

BUGBYTES

August/September 1997

Editor's Note

Dennis Remmer
dennis@dstc.edu.au

The next meeting will be held at Garry Christensen's house, 18 Zammit St, Deception Bay, on Wednesday night 30th July, starting at 7:30p.m.

Hi everyone. This wintery issue will find you knee-deep in problems and solutions for the V9T9 emulator, as well as a comprehensive article on the infamous MBX speech recognition system that never quite made it to the shelves. Some other tidbits round off the issue.

Best Regards...

A stylized, handwritten signature in black ink, appearing to be 'Dennis Remmer'.

Desperately need help with V9T9 utils & module transfers

(Editor: This is an interesting discussion that recently occurred on the TI list from the Internet. I've edited the original posting with the responses in italics to make it easier to follow. I've also taken the liberty to expand any acronyms.)

Original posting from:
Randy
randyb@utilicom.com

*Response from:
Brian Tristam Williams
POLAR@global.co.za*

The following long message is a culmination of all my unsuccessful attempts over the last few weeks to get V9T9 and its utilities to work properly. I have invested a great deal of time printing and reading all of V9T9's docs as well as a lot of fruitless trial and error. To make a long story short, I'm at my wit's end and I'm going to give up for good on V9T9 and delete all the TI files I downloaded from the Internet if I don't make a breakthrough soon.

My only success with V9T9 6.00 (I got the one from ftp.io.org/.../opanit) is that I've had no problems using any of the module images, DOADs (Disks On A Disk), or ROMs that I fortunately already had on my PC so I could test things out. Overall I'm impressed by the quality of the emulation. But my problems are many:

1) I can't get any of V9T9's utilities to work with DOADs no matter what I try (haven't tried FIADs [Files In A Directory] yet).

Neither can I, Randy, but others report success at this. The utilities are little DOS executables that perform rather simple tasks, and I'm sure that someone could re-write them to be more tolerant... I have a 486-DX4 100Hz with DOS 6:22 and Win95, by the way. Perhaps we could try an older version of DOS - I'm sure that when Ed wrote these utilities he wasn't using the latest DOS.

I know the DOADs are correct because they work fine when accessed inside of

V9T9. I remember hearing a few other people had the same trouble, but I don't recall the solution. I am using DOS 6.20 on a Pentium 75 with 32 megs, BTW.

For instance, the Pirate Adventure disk is PIRATE.DSK, has a volume name of PIRATE, and one file named PIRATE. Using V9T9 I have no trouble using

this or any other DOAD. However, no matter what I try with any of the utilities such as TIDIR, TICHKDSK, etc I get "no files match PIRATE.DSK, continuing" if I type "TIDIR PIRATE.DSK".

When I type "TIDIR PIRATE:" I get the volume label followed by the same error message.

When I type "TIDIR PIRATE" I get:

Error on file (path)PIRATE.DSK

Bad file structure"

If I use TICHKDSK * in the DISKS (DOADs) directory every file gives the error: "Not a v9t9-compatible file, or a damaged v9t9 file". I know the files are all valid because they work under V9T9.

CONTENTS

Editor's Note	1
V9T9 Desperation!	1
The MBX System	5
3D on the TI	8
TI Mailing List Digest	9
Bad Jokes!	9



Without the use of these utilities, all of the file transfers I need to do are impossible. I've tried running both from a Windows 3.1 DOS shell and pure DOS, with and without EMS memory managers, plus every uppercase, lowercase, and logical combination of image/file names I could think of.

Another problem was with TI2XMDM. Every time I used that utility such as: "TI2XDM MUNCHMNG.BIN ." the utility locked up my PC.

You can't do that! You can only use TI2XMDM on disk files - FIADs or DOADs. And since there is a problem using the utilities on DOADs, you HAVE to use a FIAD. So, TI2XMDM will only work on files with V9t9's header. I think that the best way to transfer a program such as MUNCHMNG.BIN to the TI is to leave it as-is. Use a terminal program on the TI and save it as a PROGRAM file - I'm just guessing here, so I'd welcome corrections. Another thing for me to look at.

Maybe I don't have the syntax down right and that's part of the problem, but given that nothing else works, I think there may be something wrong with the whole archive. Has anyone else successfully used the utilities in 600V9T9.ZIP from the Opanit ftp archive?

2) When I follow the instructions in TRANSFER.TXT on how to use FORTH to send the "TRANS" program over to the TI, as soon as I type "TRANSFER" as instructed (or even "HELP") I get a blank screen and nothing happens. What am I doing wrong?

There's a strange one. When I first got V9t9, I managed to run the TRANSFER program, but I've lost familiarity with it. Now, I tried this, and I ALSO just got a blank screen - I'll have to look into that.

As an alternative, I tried Xmodeming TRANS to the TI and use the resulting DIS/FIX 128 file with the CALL LOAD/LINK commands from the Receiver program on the TI. That didn't work either.

I type FORTH.BAT, select the FORTH module and get the FORTH prompt (which says "type HELP for help" among other things). When I type "TRANSFER" (without the quotes of course) I get

a blank screen. Same thing happens if I type HELP. The PC hasn't locked up because I can still select other modules, exit with CTRL-BREAK, etc. However, I can't get the TRANS program over to the TI as described in TRANSFER.TXT

So I tried transferring the TRANS file (a FIAD, I guess) in the DISK directory using ProComm and Mass Transfer. The file successfully got to the TI as a DIS/FIX 128. I tried using the CALL LOAD and CALL LINK statements from the TI's Receiver program I typed in by themselves on this TRANS file, but I got a "BAD CHARACTER IN xxx" on the line with the CALL LOAD.

TRANS isn't a DIS/FIX 128 file. It needs to be a DIS/FIX 80 file of size 64 (sectors). I also couldn't use The FORTH transfer program, so I had to use Extended BASIC transfer programs that I wrote, called OBTRANS (on V9t9) and OBRECV (on the TI). The program was generic enough to transfer any DIS/FIX 80 file. I'll check it out again for you.

