

99/4 TI BASIC no module 14536
XB module 13928

Texas Instruments' 99/2 Basic Computer

A look at the design process from concept to prototype

Harry Littlejohn and Mark Jander
Texas Instruments Inc.
Consumer Products Group
POB 10508, MS 5849
Lubbock, TX 79048

Designing a new low-cost computer system is a delicate balancing act. Your system must satisfy the needs of your market, respond to the challenges presented by your competitors, and still be profitable to build and market. In designing the new Texas Instruments 99/2 Basic Computer, we managed to achieve all of the above requirements in only four and a half months, which may be record time for a new system.

The TI-99/2 was developed to compete directly with such low-cost computers as the Sinclair ZX81 and the Timex/Sinclair 1000. The tremendous demand for computers that sell for less than \$100 demonstrated a growing need for a low-cost computer-literacy product for both children and adults. Our primary goal as designers of the TI-99/2 was to build a home computer that would enable people of all ages to take part in the computer revolution around them. It would also be used as a learning computer for students from grade school through college.

To satisfy consumers' needs, a low-cost computer must function beyond the computer-literacy stage; only then can it enable novices to improve their newly developed skills, through activities such as personal record keeping, education aids, and even mind-challenging games. Furthermore, the computer must be capable of growing with individuals who want to develop more advanced skills



Photo 1: The Texas Instruments Basic Computer 99/2 is designed to compete directly with low-priced computers such as the Timex/Sinclair 1000. BASIC programs written on the 99/2 can run on the TI-99/4A, but the reverse is not true.

or applications.

Texas Instruments set what we considered to be very aggressive goals in the design and manufacture of the TI-99/2. In this article we'll discuss those goals and how we met and sometimes, in our opinion, surpassed them in the course of the four and a half months between the initial concept and the completion of working prototypes for the January 1983 Consumer Electronics Show (see photo 1).

Design Requirements

We believed a major portion of our market would be first-time computer buyers looking for a minimal investment in the world of computers. We kept the price low, hoping that those who decided computer programming was not for them would not regret the expense. Our basic assumption about

these potential buyers was that they would base their purchase decision on price, not features. Computing power, expandability, and the availability of peripherals would be a relative measure for choosing from among several computers in the same price range.

We felt strongly that the first-time buyer must not be intimidated by the product. The console, we decided, must be simple, and the keyboard must be no more complicated than a typewriter keyboard. A cluttered keyboard, with multiple-function keys—and, therefore, multiple legends—would generate an immediate "I'll never learn to use that" attitude. In addition, we thought the computer should be sufficiently simple to operate and self-contained enough to ensure ease of use.

Once the beginner starts to develop computer skills, the computer needs peripherals that make new applications accessible. An economical program and data-storage unit was necessary, as were preprogrammed application software packages to increase the computer's utility. We decided that the system would also need a low-cost printer for listing and as a way of keeping track of program output. In addition, the system needed to handle the memory expansion that becomes necessary as programming expertise increases.

For computer owners who progress to more advanced applications,

ype

was that they
 purchase decisions
 . Computing
 and the avail-
 would be a
 choosing from
 rs in the same

the first-time
 idated by the
 we decided
 the keyboard
 icated than a
 A cluttered
 multiple-func-
 e, multiple
 te an immed
 use that" at
 e thought the
 fficiently sim-
 self-contained
 of use.

rts to develop
 mputer need
 new applica-
 nomical pro-
 unit was nec-
 programmed ap-
 packages to in-
 utility. We
 n would also
 r for listing
 g track of pro-
 n, the system
 memory expan-
 ssary as pro-
 reas
 who progres
 applications.

range of more sophisticated periph-
 erals is necessary. We saw a need for
 communications or networking de-
 vices; high-density, fast-access mass
 storage; and printer/plotter-type
 peripherals to provide a very power-
 ful but low-cost computer system (see
 photo 2). Growth beyond these levels
 of sophistication would probably en-
 tail buying a computer with greater
 capabilities.

To ease the upward migration to
 more sophisticated computers, we
 decided to make our computer part of
 an integrated product line based on
 the 99/4A. Integrating the computer
 with other TI products would enable
 us to provide compatible software
 and peripherals for other members of
 the TI Home Computer Family. Pro-
 gramming techniques and computer
 programs developed on a low-cost
 machine could be used on the more
 advanced machine without being
 rewritten. Peripherals for the new
 machine would then be compatible
 with the more expensive console, and
 none of the equipment and program-
 ming exchanges would become
 obsolete.

