# BITS, BYTESEPIKELS

Cc+ 1771

LIMA 99/4A USERS GROUP

Volume 7, 48



# A Special Ancient History Issue Ti's 99/2 97/7 and 79/8 Computers The poes that got away!

THE TI "99/2 BASIC COMPATER" a hands on review by Charles Good Lina Ohio User Group

Did you know that in 1983 II came very close to selling a \$100 home computer based on the 9995 CPU, the same CPU planned for the never released 99/8 and found today in the Somewe? Officially called the 99/2 BASIC COMPUTER, this chose entry level machine was envisioned by TI as an inempensive amons of introducing the public to home computing and as a learning tool to teach BASIC programming. The 99/2 produces a silent black and white display on any TV and comes with a version of BASIC that runs 10 TIMES FASTER than TI BASIC. This is really amazing speed, and compares favorably with the speed of BASIC on some modern MS-DOS computers. I recently acquired one of these "never released" computers and have been playing with it extensively. A photo of the 79/2 appears in the December 1984 Issue of MICRUPENDIUM Showing a little white paper sticker in the upper left side of the computer bearing the notation "Hex OK". Ny 99/2 has a similar paper sticker, apparently indicating a quality control text of the 99/2's Hexbus interface. I personally tnow of only two other 99/2's in private hands.

THE LITERATURE, AND A SHORT HISTORY OF THE 99/2:

It was the Summer of 1982 and the home computer price wers were just beginning. II decided to produce a computer to compete directly with the \$100 Timex/Sinclair 1000, and in 4 1/2 months they had working wire wrap prototypes of the 99/2 for display at the January 1983 Consumer Electronics Show. Photos of the 99/2's on display at this show appeared in March 1983 issue of 99er Home Computer Magazine showing a machine specific command module inserted in its rear expansion port. It is interesting that these show a difference from later production prototypes like the mme I own. Later prototypes like mine have the Hexbus port on the extreme right side of the back of the computer, while the January 1983 CES photos show the Hexbus port in the middle of the back panel.

Product development and the FCC certification process continued. I have a copy of the final "TI 99/2 MAIN LOSIC ROARD SCHEMATIC" containing the signatures of Mark Jander, project design engineer, and several other TI people involved in the project. The last of these signatures is dated 5/5/83. The time between initial product conception, FCC certification, and the first limited production run was about 8 months. An advertising campaign was developed and a two page ad showing Bill Comby holding a 79/2 under his chim actually appeared in the Pday 1983 issue of Popular Science (inside front cover). An article was written by the 79/2

project development team for BYTE magazine (BYTE, June 1983, pages 128-134) that gives lots of technical details about the 99/2 computer and includes a photograph of the computer's circuit board. A review comparing the 99/2 other very cheam computers of that time appeared in COMPUTERS & ELECTROMICS, June 1983, pp. 48-51. The 99/2 is also illustrated and briefly described in the March 1983 COMPUTE! (p. 30-31) and the May 1983 issue of POPULAR COMPUTING (p. 28).

And then the price wars really took off! According to ads in 99er Home Computer Magazino (April and May 1903) the price of a new 99/4A was \$150. By mid-March the \$70 cash price of the Timex/Sinclair 1000 was reduced even more with a \$15 rebate (Computers & Electronics, June 83, p. 51). In June 1903 II initiated a \$50 rebate bringing the price of the /4A to \$100. This was supposed to be the selling price of the 99/2, so just as full scale production was about to begin the May issue of 99er HCM announced that the /2 project was put om indefinite hold. The last 1983 published photo of the 99/2 I know about is on the inside back cover of the July 1983 issue of ENTHUSIAST 99 (volume 1, \$2). In this advertisement for itself, ENTHUSIAST 99 shows the 99/2 just harely visible in the background shadow, right next to a 99/4 (without the A). How appropriate!

SUMMARY OF THE 99/2's FEATURES:

--9995 CPU running at 10.7 MHz. The 99/4A's 9900 CPU runs at about 3.3 MHz.

--32K ROM with built in BASIC closely resembling TI BAGIC except that color, sound, and joysticks are not supported.

--4K RAM plus 256K "scratch pad RAM" directly on the CPU chip. 32K RAM and/or ROM memory expansion is possible.

--Flicker free black and white TV display of 32 characters by 24 lines. This is done using a "direct accery access" video controller chip that uses CPU memory. There is no "video display processor" with its own VDP memory as there is on the /4A and Geneve.

--Cassette interface compatible with the cassette interface of the 99/4A.

--Hexbus port for use with all Hexbus peripherals.

GENERAL DESCRIPTION:

If you removed the right side of the 99/4A, the part with the cartridge port and the top cooling vent holes, what you have left would just be about the length and width of the 99/2. The /2 is, however, only about 1/2 as thick as the /4A. The /2's keyboard very closely resembles that of the /4A in size, number, and position of keys. Namever, the 99/2 has "chicklet" keys that are not nearly as mice to use as the

full depression keys of the 99/4A. Although it IS possible to touch type using the 99/2, it isn't easy. You need a VERY heavy touch to depress the 99/2's keys. There are 48 keys including a CTRL and FCTN key. Where the alpha lock key is located on the /4A there is a RRFAK key on the 99/2. The BREAK key does exactly what FCTN/4 does on the /4A (and the /2), it stops the running of a BASIC program and returns to command mode. The FCTN key, in combination with the top row of number keys produces the same results as on the /4A (DEL, IMS. ERASE, etc) except for FCTM/=. This combination remots the /4A to the title screen but does nothing on the /2. (I consider the lack of a FCTN/= QUIT to be an improvement.) To exit BASIC with the /2 you either have to turn off the computer or type BYE, which I consider an improvement. As far as I can tell, the CTRL key on the /2 does nothing at all. Apparently this CTRL key can only be accessed from assembly language and not from 99/2 BASIC.