BTW, I looked at a byte listing of the TRANS file and I noticed it contained what appeared to be a hex dump in ASCII format. I have no idea what the Forth program does, but I suppose it converts the ASCII into binary and sends one byte at a time to the TI to be stored in a binary file usable by CALL LOAD/LINK. Maybe since the file I transferred was still ASCII that's why I got the "BAD CHARACTER" error? If I converted it to binary before transferring it, would it work then?

No - the file is an uncompressed assembly-language object file (gee, I've learned SOMETHING about assembly ;), which IS a hex-dump. Leave it as an ASCII hex-dump. I think your problem is that it's supposed to be a DIS/FIX 80 file.

3) Is it possible to take module dumps (images) that work with V9T9, transfer them to the TI, use an Extended Basic loader program, and have them work on the TI as if they were real modules? If yes, is there a generic loader program I can get that will work on all module images?

4) Some TI dealers have sold module images on disk, supposedly with permission, that work just like a real module on

the TI. Are these module images in the same GROM and Bank C & D segments as V9T9 would use? Can they be transferred to V9T9, processed with XMDM2TI, and then added to MODULES.INF? Or are these not "pure" module segments so they must be used as disk files with the Extended Basic loader provided?

How does this loader program work? Unlike V9T9 modules which have G, C, and D extensions for where the image is loaded, all of the module images on disk do not follow this format. The loader program only lists the first segment, so how does the loader program know what filename comes next and where to put it? Are the smarts to this loading process in the LOAD: code in each of these modules?

For example, I have the module image on disk for Treasure Island. The files on this disk have the names "TISLAND", "TISLANE", and "TISLANF". All are of the PROGRAM type. This module image has a loader program in Extended Basic which is easy to understand except for a few mysterious numbers in CALL LOAD which I have no documentation for. What does CALL LOAD(8196,254,0) do? Or how about FOR S=1 TO 4::CALL LOAD(-S*6144,0)::NEXT S ? Finally, there's a CALL LINK ("LOAD","DSK1.TISLAND") statement. Besides the CALL INIT, I can type the 5 statements above and have the Treasure Island module run successfully. I know that CALL LINK loads the TISLAND file somewhere into memory and executes code at the LOAD: entry point. I have no idea how it knows the filenames of TISLANE and TISLANF and where in memory to load them. I don't suppose TISLAND is the GROM segment, TISLANE is the C bank, and TISLANF the D bank?

For instance, I couldn't Xmodem these three files over to the TI, run them through XMDM2TI, rename TISLANE to TISLANG.BIN, TISLANE to TISLANC.BIN, and TISLANF to TISLAND.BIN, edit MODULES.INF appropriately, and then run it in V9T9? That's what I was hoping I could do, but I somehow doubt it's that easy. I have a feeling these images have special loading code with the other filenames hard-coded inside. The files are sort of daisy-



chained. If this is the case, it is easy to convert these modified images into something usable by V9T9?

One critical piece of information buried in V9T9's manual was that the GROM segment always has AA hex as its first byte. However, no indication was given on how to identify a misnamed bank C file from a bank D one. Is there a way besides trial and error?

5) If I understand right, V9T9 FIADs, DOADs, and modules do not have the 128 byte TIHEADER, but all files transferred to the TI must have a TIHEADER?

The files in the DOADs have exactly the same header as TI diskettes do.

FIADs have the same length header as XModem transfers (which you refer to as TIHEADERS), but they differ in content. The FIAD headers contain the name of the file, and if you change the filename in DOS so that it doesn't match the header, V9t9 will return an error message. FIAD names, like the TI, can be up to 10 characters in length. If they are longer than eight characters, V9t9 will put the last two in the extension, such as "TI-ARTIS.T". If you try to use XMDM2TI on a file such as "TIWRITER.ARK", you will get an error message, as V9t9 will try to convert this 11-character filename into a TI file called TIWRITERAR. You will need to rename the file to either TIWRITER.AR or just TIWRITER.

Don't even try XMDM2TI to DOADs. As you've said, the utilities have a problem with DOADs. Use FIADs to transfer the files, then use Disk Manager to copy them to a DOAD.

Most of the files on ftp sites with either .arc or .ark extensions are XModem transfers. These have a header containing the word TIFILES (your TIHEADER). The header is also exactly the same length as V9t9's FIADs.

**You can see whether the file is a FIAD or and XModem transfer by opening it up in Windows 3.11's Notepad.Exe. If it is an XModem file, you'll see the word TIFILES at the beginning. If it is a FIAD, you'll see the name of the file in the header. Alternatively, in DOS, you can just use the DOS command TYPE to*

check the header. The easiest method is Windows 95's 'Quick View'.

There is a potential improvement for V9t9: I don't know WHY ed decided to use a different header for his FIADs - he could've just made it so that V9t9 would simply access TIFILES-type XModem transfers in its FIAD directories. Then you could just download a file and stick it straight into one of V9t9's FIAD directories, without having to use XMDM2TI.

Therefore, before sending any file which works with V9T9 to the TI I need to run it through TI2XMDM first?

Only if you're using a terminal program on the TI which uses XModem, and a terminal program on the PC, or in V9t9, which does the same. If you're transferring from V9t9's emulation to the TI, you don't need to do this. For instance, you can save a text file in V9t9's TI-Writer by saving your file to a filename called RS232.BA=3D2400, and then load this file on your TI's TI-Writer by loading from the same file (I think - lemme check it out).

You can also save BASIC files from V9t9 by typing in SAVE RS232.BA=2400 and using OLD RS232.BA=2400 on your TI. Again, you wouldn't have to use the TI2XMDM program. Only use it if you're using terminal programs to do the transfers.

If I transfer converted text files to the TI, is it important to make sure the TI file type is DIS/FIX 80?

I'm sure that TI-WRITER would accept it as such? Again, I'd have to check this out.

How about a transferred module image - does it have to be of PROGRAM type in order to work with CALLLOAD/LINK?

In general, I'd say my biggest problem with V9T9 and all the TI files I have is the lack of detailed explanations about the various file formats. I'm never sure when I need a TIHEADER, a PROGRAM or DIS/FIX type, what type of module bank I have, or when a file off the Internet is a FIAD or ready to be Xmodemed binary file for the TI. Is there something that explains all of these things without getting too bogged down

in the gory details? It would be enough if I could visually inspect a hex byte listing of any TI file sitting on my IBM and know enough to determine the file type and format.