Having identified these require-
 ments, we selected the following
 characteristics for the TI-99/2:

- 1) a computer system for less than \$100 that consists of a console, an external AC adapter for power, and a minimum of 2K bytes of RAM (random-access read/write

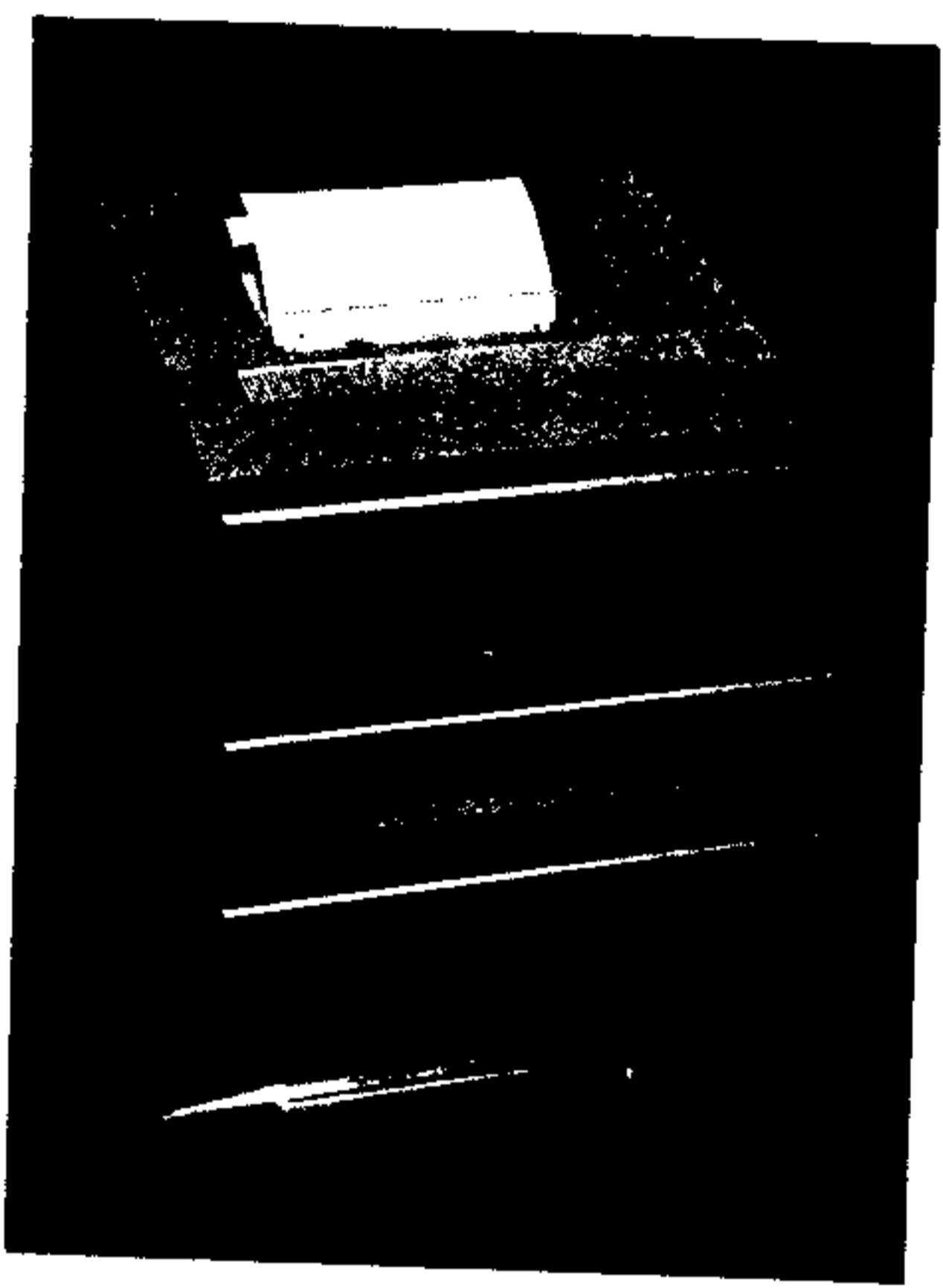


Photo 2: Peripherals for the 99/2. The line of low-cost peripherals includes, from top to bottom: the HX-1000 printer/plotter, the HX-2000 Wafertape Digital Drive tape unit, and the HX-3000 RS-232C interface.

- memory) and 16K bytes of ROM (read-only memory)
- 2) a built-in, two-channel black-and-white RF modulator (channels 3/4) for connecting the system to a standard TV (use of black and white was necessary to achieve cost goals)
- 3) a built-in interface for a standard audio-cassette recorder to provide economical mass-storage capability. Only audio in/out would be supported by the system. Cost would not allow support of motor control
- 4) a built-in Hex-bus interface to provide compatibility with the line of low-cost peripherals
- 5) an operating system that would be a subset of TI-BASIC as used on the TI-99/4A Home Computer console. Thus the software written

on the 99/2 would be upward com-
 patible with the 99/4A. For cost
 reasons, the 99/2 would not use
 the Graphics Read-Only Memory
 (GROM) programming language
 of the 99/4A and, therefore,
 99/4A command cartridges would
 not be compatible with the 99/2.
 Software compatibility would be
 provided by BASIC programs
 stored on cassette

- 6) no sound, color, or joystick capability would be supported, again because of cost constraints
- 7) the full system bus structure would be available at an expansion port on the rear of the console. This would facilitate memory-expanding Solid State Software cartridges (not 4A compatible) or any other future system expansion
- 8) a nonintimidating, 48-key typewriter-like keyboard with only two legends per key top. Layout and functionality of the keyboard would match that of the 99/4A. To achieve the goal of two legends per key, lowercase letters would not be supported.