All ports are on the back of the /2. As you face the rear panel, from left to right are the following:

--EXPANSION PORT where cartridges and expansion annoty plug in. According to the BYTE article the expansion port has all system control address and data bus signals and allows for expansion with RAM, ROM, or I/O cartridges. There is a 32K expansion memory space available that can be shared by RAM and ROM. The March 1983 99er has photos of a command would cartridge pluged into this port. Both the March 1983 99er and the BYTE article mention two command module programming tutorial titles specifically made for the 79/2. A 32K RAM memory expansion was also supposed to plug into this expansion port. I know of nobody who has actually ever seen these command modules or the memory expansion device.

--TV. An RF modulator is built into the console, so the signal that comes out of this port is modulated. This means you MUST use a TV to display the output. You can't use a composite monochrome (or color) monitor to improve resolution, although a composite monochrome video signal is available via the expansion port.

-- "CASSETTE IN" and "CASSETTE DUT". These take take cables with "miniature phono plugs" at both ends and connect to the cassette recorder's earphone and microphone jacks. These cables did not come with my 99/2, but I had no trouble finding the correct cables at my local Radio Shack store. These ports support OLD CS1 and SAVE CS1. There is no automatic control of the cassette motor, but this causes on no trouble. The 99/2 does not support sound, so you hear mothing at all when saving and loading from cassette. The screen display goes blank during the actual save or load, and this helps you keep track of what is going on. The only error message is "NO DATA FOUND". There is no "EMBAR METECTED IN DATA" seconds as there is in the /4A. This caused me some confusion when I first tried to SAVE and them verify a typed in program. I kept turning up the cassette recorder volume in response to the NO DATA FOUND message. right up to maximum volume, without success. What I should have done was to reduce the volume. I have managed to find the correct volume setting on my TI Bata Recorder and my 99/2 OLDs and SAVES quite reliably to and from cassette. The 99/2

BMGIC programs I save to cassette can be successfully loaded into and run from my 99/4A.

--"CH3 - OFF - CH4" This three position sliding switch sets the video output for channel 3 or 4, or turns the computer off. This recessed switch is very difficult to get at, and is the only way other then unplugging the power supply to turn off the computer. A more convenient on/off switch would have been nice. I often leave the computer "on" for hours and just turn off the TV. There is no automatic video blanking, so when you turn on the TV several hours later your display is still there waiting for you.

--POWER. The power transformer (TI model ACT700) connects here. The two wire connector on the end of the transformer is apparently unique to the 99/2. It is a little white flat thing and I have seen nothing similar elsewhere.

--"CC PORT" This is the Hexbus port. The "CC" designation on the back of the 99/2 apparently refers to TI's CC40 computer and is meant to indicate that the CC40's peripherals can connect to this port. ALL features of TI's never released HEXBUS INTERFACE for the 99/4A are supperted by this port, including some features not mentioned anywhere in the 99/2 documentation. It is a good thing I have the 79/4A HEXBUS INTERFACE USER GUIDE or I would have missed some of these undocumented features. You can SAVE, OLD, LIST to make from this port, as well as OPEN #1:"HEXBUS...." and then 79/INT #1 or INPUT #1 for complete file management. Some of the undocumented features include OPEN #1:"HEXBUS.CA.n" to IMPUT CAtalog information from a mafertape or disk device number "n", and OPEN #1:"HEXBUS.TR.n" to IRansfer raw binary data between the computer and hexbus device n.

VIDEO DISPLAY AND GRAPHICS:

The only display available is in black and white and is comparable to the 99/4A's TI BASIC screen, with 24 lines that can contain up to 32 graphic patterns or 28 keyboard typed characters per line. The other graphic modes available on the /4A are not supported. The TV display is very clear, and does not show the annoying flicker or diagonal line interference commonly produced by the Timex/Sinclair 1000 computer.

Characters corresponding to ASCII 0-127 are stored in RGN and can be displayed on screen. Only uppercase letters and usual special characters and digits (ASCII 32-96, 123-126) can be entered directly from the keyboard, but the other ASCII characters, including lower case letters, (ASCII 77-122) can be displayed using PRINT CHRS(XX), CALL HOMM, and CALL VCHAR. The lower case letters are actually the same stupid small upper case letters normally displayed as "lower case" by the 97/4A. ASCII 0-31 are predefined graphic shapes (lines, open and closed squares and rectangles, etc).

ASCII 127 looks really strange. On the 99/40 this is the BEL character and prints as a blank space. On the 99/2 this character looks like a little black round face with short logs. The two eyes and straight south show as uncolored

(white) pixels. Because there is no CALL CHAR in 97/2 BASIC you cannot define your own custom graphic shapes, so this strange shape is probably included for use in games. I'll bet the 97/2 firmware author who created this was having fun.

TI-99/2 BASIC:

When you PRESS ANY KEY TO BEGIN from the title screen you are presented with the following:

PRESS

1 FOR TI-99/2 BASIC

I suspect that there is provision for command modules to add other items to this menu. When you press 1, the computer tells you TI-99/2 BASIC READY.

99/2 BASIC contains all the features of TI BASIC except those relating to color, sound, joysticks, and custom graphic shapes. The following is a list of 99/2 BASIC's reserved words, most of which should be familiar to you.