Whew! I think those are all of my questions. If I could get V9T9's utilities working and if I understood what I can and can't do regarding transferring module images to and from the TI, I'd be a happy camper. Right now I'm just a tired, frustrated camper.

Any help would be greatly appreciated, even if it's just the answer to one of my many questions. As Princess Leia might say "Help me, TI-99ers, you're my only hope." Thanks.

That's is for now, but here is what's on my list to check out for you (just to remind me...):

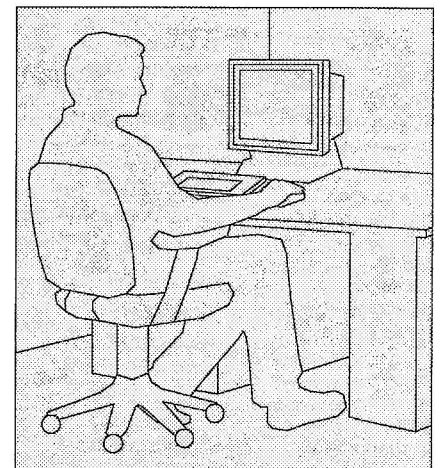
**Try the utilities on DOADs with an older version of DOS (I don't have one at the moment)*

**Get FORTH to run again, and find out why it's crashing*

**Check if you can transfer a DIS/VAR 80 or DIS/FIX 80 file from TI-Writer on both machines*

**Try transferring MUNCHMNG.BIN to the TI*

**Get my OBTRANS and OBRECV files back from wherever they are again - I haven't needed them since I transferred all of my modules and disks.*



(ED: And the follow-up! ...)

OK, here are some results to the things I had to try two days ago:

**Try the utilities on DOADs with an older version of DOS*

I still haven't gotten my hands on an older version.

**Get FORTH to run again, and find out why it's crashing*

I can't seem to get the TRANSFER program to operate consistently under FORTH. Sometimes it loads, and sometimes it doesn't. You might find that it freezes up, and then you press <CTRL>-<F12> to restart FORTH, and you try again, and it works. One note about the file 'Forth.Cnf': Initially, the lines pertaining to the RS232 access, "RS232/1 = 2,4", are preceded by '#'. This means that they are ignored by FORTH, and you have no RS232 access. You HAVE to remove the # sign from the line which pertains to RS232/1, and you have to find out which COM: port on your PC that you have linked to the TI, and what IRQ it employs. For example, if you are using COM1: for your mouse and you have COM2: linked to your TI, and your COM2: port uses IRQ 4, you need to set this line to "RS232/1 = 2,3". This is my situation.

Anyway, just forget about FORTH. It has NEVER worked for me. Since I know that you already have Extended BASIC in V9t9, because you downloaded the Zip file that I put together, you might as well do the transfer IN V9t9! Do it like this (assuming that the files 'Utils.Cnf', 'Utils.Bat' and 'Utils.Inf' in your V9t9 directory are still more or less the same as when you downloaded them):

1. Make sure that the 'RS232/1' setting in 'Utils.Cnf' contains the correct references to COM:port and IRQsetting, then run 'Utils.Bat'.

2. Type in Ed Swartz's program, and save it on your TI as DSK1.RECEIVE :

```
80 REM Receiver for TRANS
object code from PC
```

```
90 REM Legal baud rates:
300, 600, 1200, 2400, 4800,
9600

100 INPUT "Enter the baud
rate: ":B

110 OPEN #1:"RS232.BA=" &
STR$(B) & ".DA=8.PA=N",
UPDATE, VARIABLE 81, INTER-
NAL

120 OPEN #2:"DSK1.TRANS",
OUTPUT, DISPLAY, FIXED 80

130 PRINT "Receiving..."

140 INPUT #1:A$

150 IF A$="" THEN 200

160 PRINT #2:A$

170 PRINT ".";

180 PRINT #1:"O"

190 GOTO 140

200 CLOSE #2

210 CLOSE #1

220 CALL INIT

230 CALL LOAD("DSK1.TRANS")

240 CALL LINK("TRANS")
```

3. Go into Extended BASIC (or preferably Super Extended BASIC, since I know that you have it) from 'Utils.Bat', and type in this program in V9t9:

```
80 REM Transmitter for TRANS
object code to TI

90 REM Legal baud rates: 300,
600, 1200, 2400, 4800, 9600

100 INPUT " Enter the baud
rate: ":B

110 OPEN
#1:"RS232.BA="&STR$(B)&".DA
=8.PA=N",UPDATE,VARIABLE
81,INTERNAL

120 OPEN #2:"DSK3.TRANS",
INPUT,DISPLAY,FIXED 80

130 PRINT "Transmitting..."

140 IF EOF(2)THEN 200 ELSE
INPUT #2:A$

150 PRINT #1:A$

170 PRINT ".";

180 INPUT #1:B$

190 GOTO 140

200 CLOSE #2

210 PRINT #1

220 CLOSE #1
```

I wrote this program as closely as possible to Ed's RECEIVE program. Save it in V9t9 as DSK3.TRANSMIT.

4. Run both of the programs. Now you have the RECEIVE program running on your TI, and the TRANSMIT program running on your TI :) - Make sure that your machines are connected correctly, then enter the baud rate on both machines (9600 worked fine for me, but you can also enter 4800, 2400, 1200, 600 or 300), pressing <ENTER> on your TI first.

5. Assuming that the TRANS file is in the DISK directory of V9t9, which was configured as 'DSK3.' in 'Utils.Cnf', the programs SHOULD both start running, without any problems - I just tried this. You will see a period for every record that is transferred and saved. After a few lines of periods, the program will end on V9t9, and you'll have to be patient on the TI side, because it will be busy loading the TRANS program.

