

CALL COLR19 sets all foreground, background, and screen colors to Black on Light Red.

CALL COLR1A sets all foreground, background, and screen colors to Black on Dark Yellow.

CALL COLR1B sets all foreground, background, and screen colors to Black on Light Yellow.

CALL COLR1F sets all foreground, background, and screen colors to Black on White.

CALL COLRF1 sets all foreground, background, and screen colors to White on Black.

CALL COLRF2 sets all foreground, background, and screen colors to White on Medium Green.

CALL COLRF4 sets all foreground, background, and screen colors to White on Dark Blue.

CALL COLRF6 sets all foreground, background and screen colors to White on Dark Red.

CALL COLRFC sets all foreground, background and screen colors to White on Dark Green.

CALL DSDD will format a 1280 sector disk in drive 1 if you are using a Myarc floppy controller with a 40 track Eprom. A 1440 sector disk will be formatted if using a Corcomp floppy controller, a Myarc floppy controller with an 80 track Eprom, or a Myarc hard and floppy disk controller. You will be prompted to insert a disk in drive 1 and press enter to continue. Pressing any other key aborts the formatting process. Using this command with a TI controller will only be a waste of your time as the floppy controller will go through the whole formatting process only to give you an error at the end. Don't say I didn't warn you!

CALL DSQD will format a 2880 sector disk in drive 2 if you are using a Myarc floppy controller with an 80 track Eprom or a Myarc HFDC properly configured for 80 tracks. You will be prompted to insert a disk in drive 2 and press enter to continue. Pressing any other key aborts the formatting process. Using this command with a TI or /Corcomp controller will only be a waste of your time as the floppy controller will go through the whole formatting process only to give you an error at the end, and may damage your disk and/or drive. Don't say I didn't warn you!

CALL DSSD will format a 720 sector disk in drive 1. You will be prompted to insert a disk in drive 1 and press enter to continue. Pressing any other key aborts the formatting process.

CALL CRASH will generate a crash sound.

CALL GPEEK(address, numeric variable list) will read the contents of addresses in GROMS. The function is similar to CALL PEEK.

CALL HELP will display a help screen of some of these new commands.

CALL HONK will generate the familiar error tone.

CALL LRGCPS will load the large capitals character set from the TI title screen.

CALL MLOAD("filename", mode) will load a program image file (E/A option 5) into memory and run it if MODE=1. Example - CALL MLOAD("DSK1.UTIL1",1) will load and run an E/A option 5 file named UTIL1 from floppy drive #1.

CALL MOVE(mode, start address, target address, # of bytes) will move a block of memory. The four modes available are: 1 - from VDP RAM to VDP RAM, 2 - from VDP RAM to CPU RAM, 3 - from CPU RAM to VDP RAM, 4 - from CPU RAM to CPU RAM

CALL MSAVE("filename", start address, # of bytes) will save a portion of memory in program image format.

CALL NEW is the same as the NEW command except that it can be used within a running program.

CALL NYANYA generates the NYANYA sound our children love to do so often.

CALL QUIT is the same as CALL BYE, that is, it will return to the title screen from a running program.

CALL QUITOF disables the function of the QUIT key (FCTN =).

CALL QUITON restores the function of the QUIT key (FCTN =) after a CALL QUITOF.

CALL RESTORE(numeric variable) is the same as the RESTORE statment except that a variable may be used. Example - 100 A=500 :: CALL RESTORE(A)

CALL SCREENOF turns of the screen.

CALL SCREENON turns on the screen again after a CALL SCREENOF.

CALL SPROF stops the motion of all sprites.

CALL SPRON restarts the motion of all sprites after a CALL SPROF.

Call SSSD will ofrmat a 360 sector disk in drive 1. You will be prompted to insert a disk in drive 1 and press enter to continue. Pressing any other key aborts the formatting process.

CALL VPEEK(address, numeric variable list) will read the contents of addresses in VDP RAM. The function is similar to CALL PEEK.

CALL VPOKE(address, numeric variable list) will write values to VDP RAM. The function is similar to CALL LOAD.