ABS	
APPEND	L0 <b>6</b>
ASC	ACHL
ATN	HEM
BASE	NEXT
BREAK	NUM
BYE	NUMBER
CALL	OLD
CHR\$	ON
CLEAR	OPEN
CLOSE	OPTION
CON	0UT <b>PUT</b>
CONTINUE	PEEK
COS	PERMANENT
DATA	PCKE
DCL	POS
DELETE	PRINT
DIM	RANDONIZE
DISPLAY	READ
EDIT	REC
ELSE	RELATIVE
END	REM
E <b>OF</b>	RES
EXP	RESE <b>QUENCE</b>
FIXED	RESTORE
FOR	RETURN
GCHAR	RND
60	RUN
GOSUB	SAVE
50 <b>70</b>	SE6\$
HCHAR	SEQUENTIAL
IF	SGN
INPUT	SIN
INT	SQR
INTERNAL	STEP
KEA	STOP CTD#
LEN	STR\$
LET	SUB
LIST	TAB

TAN
THEN
TD TRACE
UNBREAK
UNTRACE
UPDATE
VAL
VARIABLE
VCHAR

POKE and PEEK allow the user some access to assembly language programming. These are not found in FI BASIC. CALL MCHL(address) allows you to execute assembly language (machine language) code starting at the specified address.

CALL KEY(KEYUNIT,K,S) on the 99/2 only recognizes key units 0 and 1, both of which are interpreted the same. In TI BAGIC you can have your choice of 5 key units, each of which returns different sets of values for K when the same specific keys are pressed.

There is no GPL or GROM in the 99/2. The 99/2 DASIC interpreter is written in assembly, unlike the GPL DASIC interpreter of the /4A. This means that 99/2 BASIC is interpreted only once, not twice as is the case with TI DASIC on the 99/4A. This single interpretation of 99/2 BASIC, plus the faster speed of the 99/2's 9995 CPU (10.7 MHz) compared to the 9900 processor of the 99/4A (about 3.3 MHz) means that 79/2 BASIC is really fast!

100 FOR N=1 TO 10000 110 PRINT N 120 NEXT N

The above program takes 30 minutes 20 seconds in TI DASIC with the 99/4A. It takes only 3 minutes and 3 seconds in 99/2 BASIC. In this case, 99/2 BASIC is a blistering TEN TIMES FASTER than TI BASIC. For comparison, the same BASIC program took exactly 6 minutes to run to completion on my Tandy 1000HX, an XT clone running at 7.16 MHz. Barry Traver told me, "I saw a demo of the 99/2 at the original west coast TI show years ago. A guy typed in a benchmark program on the 99/4A and then set it going with RUN. He then walked over to the 99/2, typed in the same program and RUN. The 99/2 caught up with and passed the 99/4A. The 99/2's program terminated first."

#### DOCUMENTATION:

The 99/2 was supposed to be packaged with 4 instruction books, each book slightly more advanced than the previous. A demonstration cassette tape with three programs, "Common Blast", "Addition Tutor", and "Loans" also was supposed to be packaged with the 79/2. Home of my sources know anything about the cassette tape except the titles.

I have copies of what is claimed in the header on each page to be the "FIMAL BRAFT" of the four 77/2 heads, apparently printed by a main frame computer printer. The top of each page bears one of these cryptic notations: 17181, 02667, 03187, 03179, and 03269. The books are titled "Getting Started", "BASIC for Beginners", "Advanced BASIC Programming", and "BASIC Reference Guide", also known as books 1,2,3 and 4. Book 4, the Reference Guide, is almost identical (often word for word and sample program by sample program identical) to the 99/46's User's Reference Guide. Books 2 and 3 resemble, but are not identical to, the 99/40's Buginner's Basic book (the blue book). Some nice application software listings are found at the end of book 3.

It is obvious to me that these four books are not really FINAL drafts. There are lots of errors. Book one status that the zero is slashed so you can distinguish it from the letter O. It isn't. The docs say that FCTN/= (QUIT) resets the computer to the title screen. It doesn't. One of the sample programs in book 4 uses CS1 and CS2, but the 99/2 only supports CSI as a mass storage device. The MEMORY FULL error message is mentioned several times in the books when im fact the 99/2 generates the message OUT OF MEMORY. There are other error messages that differ between the /4A and the /2. The documentation lists these messages as they would appear on a /4A. Some of the /4A's error messages are lacking on the /2, but still mentioned in the /2 books. For example, the ERROR DETECTED IN DATA cassette error message is mentioned several times in the books when no such message is, generated by the 99/2. Important aspects of the Newborn interface, such as the CAtalog feature, are not mentioned. I have discovered some error messages that are not mentiomed in any of the 99/2 books. For example, under two different sets of circumstances I have run into the message INTERNAL EMBER, PRESS ENTER. Pressing (enter) then resets the computer to the title screen. Book 3 (Advanced BASIC) contains a heavily commented listing of a BASIC program designed to produce a meat moving graphic display with the 99/2's built in graphic characters (ASCII 0-31). The only problem is that the listed program is too large to fit into the 99/2's limited memory. Defore you are finished typing in the program as listed, the computer informs you that it is DUT OF MEMORY.

I have been told by knowledgeable sources that there are at least three kinds of 99/2's known to exist; wire wrap prototypes shown at the Jan 83 CES, production versions with 3 ROM chips such as that photographed in the BYTE article, and 79/2's with only one larger capacity ROM. Obviously my 79/2 is not one of the wire wrap jobs. I am afraid to pup the cover off my 99/2 (the cover is held in place by spring loaded metal clips, not screws) to see how many ROM's it has. The documentation errors mentioned above may be due to my 79/2 being different than those available to the documentation authors. Another possibility is that my 79/2 "FIRM. DRAFT" documentation was written by individuals who only had printed specifications and did not have hands on access to an actual 79/2.

CONCLUDING REMARKS:

With the promised 32K memory expansion attached, the 99/2 mound probably he easy to program in assembly for powerful applications. There is only one kind of programmable memory, CPU RAM. There is no GROM/GRAM or VDP RAM to slow things down and confuse the assembly programmer. BASIC programming on the 99/2 is also easy, and 99/2 BASIC's speed is probably amparalleled among 99/4A related products. Even TI Extended Basic does not begin to approach the speed of the 99/2's BASIC.

