

CORRECTION- not covered in any update or appendix- please amend the text under DIM re integer arrays.

referred to on the entry for DIM:

100 DEFINT B(10) will produce a syntax error

You don't replace DIM, you add DEFINT like this:

```
100 DIM DEFINT B(10)
```

ADDENDUM FOR MYARC EXTENDED BASIC II

SUMMARY:

Pages 1-3 Installation Instructions

Pages 4-7 Version 2.0 variations:

CLS

RUN FILE,CONTINUE

PWD

CHDIR

!@P*

More details on OLD/SAVE/RUN
GRAPHICS(3)

amended: LINK will only link to sub-program names.

FILL

LIST

NOT SUPPORTED: HI RES MARGINS

REAL

SAVE...SPEECH

SPGET

CS1, CS2

Pages 8-13 Version 2.10 variations:

NEW: SIZE

REMOVED: FREESPACE

AMENDED; REAL NOW DEFREAL

Info: RAM USAGE TABLES

Info: ASSEMBLY USAGE DETAILS

Pages 14 - Version 2.11 variations:

New Disk utilities: 128KOSN, TIVDP

Manual amended: DEF only takes ONE parameter

OLD will NOT load DV80 files

Page 15- Extra from sjs:

DEFINT AND ARRAYS

Version 2.12 was a bug-fix version and no
addendum seems to have been issued.



MYARC Extended BASIC II
128K Operating System

INSTALLATION INSTRUCTIONS

Your new MYARC Extended BASIC II kit includes one of each as follows:

MYARC Extended BASIC II Command Module
MYARC Extended BASIC II Floppy Diskette
MYARC Extended BASIC II EPROM Semiconductor (IC)
MYARC Extended BASIC II Instruction Manual (240 pp)

Use of the command module and the floppy diskette are covered in the MYARC Instruction Manual.

The MYARC Extended BASIC II EPROM (Erasable Programmable Read-Only Memory) contains extensive and complex software necessary for the high-speed performance of MYARC's state-of-the-art Extended BASIC II and it must be installed as firmware into your MYARC 128 (or 512k) Memory Expansion Card. It replaces the EPROM now in your Card which does not contain the MYARC Extended BASIC II routines.

In order to maintain our present prices, we ask that you take the time to return your old EPROM to us. Thank you.

PROCEDURE FOR EPROM REPLACEMENT **(See Note on p.3)

1. Shutdown the console, PEB, and all other peripherals connected to your system.
2. Wait at least 2 minutes for power discharge, remove the PEB cover and lift out your MYARC 128K Memory Expansion Card from the PEB.
3. The 128K Memory Expansion Card is opened by separating the two "clamshell" halves at the edge OPPOSITE to the edge with the MYARC label. Note the two plastic catches at BOTH ends of this separation edge which lock the two clamshells together. You will need to depress BOTH catches, first at one end and then at the other end of the separation edge.

The separation is best done using a medium size screwdriver (1/4 to 3/8" wide) to depress each catch INWARD while simultaneously pushing apart the two clamshells at that end. At each end, start with the outer catch first.

After separation, lay the two halves on a table. The back side of the circuit board will be exposed.

-continue on other side-

The two clamshells are NOT identical - note which clamshell holds the circuit board and the orientation of the board in that clamshell.

4. Holding it by the edges, lift out the circuit board and place it on a protected surface, back-side DOWN and card-edge connector contacts TOWARDS you.

REMOVING EPROM FROM CIRCUIT BOARD

5. The EPROM is a large 28-pin IC and is located in the bottom right-hand quadrant of the board (nearest you) at location marked U26. The EPROM is further recognized by a small label (marked M.1x series) affixed over and covering the window in the center of the EPROM.

Note that the printing on the EPROM is in the same direction as all the other IC's on the board.

6. To avoid bent or broken pins etc., extreme care must be used in removing the EPROM from its socket. Unless you're using a 28-pin IC remover, insert a small screwdriver end (up to 1/4" wide) first at one end, between the EPROM and the socket and GENTLY pry up that end of the EPROM a slight amount. At the other end, similarly place the screwdriver between the EPROM and the socket and pry up that end about the same amount. Repeat the process alternately at each end so that the EPROM lifts up uniformly and easily from its socket.