6. IF everything is OK up until this point, you should exit V9t9 with <Ctrl>-<Break>, and then, if you're in V9t9's directory, type in "RECV". This will run the PC's receiver program under DOS. By this time your TI should have started running the program TRANS. Now follow the instructions, and you SHOULD be able to transfer all of your modules.

7. Send them to me ;) !

**Get my OBTRANS and OBRECV files back from wherever they are.*

OK, you don't need those anymore, now that you have the above two programs. I couldn't find the original two programs, which were freshly written on both machines, but the above program makes use of Ed Swartz's original RECEIVE program. What I learned from this little adventure, which I didn't realise when I first typed in the program is this: When I wrote my own program on BOTH sides, I found that the TI did not have any way of telling the PC to stop sending information. I found that while the TI was writing the first record to disk, the PC would just keep sending, resulting in a buffer overflow on the TI's RS232, which would give you an error message at the next "INPUT #1:A\$" statement. This baffled me, and at the time I learned



a lot about RS232 cables (!). But the cabling didn't solve the problem. My eventual solution back then was to introduce a 'CALL SOUND' delay onto the PC side, so that it would give the TI time to write the record to diskette and wait for the next record, before it was sent. This was a really ugly way of doing things, and I didn't see why it was necessary when 'SAVE' and 'OLD' operations via RS232 worked flawlessly... Then, last night, I checked out ED's program again - the program on the TI simply sends after each receive! That is, if you look at Ed's program, after every input statement, the TI sends back an "O" character to the PC, which I've made provision for on the PC, and THEN waits for the next record. My questions to those knowledgeable on the subject:

- Is this called 'handshaking'? Allen, are you reading this?
- Has anyone out there successfully gotten the TRANSFER program in FORTH to work? Even when it does run for me, I get I/O errors - I presume that's because of the absence of this 'handshaking'.

**Check if you can transfer a DIS/VAR 80 or DIS/FIX 80 file, using TI-Writer on both machines*

[Reponse from jgrosrou@nbnet.nb.ca]: Using the terminal program in Windows 3.11 on the PC and TI-WRITER on the TI you can transfer to the PC at 9600 baud(19200 on the Geneve) but you can only transfer from the PC to the TI at 600 baud. Apparently this is an inherent limitation of TI-WRITER. The other settings are odd parity and 7 data bits. I assume that you will have the same limitation transferring from v9t9 using TI-WRITER on both machine. The above information appeared in an article by Charles Good in the September 1995 issue of Bits, Bytes & Pixels the newsletter of the Lima Ohio User Group.

**Try transferring MUNCHMNG.BIN to the TI*

This will have to wait - I've taken my TI apart to try and see if I can organise a composite video output.

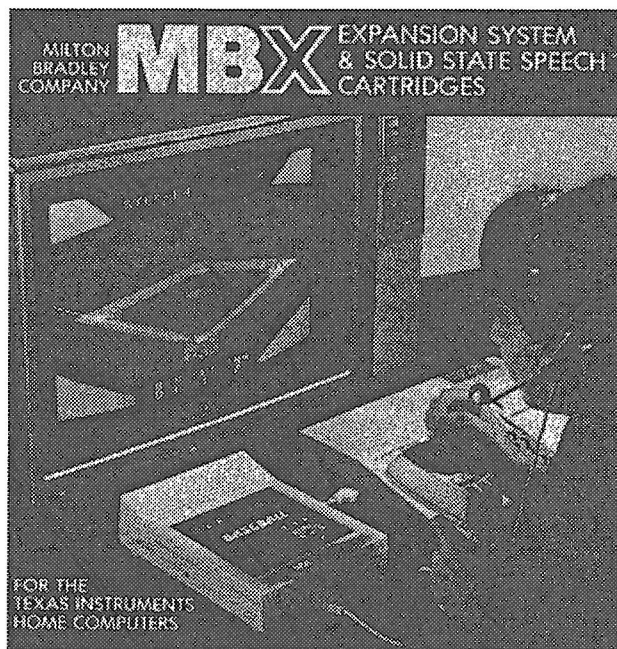
Never-released TI Peripherals: The Milton Bradley MBX System

Charles Good
good.6@osu.edu

(Editor: originally published in the Lima UG Newsletter - December 1990, and reproduced online by Rob Patten at http://overplex.irdg.com/mbx_rev.htm)

This device was (and maybe still is) literally years ahead of the competition when first introduced to the public at the January 1983 Consumer Electronics Show. When attached to the TI99/4A it allows speech recognition with specific Milton Bradley game and educational modules. The user speaks instructions into a microphone, and the 99/4A understands the spoken words and responds accordingly. With the MBX system, our old fashioned 99/4A's can do tricks that even the most sophisticated modern home game machines can't do. Voice recognition is NOT available even today for Nintendo and Sega game systems. These days you can find voice recognition hardware costing hundreds of dollars advertised in Computer Shopper for use with MSDOS and MAC computers. In T.I.'s last complete price list of 99/4A products published in June 1983, the MBX system lists for \$129.95. It's too bad only only about 300 were ever made!

Although the title of this article might suggest that the MBX system was made by T.I., this is not so. The MBX was manufactured and sold by the Milton Bradley Company. T.I., under license from Milton Bradley, manufactured and sold the specific software modules designed for use with the MBX. The MBX system comes packaged in a box with the the same kind of "photograph of the product on a black background with white letters" style found on 99/4A console boxes. The actual MBX hardware is in the same gray plastic used for the most

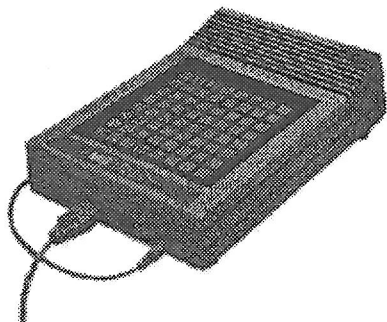


recent 99/4A consoles and official T.I. cassette program recorders. As a registered 99/A owner, I received by mail in early November 1983 from T.I. (not from Milton Bradley) an advertisement describing the MBX and MBX specific software. Apparently Milton Bradley intended to have the MBX on store shelves for the 1983 Christmas season, but canceled all further production after BLACK FRIDAY. There is no serial number on my recently purchased used MBX, but it bears a sticker that says "MBX Control number 8310". This may mean that my unit was manufactured in the 10th week of 1983. My guess of 300 MBX units actually produced is based on the very limited availability of this product for sale at TI shows I have attended and in the possession of T.I. owners known to me, as well as the fact that UNISOURCE once advertised that they had 200 MBX's for sale.