Unfortunately, the 32K 99/2 memory expansion device is met known to exist, and the slightly over 4K of RAM is very restrictive. The usual method of estimating free memory with TI BASIC on the 99/4A is to run this program:

1 A=A+1

2 GOSUB A

When the OUT OF MEMORY message appears, type PRINT ARE to get the number of bytes of free memory. With TI BASIC the 77/4A without memory expansion and without any cartridge in the cartridge port gives an answer of 14536 to the PRINT ARE command. My 99/2 shows only 4302.

What little memory the 99/2 does have has to be used to control the video display as well as to store and execute maste programs. A BASIC program starting at line number 100, incrementing line numbers by 10, and ending with line number 1000 is about all that can be squeezed into the 99/2's RAM.

Speed, and the potential of expanded memory give the 99/2 lots of potential as a serious computer capable of useful applications. However, without memory expansion, the 99/2 is little more than what TI envisioned for the product, a learning tool.

\*\*DONE\*\*

The two page advertiseent on the following page is found just inside the front cover of the May 1983 issue of POPULAR SCIENCE. TI's spokesman Bill Cosby is shown holding a 99/2 under his chin.



# TI's new Basic Computer. The one to start with and get smart with.

Under \$100. Meet the lowest priced, 16-bit comouter available. A major breakthrough in computer rechnology.

Get ready. Get set. Begin. Only the TI-99/2 Basic Computer with builtn TI BASIC has a plug-in colid state cartridge that eaches you, step by step, he basics of programming. And our operator's manual If you're wondering if this s clear and to the point. You learn quickly, because it doesn't assume you're an engineer.

A real computer. Powerful 4K memory. Fullzize, 48-key keyboard. Quality, licker-free black and white display. With a thoice of popular software pplications.

Easily expandable. When you're ready for more, plug in more memory (an additional 32K). Add a Wafertage "Drive or Cassette Program Recorder. Modem for telecommunications Printer. And a plug-in cartridge that teaches you how to pro-

Money well-spent.

gram in TI BASIC.

is the right computer for you, touch the keyboard, look at the display, watch this powerful computer at work. Even compare it to the Timex Sindair 1000". the other computer under \$100. Convince yourself.

Start with a visit or call. Ask your local TI retailer to show you the best basic computer buy for your dollar Or call us at (800) 858-4565.

Creating useful products and services for you.



#### THE TI 99/7 COMPUTER

This device was apparently never built, but its 1901 design specifications were probably the inspiration for the 99/8. The 99/7 is discussed in a June 16, 1980 BUSINESS NEEK article

We have a copy of the following documentation from TI: "TI 99/7 COMPUTER HARDWARE THEORY OF OPERATION, Corporate Engineering Center, Dallas, By Richard Tarrant", dated 13 July 81. Based on this document, it appears the 99/7 was going to be based on the 9900 CPU, fully software compatible with the 99/4 (without the A), and include ALL OF THE FRELEWING WITHIN THE SAME CASE (not as separate peripherals)—64K of short wait state memory, speech, a "pascal grom", an 80 column thermal printer, an acoustic modem, and a 5.25 inch disk drive controller.

The following is quoted from this document:

"The TI 99/7 computer incorporates the 99/4 home computer mainframe architecture with the 99/4 peripheral controllers plus expanded memory into the case plastics of the Digital Systems 765 terminal for compactness and portability. The TI 79/7 will retain software compatibility with the 99/4 unite providing additional faster access memory as well as a larger beytheard and 80 column printer

"A comparison of features between a 99/4 with its outernal peripherals and the 99/7 is as follows:[(memory in bytes/microseconds access per (word) or (byte)]

99/4 (## means external peripheral required)

Keyboard......40 key polled

Video Display........40 char x 24 line color NTSC Cassette Interface.....1 port 650 baud 2 unidirect.

Jeystick Interface.....Dual scan/1 port

External I/O.....40 pin full cpu with B bit data

Expansion CPU RAM.....32K/2.0(W) ##

Speech.....TMS0285+ 2-TMS0350##

Printer.....32 col/30 cps##

Disk Controller......3-single density single side##

RS232......Dual 9600 baud##

Medea.....300 baud acoustic originate##

99/7

Keyboard...........50 key + 4 reck switch, pell

or interrupt.

Video Display......40 char x 24 line color HTSC Cassette Interface.....1 port 650 band 2 bidirect.

Joystick Interface.....Dual scan/1 port

External I/O......9 pin with 8 bit latched data

Expansion CPU RAM.....64K/0.7(W)

Speech......THS0285+ 2-TMS0350

Printer.....80 col/30 cps

Disk Controller.....3-single density double side

RS232.....Dual 9600 baud Modes......300 baud acoustic

originate/sessor

\*\*DONE \*\*

WE TI 97/8 COMPUTER notes by Charles Sood Lina Ohio User Group

We have all heard of the legendary never released 79/8 computer. Probably the best published description of this computer to date are found in the December 1984 issue of MICHOPENDIUM, where it is stated that the suggested retail price of the 99/8 would have been about \$600. Photos of this computer in the article show the nameplate "Texas Instruments Computer 99/8" just above the left side of the keyboard. Several working 99/8s are known by me to be in private hands, and I have seen some on display, but not operating, at a comple of TI shows. I have, however, until recently never really been clear what this machine could do. Certainly the best way to judge the capabilities of a computer is to mae it. I have never had the opportunity play with a 99/8, but I have done the next best thing.