CALL WAIT(duration) causes a delay. DURATION must be a value from 0 to 16382. One second of delay equals a duration value of 50.

CALL XB will re-start XB v2.7 and autoload DSK1.LOAD if no key is held.

CALL XXB will load Barry Traver's XXB version 1.5 into lower memory. It is recommended that CALL INIT be executed immediately prior to CALL XXB to insure all memory is configured properly. All XB v2.7 and XXB v1.5 commands will then be available to the user, making for a very powerful Extended Basic environment. Docs for XXB v1.5 are included here.

XXB/DOCS by Barry A. Traver

What is XXB? Well, just as XB is Extended BASIC, so XXB is Extended Extended BASIC. And just as XB gives you 40 commands not found in TI BASIC, so XXB (Version 1.5) provides you with about 45 commands not available in normal Extended BASIC.

Those new commands (some of which can be accessed by more than one name) are as follows: ACCEPT, BAT (DECHEX, HEXDEC), BEEP, BEGIN, BL1314, BYE, CHANGE, CHRSET (CHARST), CLRTXT, CLS, CLSALL, COLORS (COLOR, SCREEN), DISPLA (DISPLY, DSPLA, DSPLAY, DSPLY, PRINT), EA5, GRAPH (MODE28, MODE32), GRPCOL, HONK, INIT, KEYR, LGCAPS, MOVE (GOSPRT), NEW, NEWCLR, NEWSET (LOW), NOMOVE (STSPRT), NOQUIT (QUITOF), NULLIT, OKQUIT (QUIT, QUITON), PEEKC (CPEEK), PEEKG (GPEEK), PEEKV (VPEEK), POKEC (CPOKE), POKEV (VPOKE), PROTEC, READ (RSEC), RELOAD, RENAME, RUN, SCROFF, SCRON, SMCAPS, TEXT (MODE40), TXTCOL, VDPWTR (VDPREG), WRITE (WSEC), XXB (1.5).

These routines were contributed by Andy Becker, J. Peter Hoddie, Curtis Provance, Barry Traver, and others, and the combination offers up some tremendous facilities, especially for the creative Extended BASIC programmer.

The new commands can be placed in the following categories: (1) Disk Operations, (2) Character Sets, (3) Text Mode, (4) Graphics Mode, (5) Peeks and Pokes, (6) Noises, and (7) Miscellaneous (wouldn't you know!).

Let's take them up, category by category.

(1) DISK OPERATIONS

READ and WRITE (sometimes known as "RAW," standing for "Read And Write") allow you to read or write a single sector on a disk. The format is as follows:

CALL LINK("READ",DRIVE,SECTOR,STRING1$,STRING2$), where DRIVE is the drive number, SECTOR is the sector number (remember: the first sector is sector zero!), and STRING1$ and STRING2$ are two 128-byte strings that contain the information from the sector. (The reason we need two strings is that one string just isn't quite big enough: a sector contains 256 bytes of information, and the maximum length of a string is 255 bytes!)

CALL LINK("WRITE",DRIVE,SECTOR,STRING1$,STRING2$) is similar, where DRIVE is drive number, SECTOR is sector number, and STRING1$ and STRING2$ are the two 128-byte strings of information to be written to the disk.

Many have put these "RAW" routines to good use, including John Johnson (with his PRBUTL, a very nice PRBASE utility) and Rich Mitchell (who knows as much about Microsoft Multiplan (tm) as anyone I know). My pioneer ARCHIVER program used these READ and WRITE routines.

PROTEC and RENAME allow you to change file protection or filenames for disk files from XB without running a disk manager program. The format is as follows:

CALL LINK("PROTEC",DSKN.FILENAME$,FLAG), where DSKN.FILENAME$ indicates the full name of the file (including drive number) and FLAG is the flag to indicate what you want done (using FLAG=0 to unprotect a file or FLAG=1 to protect).

CALL LINK("RENAME",DRIVE,OLDNAME$,NEWNAME$), where DRIVE is the drive number, OLDNAME$ is the old name of the file, and NEWNAME$ is the new name of the file. (Note well: PROTEC and RENAME use different formats in their respective CALL LINKs.)