INSERTING THE MYARC Extended BASIC II EPROM

7. After removing the new EPROM from its packing, orient it in the SAME direction that the old EPROM had been socketed i.e., with the printing in the same direction as all the other IC's.

WARNING: The EPROM will be irreparably damaged if inserted in the wrong direction and powered up.

8. Align all 28 pins of the EPROM into the 28 (rectangular) holes of the socket in the board. To get good alignment it may be necessary to adjust the EPROM pins, generally by carefully pressing them inward (towards the center) or otherwise as required, so that all 28 pins are well aligned to slide easily into the socket holes. Before pressing down check again that all 28 pins are properly aligned over the 28 socket holes.
9. After you're sure all pins are properly aligned, carefully press down, first on one end and then at the other end of the EPROM a little at a time so that after several alternate pushes the EPROM is seated completely in it's socket.

REPLACING THE CIRCUIT BOARD INTO THE CLAMSHELLS

10. Reversing the removal procedure, replace the circuit board, back-side up, into the proper clamshell. Adjust the circuit board so that two holes in the board fit over and into, the two supports protruding up from the clamshell.

11. Orient the other clamshell over the board. The card edge opening in the upper clamshell must be over the card edge connector of the circuit board and the small opening in the upper clamshell for the LED (lamp) must align with the LED on the circuit board.
12. Interlock the two small plastic hinges at the label-edge of the clamshells and then firmly push together each end at the separator edge. The plastic catches should snap into place with applied firm pressure.
13. Reinsert your upgraded MYARC Extended BASIC II, MEXP-1 Expanded Memory Card into your PEB and replace the PEB cover before system startup.
14. Please replace the old EPROM into its packing material and return it to MYARC at the following address:

MYARC, Inc.
241 Madisonville Road
Basking Ridge, NJ 07920

A few wraps of paper around the EPROM and packing material and/or inserted between 2 sheets of cardboard in a first class envelope should be adequate packing. We're banking on return of the the EPROMs to maintain our present pricing structure. Thanks again.

NOTICE

THE SOFTWARE CONTAINED IN THE MYARC Extended BASIC II EPROM IS COPYRIGHTED BY MYARC AND MAY NOT BE COPIED OR DUPLICATED IN WHOLE OR IN PART FOR ANY REASON WHATSOEVER.

ATTEMPTS TO COPY OR TAMPER WITH THE MYARC EPROM ELECTRONICALLY OR OTHERWISE WILL PERMANENTLY DAMAGE THE EPROM.

EPROMs THAT HAVE BEEN DAMAGED DUE TO SUCH TAMPERING ARE NOT COVERED BY WARRANTY AND WILL NOT BE REPLACED BY MYARC.

****NOTE:** Users who prefer to not undertake installing the MYARC Extended BASIC II EPROM may ship their MYARC MEXP-1 Memory Expansion Card PREPAID to MYARC together with the MYARC Extended BASIC II EPROM at the above address.

We will make the conversion at no cost to you and we will prepay return shipment to you. PLEASE ENCLOSE A SELF-ADDRESSED SHIPPING LABEL.

If you ship the Card and EPROM to us, we suggest you insure it for value. MYARC is not responsible for any loss or damage prior to actual receipt at our premises.



T.M.

MYARC EXTENDED BASIC LEVEL IV
VERSION 2.0

Thank you for purchasing MYARC Extended Basic II. Our gratitude to the many faithful TI users, who have been patiently awaiting delivery of their XBII orders. Your understanding has been greatly appreciated. Now that the anxiety of anticipation has ended, we sincerely hope that you will enjoy using MYARC XBII.

MYARC XBII is the most ambitious development for the TI99/4(A) to date. Although extensive error checking procedures were employed, there may still be some bugs. We shall endeavor to rectify any that may be found. After all, TI's Extended Basic was the result of three attempts.

The following addendum, covering Version 2.0, describes features over and above those described in the manual. Some features, not currently supported in Version 2.0 will be supported in Version 2.1. Once your registration card AND a blank floppy disk have been received, we will gladly provide the update to Version 2.1 without any additional cost. The updated Version 2.1 should be available.