I have obtained a copy of the "Final Draft" of the 99/8 user documentation "GETTING STARTED" (dated 8/30/83 with the code number "0811P") and "PROGRAMMER'S GUIDE" (the p-System part dated 8/22/83 and the rest dated 9/15/83; with the code numbers "1511P, formally 2183L", "2117L", and "1240P"). These books give an excellent description of the capabilities of the 99/8 computer and interesting insights about the very very rare Hexbus Disk Drive/Controller peripheral. I know of no working examples of the Hexbus Disk Drive/Controller peripheral. Sources tell me that it was made for II in Gormany by Mechatronics. In my opinion, after reading them books, the greatest shortcomings of the 99/8 are the lack of an 80 column display and the inability to generate unlimited speech using its internal speech capabilities coupled with the TEII module.

In order to inform the TI community about "what might have been if TI had unly...." I am publishing the following selected material all quoted from my 99/8 user documentation. Text below in brackets () are my sum words interjected among those courses.

"Features of the Texas Instruments Computer 99/8:

"KEYBOARD--The 99/8 computer has a full size, easy to was

"DISPLAY--The 99/8 displays program lines in a 24 row, 40 column format (text mode) using a handsome character set with true lower case letters.

"SOUND AND SPEECH--The 99/8 can generate sound from 110 cycles per second to beyond the highest range of human hearing. Built-in SOLID STATE SPEECH synthesis reproduces buses speech electronically, accurately, and realistically.

"MEMORY—The 99/8 has 220 kilobytes of built in ROM. In addition, the computer has 80 kilobytes of RAM (64K of CPW RAM), which eliminates the need for memory expansion for most applications. If you need more memory, the 99/8 can access up to 15 megabytes (approximately 15 million bytes) of total random access memory. (CG note: One 77/8 owner I talked to has working TI 128K and 512K PE box RAM cards specifically designed for use with the 99/8.) TI Extended BASIc II, a versatile and powerful version of the 300GIC programming language, is resident in the computer camenie.

"ADDITIONAL PROGRAMMING LANGUAGES--The built in p-System, the Universal Operating System (both these names are trademarks of SofTech Microsystems, Inc.), allows other programming languages, such as UCSD Pascal and TI PILOT (both on diskette, sold separately), to be used.

"JOYSTICKS--An outlet is provided for Wired Resote Controllers.

"EXPANDABILITY—The built in HEXBUS Interface enables you to use the low cost HEX-BUS peripherals; outlets for connecting a cassette recorder and future peripheral devices are also included. (CG note: Apparently the Hexbus port was to be the main expansion port, at least initially. There is a 50 pin expansion for "future devices" on the side of the computer. This 50 pin 99/8 port is NOT the same as the side expansion port on 99/4A consoles. Some current 99/8 numers have a II cable that connects between this 50 pin port and the "fire hose" cable of the PE box. This is the only existing device I know of for the 50 pin expansion port.}

"TECHNOLOGY--The Computer 99/8 uses the 16 bit TMS9995 microprocessor for fast program execution. (This is the same 10+ MHz CPU used today in the Geneva. The 9995 is also found in TIs 99/2 computer.)

"When you turn the computer on, the master title screen appears.

(TI BUG)
TEXAS INSTRUMENTS
COMPUTER

111111111111111

READY-PRESS ANY KEY TO BEGIN

 "Prove any key to proceed to the next screen. This is the main selection list or the main "manu".

#### TEXAS INSTRUMENTS

#### COMPUTER

PRESS

[A] FOR TI EXTENDED BASIC II

[B] FOR P-SYSTEM

**IC1 FOR SET SPEED** 

If you have a Solid State Cartridge plugged into the slot at the top of the computer console, the name of that program usually appears as selection D on this screen.

"TI Extended BASIC II—This option enables you to access the programming language built into the Computer 99/8. When you select TI Extended BASIC II, the computer automatically executes at full speed.

"SET SPEED--This option enables you to change the **speed** at which the computer "runs" programs contained in cartain preprogrammed software packages. When the computer is turned on, it is automatically set to run at 99/4A speed. If this aption is selected, this screen will appear:

SET SPEED

(A) SLOW

**IB3 99/4A SPEED FOR GAMES** 

[C] FULL SPEED

"To proceed to the next section, "A tour of the keyboard," select TI Extended BASIC II by pressing the A key.

TI EXTENDED BASIC II ready 62720 Bytes free

"[Illustration of stacked MEX-BUS peripherals]

"DISK DRIVE/CONTROLLER 5102-A Hexbus peripheral that uses 5 1/4 inch diskettes, finds files quickly, and allows either sequential or random file access. With its double-sided, double density capabilities, the Disk Drive/Controller can store up to 320K of information on one diskette.

"Disk Drive 5202—Up to three additional disk drives can be used with the Disk Drive/Controller (CG note: up to a total of 4 DSDD drives. Since the Hexbus has only 4 data lines data transfer is in nibbles rather than bytes. I suspect that the data transfer rate between Hexbus Disk Drive/Controller and the 99/8 or other computer would be significantly SLOWER than that obtained with the TI disk controller and a 99/4A.}

\*FEATURES OF TI EXTENDED BASIC II:

"BATA TYPES--TI Extended BASIC II allows both integer and real data types for numeric variables. Because the integer data type uses less storage space and requires less processing time, using this data type when applicable (for example, with FOR NEXT loops) facilitates faster program emocution. To be stored as an integer a number must be a whole number within the range -32768 to 32767.

"SIX DIFFERENT GRAPMIC MODES—These are accessed using EALL GRAPHICS(x). A wide variety of display techniques and applications, including dividing graphics and text, are available with the six graphics modes:

\*PATTERN MODE--The default mode when using TI Extended BMSIC II is selected. Characters are in 24 rows and 32 columns with each character defined by an 8x8 pixel configuration.

"TEXT MODE--Characters are in 24 rows and 40 columns, with each character defined by an 8x6 pixel configuration.

"SPLIT SCREEN MODE: TEXT HIGH--The screen is split into two portions: the text portion (the top one third of the screen) and a graphics portion. Text is 8 rows by 32 columns; the graphics portion is 128 pixels by 256 pixels.