(2) CHARACTER SETS

CALL LINK("NEWSET") gives you a new character set, with large capital letters and true lower case.

CALL LINK("CHRSET") -- when used along with CALL CHARSET -- restores the old character set. (CALL CHARSET in Extended BASIC does not restore the lower case letters nor clear character sets 13 and 14, so CALL LINK("CHRSET") makes that simple to do. This can be important if you do any program chaining, which does not automatically restore character definitions!)

CALL LINK("LGCAPS") gives you large capital letters (like those on the TI title screen), whereas CALL LINK("SMCAPS") restores capital letters to their normal size and form.

CALL LINK("BL1314") blanks out (that is, undefines) characters in sets 13 and 14, without disturbing the definitions of other character sets.

(3) TEXT MODE

Text (40-column) mode is not normally available in Extended BASIC, unless you have, for example, Oak Tree Systems' "DEP" (Display Enhancement Package) or Paragon Computing's "EDP" (Enhanced Display Package), which are XB extensions similar to XXB. Here are the commands for 40-column mode:

CALL LINK("TEXT") puts you into text mode. Warning: text mode is an area in which the Extended BASIC programmer must walk with care. Two things to remember: (1) You must use CALL LINK("GRAPH") or its equivalent in order to return safely to graphics mode, and (2) any kind of error in text mode can throw you into never-never land!

CALL LINK("INIT",LEFTMARGIN,RIGHTMARGIN,WIDTH) lets you set your left and right margins as well as the mode "width." For text mode, MODE would normally be 40 (unless you want to print slanted lines with 39 or 41).

CALL LINK("TXTCOL",FOREGROUND,BACKGROUND) sets the foreground and background colors for text mode. An alternative that works essentially equally well is CALL LINK("COLORS",FOREGROUND,BACKGROUND).

CALL LINK("DISPLA",ROW,COLUMN,MESSAGE$) is very similar in operation to XB's DISPLAY AT, but DISPLAY AT works in a meaningful way only in normal graphics mode, while CALL LINK("DSPLY",ROW,COLUMN,MESSAGE$) can be used in either graphics mode or text mode, provided that CALL LINK("INIT",LEFTMARGIN,RIGHT-MARGIN,WIDTH) has been properly set.

CALL LINK("ACCEPT",ROW,COLUMN,MAXLENGTH,VALIDATION$,RESPONSE$) provides a multi-line ACCEPT AT for graphics or text mode. And you can set the maximum length and the validation string (features we're used to from Extended BASIC), as well as suggest a default Response. (You can enter a null string ("") for VALIDATION$ if you don't desire any checking to be done; otherwise VALIDATION$ can be up to 127 characters long. Also, if RESPONSE$ is defined before the CALL LINK, it will show up as a default entry.)

The three preceding routines - DISPLA, ACCEPT, and INIT - were written by J. Peter Hoddie.

Many CALLs accessed while in text mode create a screen "glitch," so CALL LINK ("KEYR",RESPONSE$) is provided to avoid that problem. (One limitation that comes with working with text mode from Extended BASIC is that apparently all CALLs to user-defined subprograms will cause such glitches.) See the next section for more information on CALL LINK("KEYR",RESPONSE$).

CALL LINK("SCROFF") turns off the screen display, while CALL LINK("SCRON") turns it back on again (e.g., after you've finished putting up your screen display). (This routine is most useful, however, when you're doing graphics, which is something you're more likely to be doing in graphics mode.)

CALL LINK("CLRTXT") will clear the screen in text mode. It operates in a similar way to CALL CLEAR in graphics mode. CALL LINK("CLS") will also clear the screen in text mode.

Finally, CALL LINK("GRAPH") takes you safely back to graphics mode in Extended BASIC.

(4) GRAPHICS MODE

CALL LINK("GRPCOL",FOREGROUND,BACKGROUND) sets the foreground and background colors for character sets 1 through 12 in graphics mode. An alternative that works essentially the same is CALL LINK("COLORS",FOREGROUND,BACKGROUND). You may want to do a CALL SCREEN(BACKGROUND) as well in your XB program, since this routine does not do that automatically for you.