There are three parts to the MBX "system", the control box, the joystick, and the headset/microphone. The heart of the system is of course the control box. It measures 10 x 7.5 x 2.5 inches and includes its own built in speech synthesizer. This box plugs into the joystick AND the cassette recorder ports of the 99/4A console. The MBX system is designed to be used with just the console and specific software cartridges. There is no provision in any of the MBX software modules for disk usage. Since one of the MBX connections occupies the cassette jack, you can't use a cassette recorder

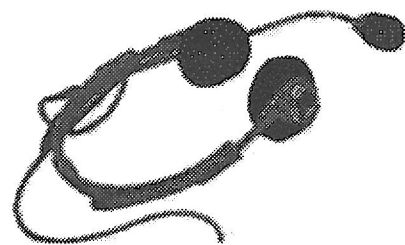


either. You must disconnect the regular speech synthesizer to use MBX. To hear speech, the two speech synthesizers cannot coexist.

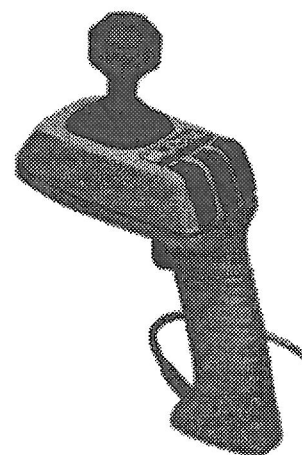
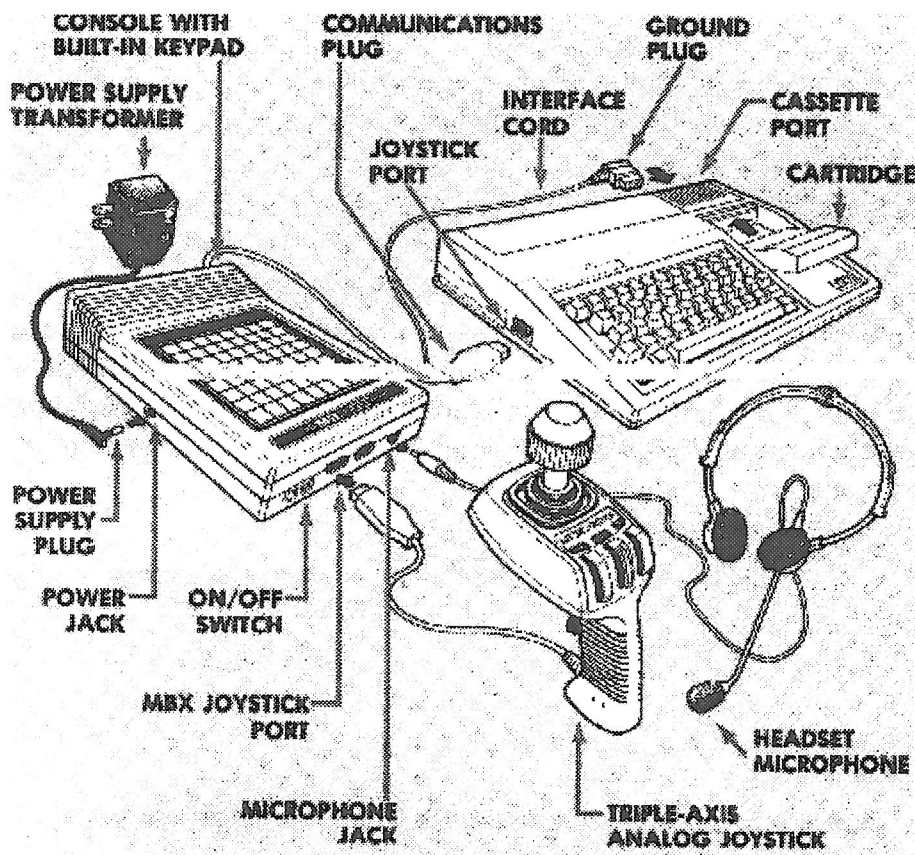


The control box has a side port for the required AC power source. On the front of the control box are two 9 pin male D ports for joysticks, a jack for the headset/microphone, and an on/off switch. When you slide this switch to the ON position the MBX control box responds by saying "ready" in a well modulated female voice. This voice, and all speech generated by the MBX system, comes from a speaker at the top back of the control box, not from the monitor speaker. Music and other non-speech sounds con-

tinue to be heard from the monitor's speaker. Only spoken words (synthesized speech) are heard from the MBX system's speaker. You have to turn on the MBX before you turn on the console in order for the 99/4A to recognize the presence of the MBX. When activated, the MBX system disables the FCTN/0 QUIT console keypress. On the top of the control box is a 64 position membrane keypad. The top row of 8 keys on this keypad functions in the same way with all the MBX software modules that utilize this keypad. These top row keys include RESET, VOLUME UP (the volume of the speech coming from the MBX's built in speaker, not the music and sounds coming from the monitor speaker), VOLUME DOWN, MIC (toggles on and off the ability of the microphone headset to "hear" spoken words), YES, NO, PAUSE (stops game action), and GO. The action of other 56 positions on the control box keypad is specific to the particular software module in use. A very decorative keypad overlay comes with those software modules designed to utilize the rest of the MBX control box keypad. These overlays slip easily and snugly over the top of the keypad.



The headset superficially resembles a set of "walkman" earphones, but in fact contains no earphone speakers. The things that cover your ears are just pads. The microphone is positioned in front of your mouth and its position is adjustable. Physically the headset unit is flimsy. The wire leading to the microphone is thin and subject to stretching and damage at the point where it enters the adjustable microphone arm of the headset as the microphone arm is adjusted back and forth. Fortunately a handheld microphone designed to plug into a cassette recorder will also work with the MBX if the headset microphone breaks. The advantage of the headset over a handheld microphone is that the headset allows easy two handed manipulation of the special MBX joystick.