\*SPLIT SCREEN NODE: TEXT LOW-- The same as Split Screen:
Text High except that the text is at the bottom one third of
the screen, with graphics at the top.

\*NIGH RESOLUTION NODE--A full screen graphics mode with 192x256 pixels.

"MULTICOLOR MODE--Rather than in characters or pixels, the screen displays in blocks, with 48 x 64 blocks.

"Screen Margins--CALL MARGINS enables you to redefine the screen margins and thus define a text "window" on the screen.

\*FREESPACE(0) returns the amount of unused memory space.

"WALHEX returns the decimal (base 10) value of a hexadecimal (base 16) number.

"TERNCHAR returns the key code of the key combination (such as ENTER, PROC'D, BACK, BEGIN) that terminated the last INPUT, LINPUT, or ACCEPT statement.

"MSSENBLY LANGUAGE SUPPORT—TMS9795 (of which TM97900 code is a subset) assembly language subprograms may be loaded and run. The subprograms INIT, LOAD, LINK, PEEK, PEEKV, and PMKEV are used to access assembly language subprograms. Relocatable assembly language programs created for TI MASIC will execute correctly, although NUMREF may return an integer value instead of a floating point value. In addition, a string reference error may occur with STRREF because STRREF (in TI BASIC) only allows strings up to 255 characters. All PMKEs and PEEKs as used in previous BASICs fail in TI Extended BASIC II. The INIT subprogram with no parameter allocates 8K bytes of memory for assembly language subprograms. If a parameter is specified, more or less (up to 24366 bytes) may be allocated. Examples:

CALL INIT allocates 8K bytes of memory space.

CALL INIT(200) allocates 200 bytes of memory space.

CALL INIT(0) releases all memory previously allocated.

"DISPLAY VARIABLE 80 FORMAT--Files in DV80 format, created by the LIST command or a text editing or mord processing program, may be loaded with the OLD command. (C6 Note: Does this mean that you can just display 9400 on screen, or can you SLD and NUM BAGIC code created with a word processor? I don't know.)

"KEY CHIRP--TI Extended BASIC II enables you to turn on an audible "chirp" that sounds whenever a key is pressed. Enter the following command to turn on the key chirp: CALL LOAD (VALHEX("84BD"),1). Enter the following command to turn the key chirp off: CALL LOAD(VALHEX("84BD"),0).

"STRING LENGTH--BASIC II permits strings up to 4070 characters in length. II BASIC and II Extended BASIC parmit stings of up to (only) 255 characters in length.

"RECORD LENGTH--TI Extended BASIC II allows diskette data files created on the Drive/Controller to have VARIABLE records with lengths up to 4090 bytes. TI BASIC and TI Extended BASIC allow VARIABLE records to be only 254 bytes long. Diskette data files with FIXED length records are limited to 255 bytes (same as TI BASIC). TI Extended BASIC II allows cassette data files to have FIXED length records up to 4032 bytes. TI BASIC and TI Extended BASIC limit casestte data files to 192 bytes or less.

"SOFTWARE CARTRIDGES--CALL to routines contained in a plug in cartridge are not accessible in TI Extended BASIC II. Thus, programs that use the Personal Record Keeping cartridge will not execute properly. Text to Speech cannot be accessed from TI Extended BASIC II with the Terminal Emulator II cartridge. Otherwise the Terminal Emulator II cartridge functions normally.

"RESERVED MORDS--The following reserved words are additions to the TI Extended BASIC reserved word list: ALPMA, FREESPACE, INTEGER, LALPHA, REAL, TERMCHAR, and VALHEX.

"CALL DRAW, CALL DRAWTO, CALL FILL, and CALL BCOLOR--These are graphics subprograms which enable you to plot graphics and add color to them on the screen (in graphics mode and in the graphics portion of a split screen). CALL DRAW and CALL DRAWTO draw or erase lines between specified pixels, thus making elaborate figures or drawings possible. (The screen is comprised of a grid of 256x192 individual pixels.) CALL FILL colors the area surrounding a specified pixel. CALL DCOLOR sots the graphics colors that are used by CALL DRAW, CALL DRAWTO, CALL FILL, CALL HCMAR, and CALL VCHAR.

"Some HexBus peripherals can be accessed by using the general format for file specification described earlier. (This is the method we are all familiar with in BASIC file access with the 79/4A.) Hexbus peripherals that may be accessed with this alternate method of addressing are the Bisk Drive/Controller, the RS232, and the HexBus Hodem.

Alternate Device name! Hexbus Number

100. DISKNAME
101
102
103
104
70

"MEXIMS SMCCHMANDS--The CA (catalog) command cannot be used with the Meximus Disk Drive/Controller peripheral. (CG note: CA is supposed to generate a catalog of programs stored by the 99/8 on a wafertape. On my 99/2 computer CA does generate a catalog of files on a Wafertape or Mechatronic Omichdisk, and is supposed to work similarly with TI's never released Hexbus Interface for the 99/4A.)

"The available characters (ASCII 0-255) and character sets (32 character sets numbered 0-31) in Pattern Mode are listed.

"The p-System:

"The P-code interpreter, which is built into your computer, enables you to execute existing p-System programs. To develop your own programs you must have a TI disk system (sold separately). Also necessary are some or all of the following TI products (sold separately):

- 1. p-System Editor/Filer/Utilities.
- 2. UCSD Pascal Compiler.
- p-System Assembler/Linker.

"Mute: These products are designed specifically for wee with the 99/8. Software designed for use with the TI 99/46 Name Computer may not work when used with the 99/8.

"With the p-System you can execute high-level language programs such as UCSD Pascal and TI PILOT. UCSD Pascal is compiled and TI PILOT is interpreted to an intermediate language called "P-code" or "pseudo-code." The p-system interprets the P-code and instructs the computer to execute the appropriate machine language instructions.

