

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 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,RIGHTMARGIN,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 reinitialize 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 (or COLORS)	>3A34	CALL LINK("COLOR",FOREGROUND,BACKGROUND) 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 (or COLOR)	>3A34	CALL LINK("COLORS",FOREGROUND,BACKGROUND) 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 (or PEEKC)	>2A94	CALL LINK("CPEEK",LOCATION,SIZE,STRING\$) Purpose: to look at CPU RAM.
CPOKE (or POKEC)	>2AEA	CALL LINK("CPOKE",LOCATION,STRING\$) Purpose: to load information into CPU RAM. Comment: This is an efficient alternative to the more usual CALL LOAD(address,value...).
DECHEX (or BAT) (or HEXDEC)	>277A	CALL LINK("DECHEX",STRING\$,NUMBER) Purpose: to convert numbers from decimal to hex. 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 (or DISPLY) (or DSPLA) (or DDISPLAY) (or DSPLY) (or PRINT)	>38DC	CALL LINK("DISPLA",ROW,COLUMN,MESSAGE\$) Purpose: to provide DISPLAY AT for graphics or text mode. Comment: If used with INIT, this command can "do windows," print slanted lines, etc.
DISPLY (or DISPLA) (or DSPLA) (or DDISPLAY) (or DSPLY) (or PRINT)	>38DC	CALL LINK("DISPLY",ROW,COLUMN,MESSAGE\$) Purpose: to provide DISPLAY AT for graphics or text mode. Comment: If used with INIT, this command can "do windows," print slanted lines, etc.
DSPLA (or DISPLA) (or DISPLY) (or DDISPLAY) (or DSPLY) (or PRINT)	>38DC	CALL LINK("DSPLA",ROW,COLUMN,MESSAGE\$) Purpose: to provide DISPLAY AT for graphics or text mode. Comment: If used with INIT, this command can "do windows," print slanted lines, etc.
DSPLAY (or DISPLA) (or DISPLY) (or DSPLA) (or DSPLY) (or PRINT)	>38DC	CALL LINK("DSPLAY",ROW,COLUMN,MESSAGE\$) Purpose: to provide DISPLAY AT for graphics or text mode. Comment: If used with INIT, this command can "do windows," print slanted lines, etc.

DSPLY (or DISPLA) (or DISPLY) (or DSPLA) (or DSPLAY) (or PRINT)	>38DC	CALL LINK("DSPLY",ROW,COLUMN,MESSAGE\$) Purpose: to provide DISPLAY AT for graphics or text mode. Comment: If used with INIT, this command can "do windows," print slanted lines, etc.
EA5	>2B3A	CALL LINK("EA5",DSKN.FILENAME\$) Purpose: to load in an EA5 program from XB (from cassette or from disk).
GOSPRT (or MOVE)	>3C1E	CALL LINK("GOSPRT") Purpose: to start sprite motion.
GPEEK (or PEEKG)	>307E	CALL LINK("GPEEK",LOCATION,SIZE,STRING\$) Purpose: to look at GROM or GRAM.
GRAPH (or MODE28) (or MODE32)	>3B54	CALL LINK("GRAPH") Purpose: to set up for graphics mode.
GRPCOL >	3A7E	CALL LINK("GRPCOL",FOREGROUND,BACKGROUND) Purpose: to change colors of character sets 1 through 12 for graphics mode.
HEXDEC (or BAT) (or DECHEX)	>277A	CALL LINK("HEXDEC",STRING\$,NUMBER) Purpose: to convert numbers from hex to decimal. 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 (or NEWSET)	>303A	CALL LINK("LOW") Purpose: to call new character set with true lower case.
MODE28 (or GRAPH) (or MODE32)	>3B54	CALL LINK("MODE28") Purpose: to set up for graphics mode.

MODE32 (or GRAPH) (or MODE28)	>3B54	CALL LINK("MODE32") Purpose: to set up for graphics mode.
MODE40 (or TEXT)	>3AF4	CALL LINK("MODE40") Purpose: to set up for text mode.
MOVE (or GOSPRT)	>3C1E	CALL LINK("MOVE") 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 (or LOW)	>303A	CALL LINK("NEWSET") Purpose: to call new character set with true lower case.
NOMOVE (or STSPRT)	>3C08	CALL LINK("NOMOVE") Purpose: to stop sprite motion.
NOQUIT (or QUITOF)	>29C6	CALL LINK("NOQUIT") 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 (or QUITON)	>29D6	CALL LINK("OKQUIT") (or QUIT) Purpose: to enable quit key.
PEEKC (or CPEEK)	>2A94	CALL LINK("PEEKC",LOCATION,SIZE,STRING\$) Purpose: to look at CPU RAM.
PEEKG (or GPEEK)	>307E	CALL LINK("PEEKG",LOCATION,SIZE,STRING\$) Purpose: to look at GROM or GRAM.
PEEKV (or VPEEK)	>3C34	CALL LINK("PEEKV",LOCATION,SIZE,STRING\$) Purpose: to look at VDP RAM.
POKEC (or CPOKE)	>2AEA	CALL LINK("POKEC",LOCATION,STRING\$) Purpose: to load information into CPU RAM. Comment: This is an efficient alternative to the more usual CALL LOAD(address,value...).
POKEV (or VPOKE)	>3C8A	CALL LINK("POKEV",LOCATION,STRING\$) Purpose: to load information into VDP RAM.
PRINT (or DISPLA) (or DISPLY) (or DSPLA) (or DSPLAY) (or DSPLY)	>38DC	CALL LINK("PRINT",ROW,COLUMN,MESSAGE\$) Purpose: to provide DISPLAY AT for graphics or text mode. Comment: If used with INIT, this command can "do windows," print slanted lines, etc.

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 > (or OKQUIT) (or QUITON)	29D6	CALL LINK("QUIT") Purpose: to enable quit key.
QUITOF > (or NOQUIT)	29C6	CALL LINK("QUITOF") Purpose: to disable quit key.
QUITON > (or OKQUIT) (or QUIT)	29D6	CALL LINK("QUITON") Purpose: to enable quit key.
READ (or RSEC)	>32CA	CALL LINK("READ",DRIVE,SECTOR,STRING1\$,STRING2\$) 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 (or READ)	>32CA	CALL LINK("RSEC",DRIVE,SECTOR,STRING1\$,STRING2\$) 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 (or VDPWTR)	>3CCE	CALL LINK("VDPREG",REGISTER,VALUE) Purpose: to allow a VDPWTR from XB.
VDPWTR (or VDPREG)	>3CCE	CALL LINK("VDPWTR",REGISTER,VALUE) Purpose: to allow a VDPWTR from XB.
VPEEK (or PEEKV)	>3C34	CALL LINK("VPEEK",LOCATION,SIZE,STRING\$) Purpose: to look at VDP RAM.

VPOKE (or POKEV)	>3C8A CALL LINK("VPOKE",LOCATION,STRING\$) Purpose: to load information into VDP RAM.
WRITE (or WSEC)	>32D4 CALL LINK("WRITE",DRIVE,SECTOR,STRING1\$,STRING2\$) Purpose: to write a single disk sector.
WSEC (or WRITE)	>32D4 CALL LINK("WSEC",DRIVE,SECTOR,STRING1\$,STRING2\$) Purpose: to write a single disk sector.
XXB	>269C CALL LINK("XXB") (or 1.5) Purpose: to verify that XXB has been loaded.