CALL LINK("INIT",LEFTMARGIN,RIGHTMARGIN,WIDTH) also has its uses in graphics mode, where the mode should normally be set to 32 (although 31 and 33 are also useful if you want to print slanted lines to the screen). CALL LINK("BEGIN") is equivalent to CALL LINK("INIT",3,30,32), which sets things up for normal operation using "ACCEPT" and "DISPLA" in graphics mode.

CALL LINK("DISPLA",ROW,COLUMN,MESSAGE$) and CALL LINK("ACCEPT",ROW,COLUMN, MAXLENGTH,VALIDATION$,RESPONSE$) grant us some special uses also in graphics mode (e.g., the latter provides us with a multi-line ACCEPT AT - see discussion in preceding section for more information).

CALL LINK("KEYR",RESPONSE$) is a good replacement for the following common code:

```
     100 CALL KEY(0,K,S) :: IF S<1 THEN 100 ELSE RESPONSE$=CHR$(K)
```

Thus it has some use in graphics mode as well as in text mode, for which it was originally written.

CALL LINK("NOMOVE") keeps sprites from moving, whereas CALL LINK("MOVE") starts them moving simultaneously.

CALL LINK("SCROFF") turns off the screen display, while CALL LINK("SCRON") turns it back on again (e.g., after you've finished putting up your graphics).

(5) PEEKS AND POKES

My PEEKs and POKEs are different from usual in that they deal entirely with strings rather than values, but this is much more memory-efficient for Extended BASIC.

CALL LINK("POKEC",LOCATION,STRING$) and CALL LINK("POKEV",LOCATION,STRING$) poke strings into CPU RAM and VDP RAM respectively.

Likewise, CALL LINK("PEEKC",LOCATION,SIZE,STRING$) and CALL LINK("PEEKV",LOCATION,SIZE,STRING$) are able to peek into a certain location for a string of designated size, again for either CPU RAM or VDP RAM. In similar fashion, CALL LINK("PEEKG",LOCATION,SIZE,STRING$) allows you to look at GROM or GRAM. (This last routine comes from Andy Becker.)

So far as VDP RAM is concerned, there is no compensation for the >60 (96 decimal) offset for the screen, but that should not be a handicap, since we have other utilities to deal with the screen (e.g., ACCEPT, DISPLA, etc.).

CALL LINK("VDPWTR",REGISTER,VALUE) allows you to do a VDPWTR from XB. Unless you know some assembly language, you may not get much use out of this routine, but it does have some powerful applications.

(6) NOISES

CALL LINK("BEEP") generates the familiar accept tone, while CALL LINK("HONK") generates the familiar error tone.

(7) MISCELLANEOUS

CALL LINK("CHANGE",STATEMENT,POSITION,NEWNUMBER) allows you to change line numbers after GOSUB, GOTO, RESTORE, USING, etc. For example, if line 200 says ON X GOSUB 1000,2000,3000, then a CALL LINK("CHANGE",200,2,4000) would change line 200 to read ON X GOSUB 1000,4000,3000. (This useful routine was contributed by Curtis Provance.)

CALL LINK("NULLIT",ARRAY(),LOW,HIGH) and CALL LINK("NULLIT",ARRAY$(),LOW,HIGH) allow you to re-initialize immediately elements of an array to zeros or null strings. (This routine and the next three are by Peter Hoddie.)

CALL LINK("BAT",STRING$,NUMBER) enables you to convert numbers from decimal to hex (or vice versa). Comment: Note that STRING$ and NUMBER in the CALL LINK must be variable names, not constants. First set string to what you want to convert (hex or decimal), and then do the CALL LINK: the string will now be the hex string, and the number will be the decimal number. Example: if you set A$ to ">20" and then do a CALL LINK("BAT",A$,A), A$ will be set to ">0020" and A will be set to 32. Another example: if you set A$ to "255" and then do a CALL LINK("BAT",A$,A), A$ will be set to ">00FF" and A will be set to 255.

CALL LINK("EA5",SOURCE$) allows you to load in an EA5 program from XB. If you are loading the program from cassette, use "CS1" as the source (or "CS1.X" if there is more than one EA5 file to be loaded); if you are loading the program from disk, use the format "DSKN.FILENAME".