"The built in MINI FILER program in the Operating System (unit #14) file named SYSTEM.TI.FILER provides many of the file management capabilities of the p-System Filer program. CMIMI FILER is in ROM. The p-System Filer program is extra cost disk software.? Use the "5"o command from the system command level to access the MINI FILER.

"When you enter the MINI FILER, a menu of "Special p-Bystem Commands" is displayed. To select a menu option, press the letter that precedes it.

- A. Run a program
- B. Coov a disk
- C. Copy a file
- D. Delete a file.
- E. List files on a disk.
- F. Format a new disk.
- 6. Clear a disk directory.
- N. Combine free disk space.
- I. Change name.
- 4. Return to standard p-System menu.
- 2. Kelo
- \$. Set single disk system. (Toggles to "multiple" disk)

"APPENDIX O: ABREXALY LANGUAGE SUPPORT ROUTINES

unable you to access special capabilities of the computer through TM97700 assembly language. With these etilities, you can change the values of the VMP chip, access the BOR for poripheral devices, scan the keyboard, kink a program to GPL rantines, and link to the Editor/Assembler loader. Remember that these can only be used in TI assembly language programs.

"The following list gives each of the utilities predefined in the REF/DEF table: VSBN VMBN VSBR VMBR KSEAN GPLINK XMLINK DSRLNK LDADER.

"The TI Computer 99/8 has more utilities available through the Editor/Assembler than did the II 99/4A Nome Computer. As a result, the IMLINK tables have been changed, so old assembly language programs may need to be updated

##DONESS

#### LETTER TO THE EDITOR

25 August 1991

Dear Charlie:

Yes, at least for the time being we have music for the Tandy Computer. Enclosed is a sampler disk and a catalog of what we have for the Tandy computers. It's a subset of what we offer on the TI. We have had totally miserable sales of these Tandy products, and plan to discontinue the product line altogether. [Editor's comment: Maybe II users aren't so cheap after all, at least compared to Tandy users!]

Me're working on music to support the TI MIDI INTERFACE. Also in work is a game program "SCUDBUSTERS", on a Desert Storm theme. Both scheduled for introduction at Chicago. See you there.

Bruce Harrison Harrison Software 5705 40th Place Hyattsville MD 20781 301-277-3467

167 ##DONE##

#### \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* BITS. BYTES & PIXELS Published by Lima NH 99/4A User Group P.O. Box 647 ADDRESS-Venedocia OH 45894 Published monthly except July and August **GROUP OFFICERS** President-Susan Cumminus 419-738-3770 \* Vice Pres-Peter Harklau 419-234-8392 \* Treasurer-Leonard Cummings\* 419-738-3770 \* \* Newsletter & librarian 419-667-3131 # \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

<sup>&</sup>quot;The TI Computer 99/8 provides several utilities that

25 AUGUST 1991

Enclosed is a disk with my latest utility for XB programmers on the TI. IBBAP editor's note: This software is on disk 616B of the Lima User Group's software library.] It's a subroutine for assigning Random numbers to array variables. The XB demo program illustrates it's workings by assigning random numbers to an array of dimension 500 in about one second. Also included in the subroutine is the entry point called SEED, which acts like a key loop, and assigns the key struck into a variable in XB. This is needed IF the program was entered through a LOAD program, or if started from a Randisk menu. In both those cases, the Random number seed at >83CO will not have been properly set, and the subroutine SEED will fix that.

The call link to RARRAY requires four parameters. First is the lowest number in the range, second is the highest number in the range, third is the number of members of the array to assign random numbers to, and fourth is the variable name. all numbers assigned will be integers within the requested range.

Incidentally, this also works for negative numbers. try putting in -128 and 127 for the first two parameters in the call link (line 160 of the dean program), and see what happens. Yes, it makes 500 numbers between those limits. The ultimate limits are -32768 and +32767.

Great care was used in this subroutine to insure that the SEED part will assign a random number seed with the full range of possible values (65536 states). This means, in effect, that there are that many different sequences of pseudo-random numbers that can be generated. The source code file (RAMPONS) has been annotated so that those familiar with assembly will be able to understand its operations.