One joystick comes as standard equipment with each MBX system. A second joystick is listed in T.I.'s last 99/4A price list for \$29.95 and can be plugged into the second joystick port on the control box. This would give each of two players their own separate joystick. In actual use of the MBX software modules a second joystick isn't really needed. Only one player at a time uses the joystick. The

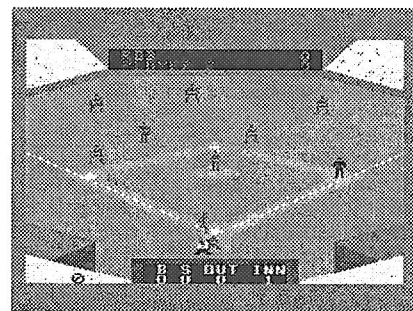
two joystick ports on the control box respond the same. There is no "joystick #1" and "joystick #2" as there is with the 99/4A console. The MBX joysticks are very fancy and cannot be used by themselves directly from the 99/4A's joystick port. Likewise, you can't use regular joysticks from the MBX console. Movement of the MBX joystick handle is very smooth. The device is described in promotional literature as a "triple-axis analog control that allows 360 degree object rotation and left to right and front to back proportional control of all movements." The word ANALOG suggests infinitely variable control. The MBX joystick's arm appears to produce the same kind of 8 direction movement typical of joysticks. The "analog" infinitely variable control is the rotating knob on the end of this joystick arm. With some MBX games this knob will rotate the object under control to face any direction, for example to orient a gun prior to shooting. In the MBX baseball game this knob controls the force of a batter's swing. Minimum swing power results in a bunt. A trigger style fire button is included with the MBX joystick, as well as three other buttons. These three buttons resemble mouse buttons and have specific purposes when using specific MBX software modules.

How does MBX allow the 99/4A to respond to voice commands? At the beginning of each session with an MBX software cartridge that allows voice recognition as an option, the user is asked if he wants to use voice recognition. This is always optional. All the MBX cartridges can be used WITHOUT voice recognition by using the keyboard and/or the MBX keypad for input instead. If voice recognition is chosen, the user is asked which commands are to be given in voice. It is possible to use voice for some commands and the console keyboard or MBX keypad for other commands, or to have all non joystick input by voice. The computer then directs the user to speak the possible commands (big, small, left, right, pencil, pen, centerfield, shortstop, etc) into the MBX microphone. This "voice training" of the MBX to recognize the user's voice patterns is repeated twice. Voice patterns are stored digitally on chips inside the MBX for the duration of the session, until the MBX is reset or shut off. This voice pattern storage is probably similar to that of some modern

telephone answering machines. My home answering machine does not store the greeting message on cassette tape. Instead, my "This is the Good household answering machine..." message that greets incoming calls is stored on a chip and played to callers every time I don't answer the phone quickly enough. As with the MBX, I can quickly erase my "stored on a chip greeting" and replace it with another on my answering machine. An MBX user can use any word desired for a particular command, as long as the user is consistent in using this word. For example, in CHAMPIONSHIP BASEBALL a user can speak the imaginary name of a fielder when asking for a particular fielding position. During voice training the computer can ask the user to speak the word "shortstop" and the user can reply "Tony". As long as the user remembers that Tony is playing shortstop, the game will work OK.

After voice training the game begins and the computer responds to sounds it hears via the MBX microphone. Users have to be careful to ONLY speak when they want the computer to perform some action. Casual conversation by the user can result in unexpected things happening as the computer interprets some of this conversation as specific spoken commands. The solution to this problem is to turn off the MIC using the MBX keypad when response to voice commands is temporarily not desired. A small symbol continuously on screen indicates the ON/OFF status of the microphone.

How well does it work? How reliable is MBX's voice recognition? It is about 80-90% reliable. Sometimes the MBX either totally ignores a verbal command, or the command is incorrectly interpreted as a different verbal command. Part of the problem is that during the excitement of game play, a player's voice may sound different than it did during voice training. In CHAMPIONSHIP BASEBALL it can be very annoying to command a throw to "second" and instead see the ball thrown to "centerfield".



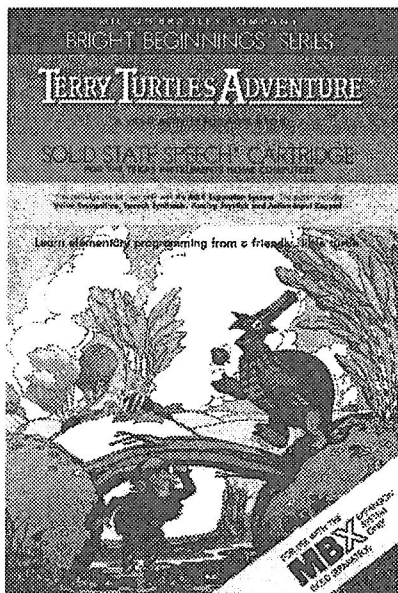
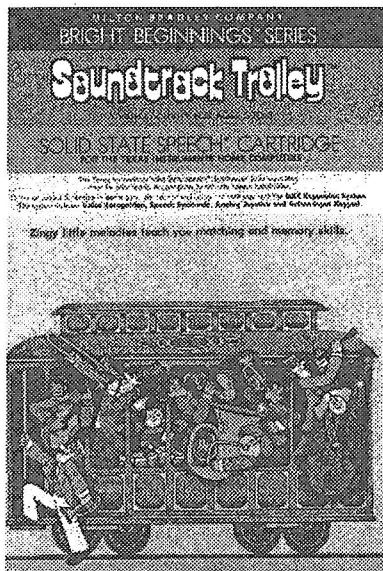
All voice commands can instead be activated from the 99/4A keyboard of the MBX's keypad with almost 100% reliability. All the modules designed for use with the MBX, even those that absolutely REQUIRE the MBX, can be used totally without voice recognition. For really serious accurate game play, one should bypass the MBX's voice recognition feature. My testing panel is divided in their preference for voice recognition. Meaghan, my 5 year old daughter, likes to use voice recognition. I think she finds voice easier than reading the MBX overlay or memorizing complex 99/4A key-press sequences. Ian and Colin, ages 12 and 9, both prefer not to use voice recognition. High scores are important to these two serious game players, and such scores are easier to obtain with accurate game control.