CALL LINK("CLSALL") closes all open files.

Some of the next CALL LINKs came as a result of information provided by Craig Miller in his excellent NIGHT MISSION manual.

CALL LINK("RUN") re-runs the program, but without the pre-scan that normally takes place when that is done. CALL LINK("RELOAD") is equivalent to RUN "DSK1.LOAD". And CALL LINK("BYE") closes all files and returns one to the title screen.

CALL LINK("XXB") can be used as a test to see whether XXB has been loaded into memory; CALL LINK("1.5") checks to see whether version 1.5 of XXB has been loaded.

CALL LINK("NOQUIT") disables the "Quit" key, while CALL LINK("OKQUIT"), re-enables the "Quit" key again.

CALL LINK("NEW") prepares the way for a New program, without clearing the screen, while CALL LINK("NEWCLR") prepares the way for a New program and Clears the screen.

XXB/1-5: REFERENCE CHART FOR CALL LINKS

NAME	ADDRESS	FORMAT AND PURPOSE
1.5 (or XXB)	>269C	CALL LINK("1.5") Purpose: to verify that XXB, version 1.5, has been loaded.
ACCEPT	>3582	CALL LINK("ACCEPT",ROW,COLUMN,MAXLENGTH,VALIDATION$,RESPONSE$) Purpose: to provide a multi-line ACCEPT AT for graphics or text mode. Comment: Enter null string ("") for VALIDATION$ if you don't desire any checking to be done; otherwise VALIDATION$ can be up to 127 characters. Also, if RESPONSE$ string is defined before the CALL LINK, it will show up as a default entry.
BAT (or DECHEX) (or HEXDEC)	>277A	CALL LINK("BAT",STRING$,NUMBER) Purpose: to convert numbers between decimal and hex (and vice versa). Comment: Note that STRING$ and NUMBER in the CALL LINK must be variable names, not constants. First set string to what you want to convert (hex or decimal), and then do the CALL LINK: the string will now be the hex string, and the number will be the decimal number.
BEEP	>314A	CALL LINK("BEEP") Purpose: to generate accept tone.
BEGIN	>354E	CALL LINK("BEGIN") Purpose: to set up for normal graphics mode (with 28 print columns). Comment: CALL LINK("BEGIN") is equivalent to CALL LINK("INIT", 3,30,32).
BL1314	>2D14	CALL LINK("BL1314") Purpose: to blank (undefine) characters in sets 13 and 14.
BYE	>29F6	CALL LINK("BYE") Purpose: to return to title screen with files closed.
CHANGE	>2A24	CALL LINK("CHANGE",STATEMENT,POSITION,NEWNUMBER) Purpose: to change line numbers after GOSUB, GOTO, RESTORE, USING, etc.
CHARST (or CHRSET)	>2D1C	CALL LINK("CHARST") Purpose: to restore character sets 9 through 14 (including lower case). Comment: Use CALL CHARSET to restore character sets 0 through 8.
CHRSET (or CHARST)	>2D1C	CALL LINK("CHRSET") Purpose: to restore character sets 9 through 14 (including lower case). Comment: Use CALL CHARSET to restore character sets 0 through 8.

```
CLRTXT  >3B48   CALL LINK("CLRTXT")
                Purpose:  to provide CALL CLEAR for text mode.

CLS     >39F2   CALL LINK("CLS")
                Purpose:  to provide CALL CLEAR for graphics or text mode.

CLSALL  >3136   CALL LINK("CLSALL")
                Purpose:  to close all open files.

COLOR   >3A34   CALL LINK("COLOR",FOREGROUND,BACKGROUND)
 (or COLORS)    Purpose:  to set screen colors for graphics or text mode.
                Comment:  Like GRPCOL, in graphics mode it sets colors for
                  character sets 1 through 12 (not 0, 13, or 14).  Also, in