In application, one submerges the object file RANDON with ALSAVE, then uses a call to SEED after placing some vessage like NEED INSTRUCTIONS? on the screen. After the link to SEED, K will contain a random number of full 16 bits

```
FILE Type 8110001 $ IDAMRAYS - POR USE WITH ALSAVE UNDER EXTENDED BASE

FOR AN ARRAY VARIABLE

TRANSPORTED THE ASSISTED BOO RANDOM NUMBERS IN ASDUT 1 SECOND

FIRST SPERATION IS TO CALL LINK("SEED",K)

HIS THIS IS DONE, RANDOM NUMBER FOODLED

FROM XB, CALL LINK("RARAY", LOWNUM, HIS HUNDUM, DIRENSION,X())

WHERE LOWNUM IS THE LOWEST VALUE DESIRED IN THE RANGE

MISHAND IS HIS MESTED IN ARRAY

IN HIS HOUND IS THE NUMBER OF MEMBERS IN ARRAY

IN CODE BY SRUCE HARRISON

RELEASED TO FUBLIC DOMAIN

I SELDE SEED TO FUBLIC DOMAIN

I SAUB 1991

NUMASS EQU >2008

NUMBER CABBERT VECTOR

KEYADR EQU >2016

KEYADR EQU >3575

KEY-UNIT LOCATION

KEYADR EQU >3575

KEY VALUE BYTE
                                                                                                                                                                                        # IBARRAYE - POR USE WITH ALSAVE UNGER EXTENSED BASIC
                                                                                                                                                                                                NUMERIC ASSISTANCES VECTOR

KEYSDARD SCAN VECTOR

KEY-UNIT LOCATION

KEY VALUE BYTE

PLOATING POINT ACCUMULATOR

CONVERT INTESER TO PLOATING POINT

CONVERT FLOATING POINT TO INTEGER

NUMERIC REFERENCE VECTOR

GPL STATUS BYTE

DEFINE ENTRY POINTS
                                                                                                                >20
>1288
0023
0024
0028
0034
0027
0030
0037
                                                                                                                >200C
>837C
RARRAY, BEED
                                    SEED
                                                                                  LWPI MB
MDV $28378,R10
ANDI RIO,>0001
CLR BKEYADR
MDVS BYESDY,B>83C1
CLR BETATUS
BLUP EKSCAN
                                                                                                                                                                                                    LOAD DUR WORKSPACE
TAKE THE VDP INTERRUPT TIMER INTO RIC
MASK OFF ALL BUT THE LOWEST BIT
                                                                                                                         10,>0001 MASK OFF ALL BUT THE LOMEST BIT
KEYADR
CLEAR KEY-UNIT
90307,90301 TAKE THE SCREEN TIMEOUT'S LOW BYTE INTO SEED + 1
GTATUS
CLEAR SELERENT TIMEOUT'S LOW BYTE INTO SEED + 1
GTATUS
CLEAR SPL STATUS
KSCAN SCAN KEYBOARD
ALLOW INTERRUPTS
ANYKEY,98TATUS HAS A KEY BEEN STRUCK?
EYIN
15 NOT, 50 SACK
100,50 STO PUT NOT, 50 SACK
100,50 STO PUT NOT SU LOW BIT OF RIO IS LOW BIT TAKEN ABOVE
10,50 STO PUT BYTE FROM VDP INTERRUPT INTO HISH BYTE OF SE
0 CLEAR RO FOR NUMBER ASSISM
FIRST PARAMETER TO PASS
KEYADR,9FAC FLOSE KEY'S ASCII VALUE AT FAC
INLANK
USE INL LINKAGE
IP
TO CONVERT INTEGER TO PURSTURE
STEO LOAD SPL WORKSPACE
STATUS CLEAR SPL STATUS BYTE
                             KEYIN ACUR
CLR
BLWP
LINI
 LIMI 2
LIMI 0
CS SANYKEY, STATI
JME KEVIN, TIO
HOV RIO, SSCO, TIO
HOV RIO, SSCO
CLR RO
LI RI:
HOV SKEVADR, SFAC
ELMP STMLLNK
DATA CIP
SUMP SEED
LWF SSEO
LUR SSEO
                                                                                                                                                                                                    ASSION RUMBER IN FARM
LOAD SPL WORKSPACE
CLEAR SPL STATUS BYTE
RETURN TO SPL INTERPRI
                                                                                                                                                                                               RETURN TO GPL INTERPRETER

LOAD DUR OWN MORKSPACE
FIRST FORANCETER NOT AN ARRAY
SET RI FOR FIRST PARAMETER
GET THE LOW-RD LIMIT NUMBER
USE XML LINK
TO CONVERT NUMBER TO INTEGER
RIZ HAS LOW NUMBER
SET RI FOR SECOND PARAMETER
SET THE FOR SECOND PARAMETER
SET THE MISH-END LIMIT MUMBER
USE XML LINK
CONVERT TO INTEGER
RIZ HAS HIGH MUMBER
ADD DWE TO HIGH MUMBER
SUBTRACT THE LOWER LIMIT
BET RI FOR THIRD PARAMETER
BET RI FOR THIRD PARAMETER
HOVE NUMBER OF MEMBERS INTO RO
FOINT RI AT FOURTH PARAMETER (ARRAY VARIABLE)
PUT A BIG MUMBER IN R4
MULTIPLY BY THE RANDOM MUMBER SEED
ADD A BIG MUMBER TO THE RESULT
                                                                             RARRAY
                                                                                # R12,R13
INC R1
INC R1
BLMP BXMLLNK
DATA CFI
MOV BFAC,R0
INC R1
L1 R4,28445
HPY #>8350,R
                              MEWNUM LI
                                                                                  RPY
RDV
                                                                                                                                                                                                 MULTIFLY BY THE MARDON NUMBER SEED LOCATION
ADD A BIG NUMBER TO THE RESULT AT
PLACE THAT RESULT AT RANDON NUMBER SEED LOCATION
CLEAR REGISTER 4 DO R4-R5 CONTAIN A NUMBER
DIVIDE BY THE NUMBER IN R13, (HISH-LON+1)
ADD THE LON LIMIT TO REMAINDER
HOVE THE NUMBER TO FAC
 0074
0075
0076
0076
0077
0000
0001
0002
0003
0004
0004
0006
0006
0006
                                                                                                               R6.31417
R5.0>03Co
                                                                                                             R4
R13,R4
R12,R5
R5,BFAC
BXMLLNK
                                                                                   ELR
Div
                                                                                  A
MOV
BLWP
                                                                                                                                                                                                 MOVE THE NUMBER TO FAC
USE XML LINK
CONVERT HUMBER TO FLOATING POINT FORMAT
ASSISM OWNE ARRAY REFER
DECREMENT ARRAY COUNT
IF NOT ZERO, ASSISM ANOTHER
LOAD SPL MORKSPACE
CLEAR SPL ETATUS BYTE
RETYMN TO SPL INTERPRETER
SWR SWM MORKSPACE
                                                                                DATA CIF
BLWP MAURABE
DEC RO
JNE NEWNUM
LMPI XEZEO
CLR CETATME
3 2006A
```

length. After SEED, basic's RANDOMIZE
will work as it's supposed to, and our
call link for array variables (RARRAY)
will also work properly.
Bruce Harrison
5705 40th Place
Hyattsville MD 20781

\*\*DONE \*\*