Once again, thank you for your purchase and your continued support of products for the TI99/4(A).

CLS

Typing in CLS will cause the display to be cleared. CLS is the equivalent of CALL CLEAR.

RUN "FILE-NAME",CONTINUE

Embedded in a program, this command allows you to load and execute a program, while maintaining the same "variable" values.

PWD - Print Working Directory
and
CHDIR - Change Working Directory

Programmers have found working directories to be useful and convenient. Working directory pertains to OLD, SAVE, and RUN from the command mode. When specifying a file, for any of these three commands, if a period is not in a specified file name, working directory is prefixed to the file name. Therefore, if working directory is DSK1. and you specify: SAVE PROGRAM1 , XBII would try to save program in memory to DSK1.PROGRAM1 .

Upon power up, working directory is DSK1. . By issuing the CHDIR command, the working directory can be changed, to any name up to 15 characters.

OLD/SAVE/RUN

MYARC XBII uses VDP RAM more extensively (256 characters) than TI Extended Basic (114 characters). Therefore, when saving a basic program, XBII will switch to internal variable 254 format sooner than TI Extended Basic, because of less available VDP RAM.

In order to load all TI Extended Basic programs, in program image format, XBII may use the character definition table of VDP RAM as load space. The characters on the display will momentarily become undefined, but will be restored immediately after the program has been loaded. To prevent characters from being redefined in the future, perform a save after the program has been loaded.

MYARC XBII will automatically convert the format to internal variable 254, usable by both TI Extended Basic and MYARC XBII. Because of the extra space available given by using the character definition table, MYARC XBII is able to load some TI Basic programs that TI Extended Basic was not able to load.

!@P+
!@P-
!@P*

TI Extended Basic supports the commands !@P= and !@P- to speed up program prescan. MYARC XBII also supports these and has an additional command of !@P*. This new command stops prescan altogether. It differs from !@P- in that it would stop searching for variables, but would continue to search for the FOR-NEXT loops and other proper Extended Basic Syntax.

Because the new command stops prescan entirely at that point, it speeds program start-up time considerably. At the same token, EXTREME CARE should be used when using this new capability.

GRAPHICS(3) Mode (Bit Mapped Mode)

When entering Graphics(3) Mode, all available VDP RAM is needed, including some of the area used by the floppy disk controller. Therefore, upon invoking Graphics(3), XBII will close all files and perform a "CALL FILES(1)". In this mode, only one floppy disk file can be open at a time.

Another consideration must be given when using Graphics(3) Mode, that of running other programs. Because all available VDP RAM is used both for loading program files and displaying Bit mapped data, only loading internal variable 254 program files is allowed while running in Graphics(3) Mode. In order to save a program in internal variable 254 format, just issue the command: SAVE "DSKX,PGMNAME",INTERNAL

DEF

User defined functions not supported in Version 2.0 WILL BE SUPPORTED IN VERSION 2.1

DEFINT

Integer variable types not supported in Version 2.0 WILL BE SUPPORTED IN VERSION 2.1

FILL

Optional parameter/character pattern not supported in Version 2.0 WILL BE SUPPORTED IN VERSION 2.1

****Version 2.0 currently fills space in fully.**

LINK

Passing of parameter list through assignments and references not supported in Version 2.0 WILL BE SUPPORTED IN VERSION 2.1

****Version 2.0 currently supports only linking to subprogram names.**

MARGINS

In High-Resolution Mode, margins have little value in Extended Basic and therefore WILL NOT BE SUPPORTED.

REAL

***Version 2.0 considers all variables as real. VERSION 2.1 WILL SUPPORT REAL WHEN INTEGER VARIABLES ARE INCLUDED.*

RECTANGLE

***Version 2.0 dictates pure, vertical or horizontal shapes. VERSION 2.1 WILL SUPPORT GENERAL PURPOSE PARALLELOGRAMS.*

SAVE

Speech not supported In Version 2.0 WILL BE SUPPORTED IN VERSION 2.1

SPGET

Not supported in Version 2.0 , WILL BE SUPPORTED IN VERSION 2.1

SUBSUBEXITSUBEND

User defined subroutines not supported in Version 2.0 WILL BE SUPPORTED IN VERSION 2.1

LIST

LIST X- Replaced with LIST X,E or LIST X E

LIST X-Y Replaced with LIST X,Y or LIST X Y

LIST -X NOT SUPPORTED

Because MYARC XBII is a disk based system, it has been decided not to support cassette I/O. Should there be sufficient demand for this capability, Version 2.1 may support this also.