                  graphics mode you may want to do a CALL SCREEN(BACKGROUND)
                  as well.

COLORS  >3A34   CALL LINK("COLORS",FOREGROUND,BACKGROUND)
 (or COLOR)     Purpose:  to set screen colors for graphics or text mode.
                Comment:  Like GRPCOL, in graphics mode it sets colors for
                  character sets 1 through 12 (not 0, 13, or 14).  Also, in
                  graphics mode you may want to do a CALL SCREEN(BACKGROUND)
                  as well.

CPEEK   >2A94   CALL LINK("CPEEK",LOCATION,SIZE,STRING$)
 (or PEEKC)     Purpose:  to look at CPU RAM.

CPOKE   >2AEA   CALL LINK("CPOKE",LOCATION,STRING$)
 (or POKEC)     Purpose:  to load information into CPU RAM.
                Comment:  This is an efficient alternative to the more usual
                  CALL LOAD(address,value...).

DECHEX  >277A   CALL LINK("DECHEX",STRING$,NUMBER)
 (or BAT)       Purpose:  to convert numbers from decimal to hex.
 (or HEXDEC)    Comment:  Note that STRING$ and NUMBER in the CALL LINK must
                  be variable names, not constants.  First set string to what
                  you want to convert (hex or decimal), and then do the CALL
                  LINK:  the string will now be the hex string, and the number
                  will be the decimal number.

DISPLA  >38DC   CALL LINK("DISPLA",ROW,COLUMN,MESSAGE$)
 (or DISPLY)    Purpose:  to provide DISPLAY AT for graphics or text mode.
 (or DSPLA)     Comment:  If used with INIT, this command can "do windows,"
 (or DSPLAY)      print slanted lines, etc.
 (or DSPLY)
 (or PRINT)

DISPLY  >38DC   CALL LINK("DISPLY",ROW,COLUMN,MESSAGE$)
 (or DISPLA)    Purpose:  to provide DISPLAY AT for graphics or text mode.
 (or DSPLA)     Comment:  If used with INIT, this command can "do windows,"
 (or DSPLAY)      print slanted lines, etc.
 (or DSPLY)
 (or PRINT)
```

```
DSPLA   >38DC   CALL LINK("DSPLA",ROW,COLUMN,MESSAGE$)
 (or DISPLA)    Purpose:  to provide DISPLAY AT for graphics or text mode.
 (or DISPLY)    Comment:  If used with INIT, this command can "do windows,"
 (or DSPLAY)      print slanted lines, etc.
 (or DSPLY)
 (or PRINT)

DSPLAY  >38DC   CALL LINK("DSPLAY",ROW,COLUMN,MESSAGE$)
 (or DISPLA)    Purpose:  to provide DISPLAY AT for graphics or text mode.
 (or DISPLY)    Comment:  If used with INIT, this command can "do windows,"
 (or DSPLA)       print slanted lines, etc.
 (or DSPLY)
 (or PRINT)

DSPLY   >38DC   CALL LINK("DSPLY",ROW,COLUMN,MESSAGE$)
 (or DISPLA)    Purpose:  to provide DISPLAY AT for graphics or text mode.
 (or DISPLY)    Comment:  If used with INIT, this command can "do windows,"
 (or DSPLA)       print slanted lines, etc.
 (or DSPLAY)
 (or PRINT)

EA5     >2B3A   CALL LINK("EA5",DSKN.FILENAME$)
                Purpose:  to load in an EA5 program from XB (from cassette or
                  from disk).

GOSPRT  >3C1E   CALL LINK("GOSPRT")
 (or MOVE)      Purpose:  to start sprite motion.

GPEEK   >307E   CALL LINK("GPEEK",LOCATION,SIZE,STRING$)
 (or PEEKG)     Purpose:  to look at GROM or GRAM.

GRAPH   >3B54   CALL LINK("GRAPH")
 (or MODE28)    Purpose:  to set up for graphics mode.
 (or MODE32)

GRPCOL  >3A7E   CALL LINK("GRPCOL",FOREGROUND,BACKGROUND)
                Purpose:  to change colors of character sets 1 through 12 for
                  graphics mode.

HEXDEC  >277A   CALL LINK("HEXDEC",STRING$,NUMBER)