What software is available? The following cartridges by Milton Bradley were specifically designed for use with the MBX expansion system. All these include speech synthesis and many also allow voice recognition. The speech synthesis of these software modules (but not speech recognition) can be accessed using the regular TI speech synthesizer without using the MBX system. They were officially released by T.I. in 1983 and 1984. The last (June 1983) 99/4A catalog published by TI lists these modules for \$50 and \$60. I have seen some of the "MBX system required" modules listed by TRITON in the past for as little as \$3. Currently they are all available from L.L. Conner Enterprise for \$15. Quoting from the booklet TEXAS INSTRUMENTS HOME COMPUTER PROGRAM LIBRARY that came packaged with many TI modules sold in late 1983:



"The BRIGHT BEGINNING SERIES includes four games which teach elementary programming, music, and other learning concepts. Ages 4-8."

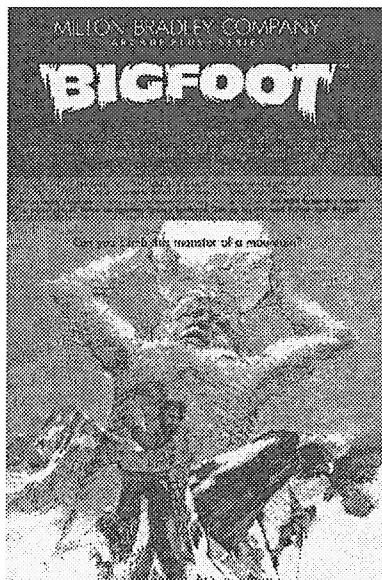
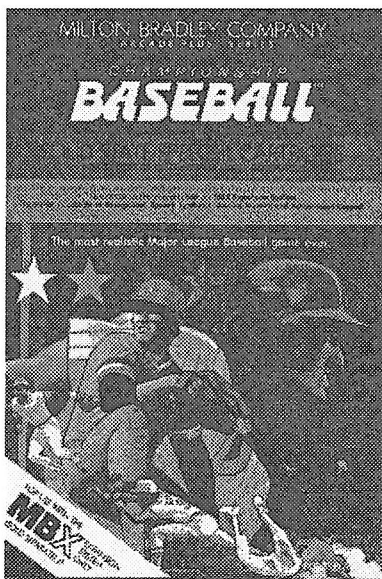
HONEY HUNT
SOUNDTRACK TROLLEY
TERRY TURTLE'S ADVENTURE
(MBX system required)
I'M HIDING (MBX system is required).



"The ARCADE PLUS SERIES has six arcade style games that take you from home town ball parks to meteor belts far, far away."

CHAMPIONSHIP BASEBALL (MBX system required)

SEWERMANIA
SUPER FLY
SPACE BANDITS
BIGFOOT
METEOR BELT



I have been told that Barry Boone has written some software that will allow programming the MBX and its non standard joystick. Such software would turn the MBX into something much more significant than a game enhancement. So far, this software has not been made available to others.

Most software cartridges allow use of the MBX joystick as an option. METEOR BELT requires two MBX joysticks if the MBX system is used in a two

player game. All software cartridges allow limited use of the MBX keypad for RESET, PAUSE, and GO.

3D on the TI

Brian Tristam Williams
POLAR@global.co.za

A while ago I asked for information about doing bitmap graphics in assembly, as I was interested in starting a little 3D going on the TI. Now, I'd been going through my garage and found some ZX Spectrum computer (anyone in the US heard of it?) cassettes, and there was a 3D viewer there! If the Spectrum could do it, then why on earth not the TI? I have never seen or heard of any kind of 3D programs on the TI-99/4A. Has anybody else?

Anyhow, I've wanted to get properly into TI assembly language, but I've been impatient, so I tried a little something with 'Super Extended BASIC's built-in 'Drawnplot' programme. The module is included in the 'TIMRaD2' file that's out there. It's a program to take a little 20-point wireframe 3D house with garage (really simple) and rotate it around either the Y- or the Z-axes, the degree to which is provided by the user. In addition to this message, I will try to post the program in 'V9t9' FIAD-format to Allen Unueco's ftp site, under the ftp.premier-web.com/pub/systems/ti99/uploads directory. Allen, if you don't like me posting miscellaneous stuff like this, please let me know! The file is called '3DHouse.'

I don't know how many of you have your TIs actually up and running with the 'Super Extended BASIC' module, so I didn't really see a need to post it as a TIFILES file. As for 'PC99', I'm not sure what format is useful.

Here is the program in text-format. I would have liked to animate the rotation, but that just isn't possible with 'Drawnplot' - if you use a 'CALL LINK("SHOW")' in the program, you can't have a 'GOTO' afterwards, in order to loop. Make sure that you enter "CALL FILES(2)," then "NEW," before running it.




```

100 ON WARNING NEXT
110 DG=PI/180 :: D=100
120 CALL CLEAR :: CALL INIT
:: CALL DRAWNPLOT :: CALL
LINK("GCLEAR"):: P=20 :: DIM
X(20),Y(20),Z(20),X1(20),Y1
(20)
130 FOR A=1 TO P
140 READ X(A),Y(A),Z(A)
150 NEXT A
160 DATA 0,0,0,0,0,80,40,0,
80,40,0,0
170 DATA 0,40,0,0,40,80,40,
40,80,40,40,0
180 DATA 15,0,0,15,20,0,25,
20,0,25,0,0
190 DATA 20,50,20,20,50,60
200 DATA
40,20,0,60,20,0,60,0,0
210 DATA 40,20,20,60,20,
20,60,0,20
220 REM Translation
230 FOR A=1 TO P
240 X(A)=X(A)-10
250 Y(A)=Y(A)-30
260 NEXT A
270 REM INPUT "x-axis rota-
tion: ":XAR
280 INPUT "y-axis rotation:
":YAR
290 INPUT "z-axis rotation:
":ZAR
300 FOR A=1 TO P
310 B=X(A)*COS(ZAR*DG)+
Y(A)*SIN(ZAR*DG)
320 C=X(A)*-SIN(ZAR*DG)+
Y(A)*COS(ZAR*DG)
330 X(A)=B :: Y(A)=C ::
B=X(A)*COS(YAR*DG)+Z(A)*SIN
(YAR*DG):: C=X(A)*-SIN(YAR*
DG)+Z(A)*COS(YAR*DG)::
X(A)=B :: Z(A)=C :: NEXT A
340 REM Perspective trans-
formation
350 FOR A=1 TO P
360 X1(A)=X(A)*D/(Z(A)+D)
370 Y1(A)=Y(A)*D/(Z(A)+D)
380 NEXT A
390 RESTORE 440 :: FOR A=1 TO
28
400 READ B,C