Lastly, although Version 2.0 has been tested on hundreds of Extended Basic programs, errors may exist. Therefore, your assistance in eliminating errors would be appreciated. Rather than phoning, please mail us examples of errors in the form of listings and/or floppy diskettes.



MYARC EXTENDED BASIC VERSION 2.10

Your patience in waiting for V2.10 has been greatly appreciated. We feel the wait was worth it in delivering to you such a fine product for the T199/4(A). If you are an owner of V2.0, please use this addendum and disregard the old. Thank-you for buying MYARC. We are working hard to keep your trust.

The following is an addendum of features covering V2.10 which are either not correct or are not covered in the manual. Following the addendum is a technical discussion on the architecture of V2.10 BASIC. Hopefully, this will cover all details necessary to program efficiently in both BASIC and 9900 assembly language.

CLS

Typing in CLS will cause the display to be cleared. CLS is the equivalent of CALL CLEAR.

RUN "FILE-NAME",CONTINUE

Embedded in a program, this command allows you to load and execute a program, while maintaining the same "variable" values.

PWD - Print Working Directory
and

CHDIR - Change Working Directory

Programmers have found working directories to be useful and convenient. Working directory pertains to OLD, SAVE, and RUN from the command mode. When specifying a file, for any of these three commands, if a period is not in a specified file name, working directory is prefixed to the file name. Therefore, if working directory is DSK1. and you specify: SAVE PROGRAM1, XBII would try to save program in memory to DSK1.PROGRAM1.

Upon power up, working directory is DSK1.. By issuing the CHDIR command, the working directory can be changed, to any name up to 15 characters.

OLD/SAVE/RUN

MYARC XBII uses VDP RAM more extensively (256 characters) than TI Extended Basic (114 characters). Therefore, when saving a basic program, XBII will switch to internal variable 254 format sooner than TI Extended Basic, because of less available VDP RAM.

In order to load all TI Extended Basic programs, in program image format, XBII may use the character definition table of VDP RAM as load space. The characters on the display will momentarily become undefined, but will be restored immediately after the program has been loaded. To prevent characters from being redefined in the future, perform a save after the program has been loaded.

MYARC XBII will automatically convert the format to internal variable 254, usable by both TI Extended Basic and MYARC XBII. Because of the extra space available given by using the character definition table, MYARC XBII is able to load some TI Basic programs that TI Extended Basic was not able to load.

SIZE

The SIZE command will display the amount of program, variable, and string space available to the program.

FREESPACE

The FREESPACE command has been replaced by the size command which provides more and better information covering memory utilization.

LIST

Please use the following commands in place of those in the manual:

LIST X- Replaced by LIST X,E or LIST X E
 LIST X-Y Replaced by LIST X,Y or LIST X Y
 LIST -X Replaced by LIST 1,X or LIST 1 X

MARGINS

In High-Resolution Mode, margins have little value in BASIC and therefore are not supported in this mode.

REAL

Version V2.10 supports both INTEGER and REAL NUMBERS. However, the variable definition name for REAL types has been changed from REAL to DEFREAL.

RECTANGLE

Version 2.10 supports all modes of drawing pure vertical and horizontal shapes. In addition, general purpose parallelograms are supported in the TYPE 1 mode (i.e. drawing the perimeter).

FILL

Optional character pattern is not supported in V2.10.

GRAPHICS(3) Mode

When entering Graphics(3) Mode, all available VDP RAM is needed, including some of the area normally used by the floppy disk controller. Therefore, upon invoking Graphics(3) mode, XBII will close all files and perform a "CALL FILES(1)". In this mode, only one floppy disk file can be open at a time. Another consideration must be given when using Graphics(3) Mode, that of running other programs. Because all available VDP RAM is used for display, only loading of internal variable 254 program files are allowed. In order to save a program in internal variable 254 format, just issue the command: SAVE "DSKX.PGMNAME",INTERNAL.