 (or BAT)       Purpose:  to convert numbers from hex to decimal.
 (or DECHEX)    Comment:  Note that STRING$ and NUMBER in the CALL LINK must
                  be variable names, not constants.  First set string to what
                  you want to convert (hex or decimal), and then do the CALL
                  LINK:  the string will now be the hex string, and the number
                  will be the decimal number.

HONK    >3158   CALL LINK("HONK")
                Purpose:  to generate error tone.

INIT    >390C   CALL LINK("INIT",LEFTMARGIN,RIGHTMARGIN,WIDTH)
                Purpose:  to set up margins, and width of window.
                Comment:  Width is normally 32 or 40 to match mode, but you can
                  do slanted lines with 31 or 33 in graphics mode or with 39 or
                  41 in text mode.
```

```
KEYR    >30FE   CALL LINK("KEYR",RESPONSE$)
                Purpose:  to provide equivalent for following line of XB code:
                  100 CALL KEY(0,K,S):: IF S<1 THEN 100 ELSE R$=CHR$(K).
                Comment:  This routine is especially useful in text mode (no
                  screen glitches!).

LGCAPS  >3052   CALL LINK("LGCAPS")
                Purpose:  to create large capital letters (like those on title
                  screen).

LOW     >303A   CALL LINK("LOW")
 (or NEWSET)    Purpose:  to call new character set with true lower case.

MODE28  >3B54   CALL LINK("MODE28")
 (or GRAPH)     Purpose:  to set up for graphics mode.
 (or MODE32)

MODE32  >3B54   CALL LINK("MODE32")
 (or GRAPH)     Purpose:  to set up for graphics mode.
 (or MODE28)

MODE40  >3AF4   CALL LINK("MODE40")
 (or TEXT)      Purpose:  to set up for text mode.

MOVE    >3C1E   CALL LINK("MOVE")
 (or GOSPRT)    Purpose:  to start sprite motion.

NEW     >29A2   CALL LINK("NEW")
                Purpose:  perform NEW without clearing of screen.

NEWCLR  >29B6   CALL LINK("NEWCLR")
                Purpose:  perform NEW with clearing of screen.

NEWSET  >303A   CALL LINK("NEWSET")
 (or LOW)       Purpose:  to call new character set with true lower case.

NOMOVE  >3C08   CALL LINK("NOMOVE")
 (or STSPRT)    Purpose:  to stop sprite motion.

NOQUIT  >29C6   CALL LINK("NOQUIT")
 (or QUITOF)    Purpose:  to disable quit key.

NULLIT  >3168   CALL LINK("NULLIT",ARRAY(),LOW,HIGH)
                 or
                CALL LINK("NULLIT",ARRAY$(),LOW,HIGH)
                Purpose:  to re-initialize elements of an array to zeros or
                  null strings.

OKQUIT  >29D6   CALL LINK("OKQUIT")
 (or QUIT)      Purpose:  to enable quit key.
 (or QUITON)

PEEKC   >2A94   CALL LINK("PEEKC",LOCATION,SIZE,STRING$)
 (or CPEEK)     Purpose:  to look at CPU RAM.
```

```
PEEKG   >307E   CALL LINK("PEEKG",LOCATION,SIZE,STRING$)
 (or GPEEK)     Purpose:  to look at GROM or GRAM.

PEEKV   >3C34   CALL LINK("PEEKV",LOCATION,SIZE,STRING$)
 (or VPEEK)     Purpose:  to look at VDP RAM.

POKEC   >2AEA   CALL LINK("POKEC",LOCATION,STRING$)
 (or CPOKE)     Purpose:  to load information into CPU RAM.
                Comment:  This is an efficient alternative to the more usual
                  CALL LOAD(address,value...).

POKEV   >3C8A   CALL LINK("POKEV",LOCATION,STRING$)
 (or VPOKE)     Purpose:  to load information into VDP RAM.

PRINT   >38DC   CALL LINK("PRINT",ROW,COLUMN,MESSAGE$)