```

```

410 CALL LINK("MOVE",
X1(B)+128,96+Y1(B)):: CALL
LINK("DRAW",X1(C)+128,96+Y1
(C))
420 NEXT A
430 CALL LINK("SHOW")
440 DATA 1,2,2,3,3,4,4,1
450 DATA 5,6,6,7,7,8,8,5
460 DATA 1,5,2,6,3,7,4,8
470 DATA 9,10,10,11,11,12
480 DATA 5,13,13,8,6,14,
14,7,13,14
490 DATA 15,16,16,17,17,4
500 DATA 18,19,19,20,19,16,
20,17,15,18

```

TI Internet Mailing List Digest Version Ready

Tom Wills
twills@theriver.com

BIG NEWS!!! The TI99 Digest is ready. There have been many requests for this service, and now thanks to my ISP, The River Internet Communication, and the Webmaster, Marcus Needham, the TI99 Digest is up and running. For those who want to use the Digest version of this list server, do as follows:

- Address your email request to **MAJORDOMO@THERIVER.COM**
- Leave the subject blank
- The subscribe to the server, simply put **SUBSCRIBE TI99-DIGEST** as the email message.
- To unsubscribe from the server, simply put **UNSUBSCRIBE TI99-DIGEST** as the email message.
- Send your email message from the account that you wish the messages to be sent to, or the account to be from which you wish to unsubscribe from. That's all there is to it. When you subscribe to the Digest, please save the message which you will receive acknowledging your subscribing. It contains unsubscribing information. If you have any problems in either subscribing to or unsubscribing from

the TI99 email list server, please send me an email to twills@theriver.com.

Some Bad Jokes :)

Source Unknown

Two Eskimos sitting in a kayak were chilly, but when they lit a fire in the craft it sank-proving once and for all that you can't have your kayak and heat it, too.

Two boll weevils grew up in South Carolina. One went to Hollywood and became a famous actor. The other stayed behind in the cotton fields and never amounted to much. The second one, naturally, became known as the lesser of two weevils.

A mushroom walks into a bar, sits down and orders a drink. The bartender says, "We don't serve mushrooms here." The mushroom says, "Why not?! I'm a fun guy!"

A three-legged dog walks into a saloon in the Old West. He sidles up to the bar and announces: "I'm lookin' for the man who shot my paw."

This guy goes into a restaurant for a Christmas breakfast while in his home town for the holidays. After looking over the menu he says, "I'll just have the eggs benedict." His order comes a while later and it's served on a huge fancy chrome plate. He asks the waiter, "Whats with the fancy plate?" The waiter replies, "There's no plate like chrome for the hol-landaise!"

Very early one morning two birds are sitting at the side of a large puddle of oil. They see a worm on the other side. So... the one flies over and the other one swims through-which one gets to the worm first? The one who swam, of course, because "Da oily boid gets da woim."

A neutron goes into a bar and asks the bartender, "How much for a beer?" The bartender replies, "For you, no charge."

Two molecules are walking down the street and they run in to each other. One says to the other, "Are you all right?"



"No, I lost an electron!" "Are you sure?"
"I'm positive!"

Did you hear about the Buddhist who refused his dentist's Novocaine during root canal work? He wanted to transcend dental medication.

TI Brisbane User Group

Committee

President: Rex Jones
3396 1662

Vice-President: John Campbell
3351 3107

Secretary: Garry/Col Christensen
3888 4857 / 3284 7783

Treasurer: Val Jones
3396 1662

Editor: Dennis Remmer
3870 5710
dennis@dstc.edu.au

Disk Librarian: John Reynolds
3357 9758

Tape Librarian: Bruce Campbell
3351 3107

Book Librarian: Trevor Campbell
3351 3107

Module Librarian: Bruce Campbell
3351 3107

All items within this newsletter may be reprinted providing the source and author are acknowledged. The views expressed in articles published in TIBUG are those of the author and do not necessarily reflect the views of the Editor, Committee Members or Members of this User Group. All items, articles, programs etc. in this Newsletter are believed to be public domain.

Contributions to TIBUG are invited from both members and non-members. Articles for inclusion in the succeeding bi-monthly newsletter are required at least two weeks before the monthly meeting and may be included in that newsletter at the discretion of the Editor.

Most original articles by members of TIBUG in this newsletter are available on disk and are available to other User Groups on request.

Submissions of articles, reviews, comments and letters from members is encouraged, however the Editor asks that those submitting keep the following in mind:

Submissions should be about the TI Community in particular, computers in general, or of sufficient general interest. The preferred media is computer file, preferably in ASCII (Text) or Microsoft-Word compatible format, submitted on MacIntosh or IBM-compatible floppy disk or via Electronic Mail to the Editor. Handwritten submissions are acceptable but please remember that they have to be retyped. Other submissions, such as typed, printed or photocopied are welcome but must of reproducible quality.

Submissions are best sent directly to the Editor:

Dennis Remmer - TIBUG Editor
PO Box 30, Toowong QLD 4066
(Email: dennis@dstc.edu.au)



TI Brisbane User Group
PO Box 3051
Clontarf MDC, QLD 4019
Australia