! P+

! P-

! p*

TI Extended Basic supports the commands ! P+ and ! P- to speed up program prescan. MYARC XBII also supports these and has an additional command of ! P*. This new command stops prescan altogether. It differs from ! P- in that ! P- allows syntax checking to continue, whereas with ! P*, it does not. Because the new command stops prescan entirely at that point, it speeds program start-up time considerably. At the same token, EXTREME CARE must be taken when using this capability.

EXTENDED BASIC II RAM USAGE

The following three diagrams depict memory usage of both CPU and VDP RAM by XBII.

BASIC II CRAM MAP

>2000 I

II

III

IV

8K

UNUSED

RAM DISK

VDP/SPEECH
ROUTINES

BASIC INTRP

USER ASSM

BASIC INTRP

I/O BUFFERS

VALUE STACK

>4000

8k

>6000

MEMORY MANAGEMENT ROUTINES

>8000

24k

>A000

BASIC

VARIABLE

STRING

BASIC

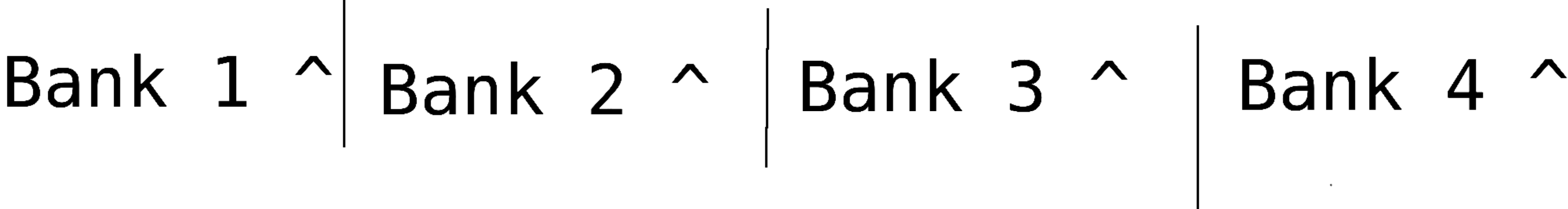
PROGRAM

SPACE

SPACE

INTRP

>FFEB



BASIC II VRAM MAP

*****				*****			
* PATTERN MODE *		* TEXT MODE *		* BIT-MAP MODE *		*****	
*****				*****			
* >0	SIT	0		* >0	SIT	0	
* >2FF		767					

* >300	SAL	768					
* >37F		895					

* >380	CT	896					
* >39F		927					

* >3A0	EMPTY	928		* >3BF		959	

* >7FF		2047		* >7FF		2047	

* >800	PDT/SDT	2048		* >800	PDT	2048	
* >FFF		4095		* >FFF		4095	

* >1000	EMPTY	4096		* >1000	EMPTY	4096	

BASIC II VRAM MAP

VDP REGS:	VREG0	VREG1	VREG2	VREG3	VREG4	VREG5	VREG6	VREG7	SMOTION
-----	*****	*****	*****	*****	*****	*****	*****	*****	*****
GRAPHICS	MODE	MODE	SIT	CT	PDT	SAL	SDT	S/T	SMT (@>7A5C)
PATTERN	>00	>E0	>00	>0E	>01	>06	>01	>17	xxxxx
			>0000	>0380	>0800	>0300	>0800	BLK/CY	>7A5C
TEXT	>00	>F0	>00	>0F	>01	xxx	>xx	>17	-----
			>0000	>03C0	>0800	>xxxx	>xxxx	BLK/CY	-----
BITMAP	>02	>E0	>0E	>FF	>03	>76	>03	>17	xxxxx
			>3800	>2000	>0000	>3B00	>1800	BLK/CY	>7A5C