 (or DISPLA)    Purpose:  to provide DISPLAY AT for graphics or text mode.
 (or DISPLY)    Comment:  If used with INIT, this command can "do windows,"
 (or DSPLA)       print slanted lines, etc.
 (or DSPLAY)
 (or DSPLY)

PROTEC  >2CB8   CALL LINK("PROTEC",DSKN.FILENAME$,FLAG)
                Purpose:  to change protection of disk file.
                Comment:  FLAG=0 to unprotect, FLAG=1 to protect.

QUIT    >29D6   CALL LINK("QUIT")
 (or OKQUIT)    Purpose:  to enable quit key.
 (or QUITON)

QUITOF  >29C6   CALL LINK("QUITOF")
 (or NOQUIT)    Purpose:  to disable quit key.

QUITON  >29D6   CALL LINK("QUITON")
 (or OKQUIT)    Purpose:  to enable quit key.
 (or QUIT)

READ    >32CA   CALL LINK("READ",DRIVE,SECTOR,STRING1$,STRING2$)
 (or RSEC)      Purpose:  to read a single disk sector.

RELOAD  >29E6   CALL LINK("RELOAD")
                Purpose:  to provide equivalent of RUN "DSK1.LOAD".

RENAME  >2C22   CALL LINK("RENAME",DRIVE,OLDNAME$,NEWNAME$)
                Purpose:  to change name of disk file.

RSEC    >32CA   CALL LINK("RSEC",DRIVE,SECTOR,STRING1$,STRING2$)
 (or READ)      Purpose:  to read a single disk sector.

RUN     >2A06   CALL LINK("RUN")
                Purpose:  to re-RUN program without the need for a repeat of
                  pre-scan.

SCROFF  >3BE6   CALL LINK("SCROFF")
                Purpose:  to turn off screen display.
```

```
SCRON   >3BC4   CALL LINK("SCRON")
                Purpose:  to turn on screen display.

SMCAPS  >3068   CALL LINK("SMCAPS")
                Purpose:  to restore normal size capital letters.

STSPRT  >3C08   CALL LINK("STSPRT")
 (or NOMOVE)    Purpose:  to stop sprite motion.

TEXT    >3AF4   CALL LINK("TEXT")
 (or MODE40)    Purpose:  to set up for text mode.

TXTCOL  >3A46   CALL LINK("TXTCOL",FOREGROUND,BACKGROUND)
                Purpose:  to set screen colors for text mode.

VDPREG  >3CCE   CALL LINK("VDPREG",REGISTER,VALUE)
 (or VDPWTR)    Purpose:  to allow a VDPWTR from XB.

VDPWTR  >3CCE   CALL LINK("VDPWTR",REGISTER,VALUE)
 (or VDPREG)    Purpose:  to allow a VDPWTR from XB.

VPEEK   >3C34   CALL LINK("VPEEK",LOCATION,SIZE,STRING$)
 (or PEEKV)     Purpose:  to look at VDP RAM.

VPOKE   >3C8A   CALL LINK("VPOKE",LOCATION,STRING$)
 (or POKEV)     Purpose:  to load information into VDP RAM.

WRITE   >32D4   CALL LINK("WRITE",DRIVE,SECTOR,STRING1$,STRING2$)
 (or WSEC)      Purpose:  to write a single disk sector.

WSEC    >32D4   CALL LINK("WSEC",DRIVE,SECTOR,STRING1$,STRING2$)
 (or WRITE)     Purpose:  to write a single disk sector.

XXB     >269C   CALL LINK("XXB")
 (or 1.5)       Purpose:  to verify that XXB has been loaded.
```

OTHER MENU OPTIONS

It would be an exhaustive effort to try and include all the documentation for all the other EA5 and XB programs on the menu. Please refer to the original documentation for these programs. If you have a specific question about any of them, feel free to ask me as I may be able to answer it in at least a semi-intelligent manner. ☺

ONE LAST COMMENT

There’s an ‘Easter Egg’ hidden in the cartridge. It’s your job to find it. ☺

Extended Basic v2.7 Suite by:

Tony Knerr
17 Marshall Circle
Downingtown, PA 19335
610-269-7447
tonyknerr@comcast.net

Credits and thanks to:

Mike Brent
Kevan Coleman
Jim Fetzner
Jon Guidry
Barry Traver
Tim Tesch
Harry Wilhelm