ASSEMBLY LANGUAGE USAGE

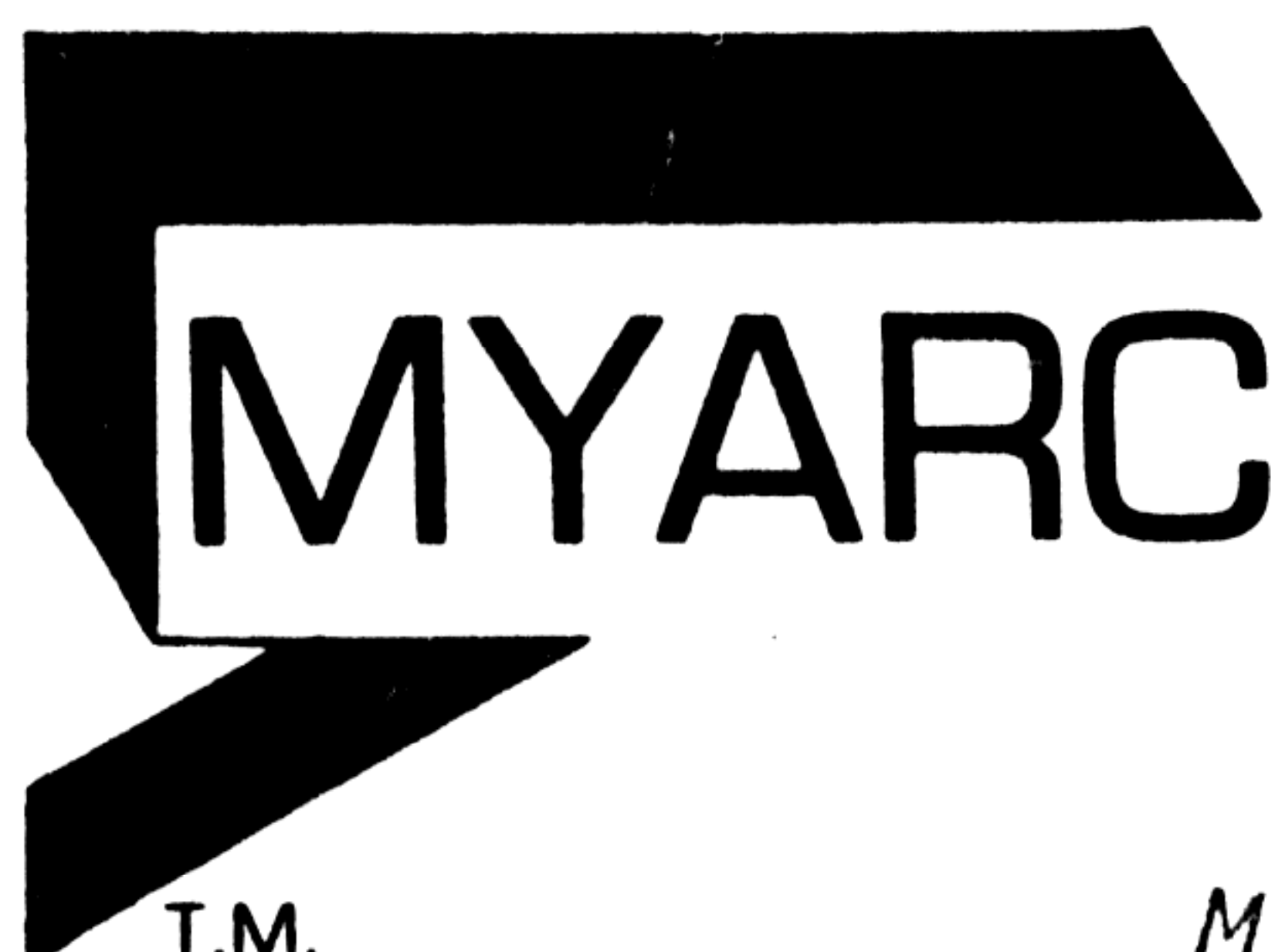
As the CPU RAM usage depicts, a little over 1K-bytes of RAM is wasted in order to load assembly language programs at the exact same location as TI XB. In order to use this extra area, immediately perform a CALL LOAD(8194,32,130) after your CALL INIT.

ASSEMBLY LANGUAGE SUPPORT

All assembly language support routines listed on pages 415-416 of the Editor/Assembler manual are supported except for the routines dealing with manipulation of data in VDP RAM. These routines are NEXT, COMPCT, GETSTR, MEMCHK, VPUSH, VPOP, ASSGNV, VGWITE, GVGWITE. If these routines are invoked using an XML, they will return as a NO-OP. The reason for not allowing the user to invoke these routines (even though they are within XBII) is because no VDP RAM contains no data to be manipulated. All data is stored in CPU RAM, and therefore there is no need for these routines. A description of these routines are given in chapter 17 in the E/A manual.

In addition, the assembly language loader in XBII is similar to that of the E/A Loader, in that compressed as well as non-compressed object code can be loaded. The following symbols are also predefined and can be used in your software by using the REF directive: PAD, GPLWS, SOUND, VDPRD, VDPSTA, VDPWD, VDPWA, SPCHRD, SPCHWT, GRMRD, GRMRA, GRMWD, GRMWA, SCAN, XMLLNK, KSCAN, VSBW, VSBR, VMBW, VMBR, VWTR, DSRLNK, LOADER, NUMASG, NUMREF, STRASG, STRREF.

As another enhancement, as the XBII manual states on page 126, when a "TYPE 5" assembly language program is used in an OLD or RUN statement, the program is loaded and executed immediately. Lastly, V2.10 supports all the returns given in section 24.11 of the E/A manual. However, if the alternate return as given in section 24.11.3 is used, the word in memory location 1/28300 should not be altered, as it contains a return linkage to XBII.



T.M.

MYARC EXTENDED BASIC II ADDENDUM VERSION 2.11

Version 2.11 has removed all bugs reported to MYARC (in release 2.10). In addition, two new utilities and a demonstration program have been added.

The first utility is a file named "128KOSN". The file differs from 128KOS in that it attempts to determine if XBII files 1-6 have been already loaded and therefore will only load "XBII7". This means that if you keep power on the RAM DISK and keep it in the 128K CPU ram mode, you only need to load "XBII7" once. You can go into and out of XBII from then on by only loading XBII7. This of course makes load time almost seven times faster. To use this capability you must rename file "128KOS" to something like "128KOSOLD" and rename "128KOSN" to "128KOS". You are now set.

The second utility is a file named "TIVDP". This utility has been developed to solve some of the problems involving compatibility with loading assembly language from XB. The utility causes VDP ram and VDP registers to be the same as used in TI XB. To invoke this utility, a program statement should be as follows:

CALL INIT::CALL LOAD("DSK1.TIVDP")::CALL INIT

This will cause the screen to go blank and therefore should be used in a running program. To return VDP ram and registers to XBII mode, simply type "NEW" and enter.

The demo program is called "DEMO3M" and shows some of the capabilities of XBII. To run it, type RUN DSK1.DEMO3M.

Lastly, a few notes. Because XBII uses windows to present information, you may find a program or two that assume scrolling to occur in columns 1, 2, and 31 and 32. This will not happen. It works as TI defined it to operate on their own 9918. Also, when using CALL SAY, the computer will spell any alpha word string that is not in the library and say "UHOH" if the word string is non alphanumeric. The DEF function can only support one and not multiple parameters. The RUN and OLD commands cannot accept DV80 format files. These are enhancements for the new MYARC ADVANCED BASIC for the TI99/4A compatible 9640 computer.

THIS PAGE NOT FROM MYARC- WRITTEN BY SJS 2020

MYARC XBII VERSION 2.11 / 2.12

The manual did not indicate how to use INTEGER numbers with ARRAYS.

Using INTEGER numbers reduced memory usage a lot, and produces a small speed increase.

I was advised directly that you could use INTEGER numbers with arrays as follows:

```
100 DEFINT DIM A(12)
```

=====

The different memory mapping required to use the extra facilities of MXB2 produced a small number of incompatibilities with existing Extended Basic programs, especially those that used elements of Assembly code. .

I was privately advised that Myarc MBX II was not compatible with Extended Basic programs which had embedded machine code, that is, those produced with utilites ACE or SYSTEX.

Assembly programs that used interrupts or which use >83C4 could also cause difficulties.

There were some problems with loading machine code programs which assumed various vdp conditions when using CALL LOAD.

It was often sufficient to transfer the DF80 files to EA5 format program files. Sometimes manually setting the graphics mode would help