To enhance user acceptance and utility of the keyboard, we needed a raised key that provided tactile feedback as it was pressed. But we also needed the economy of a Mylar key material. Our mechanical design team came up with a keyboard system that provided both advantages: using a Mylar key matrix under an

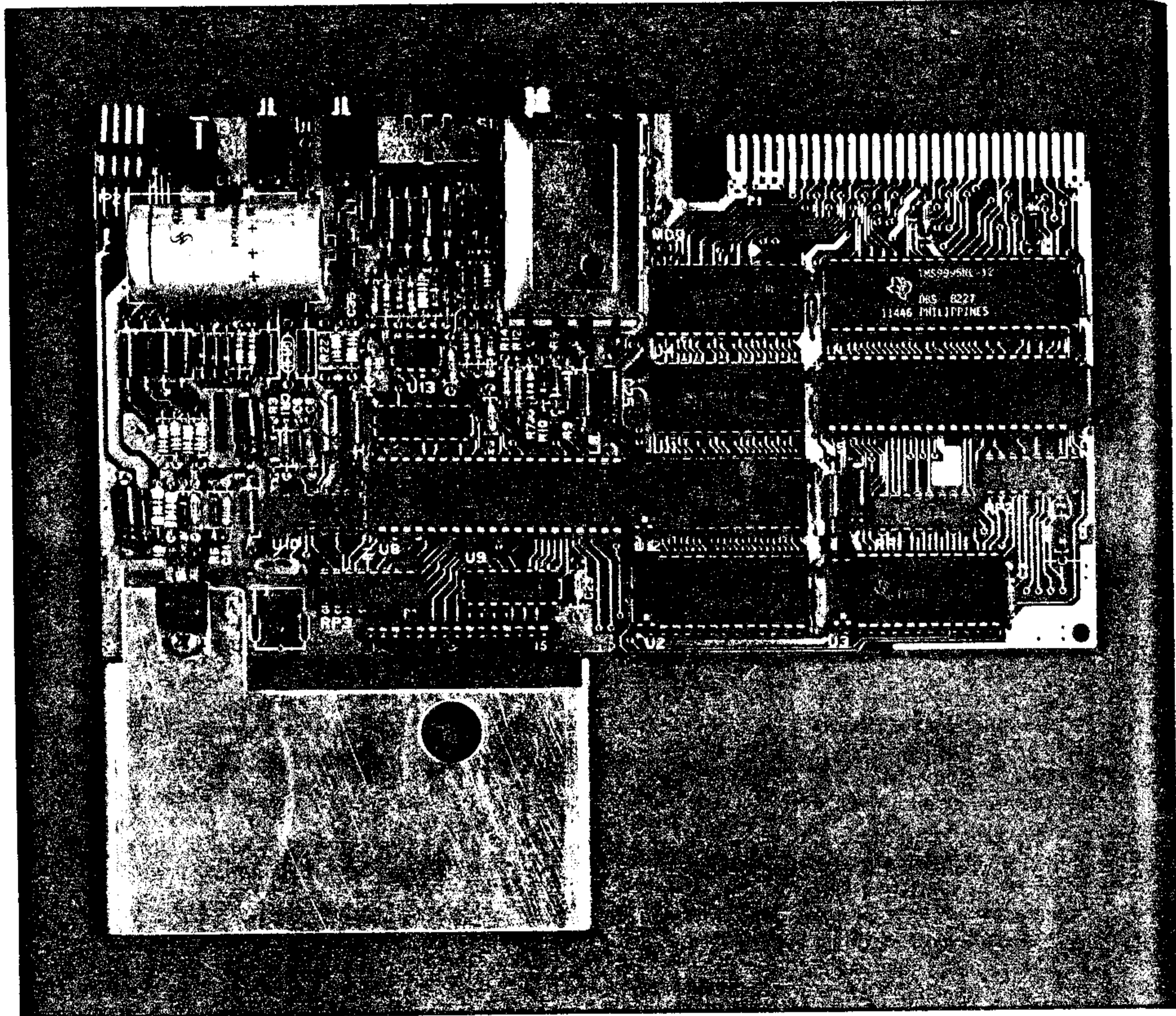


Photo 3: The main circuit board of the 99/2 contains the TMS9995 microprocessor, video controller chip, I/O controller chip, two static RAMs (4K bytes), three ROMs (the final version will have one 32K-byte ROM), a two-channel RF modulator, and the 5-volt DC regulator with heat sink.

"elastomeric" key-top assembly. The elastomeric system provides 48 raised keys in a single molded sheet—an arrangement that takes advantage of the elastomeric properties of the material. Depressing the key actuates the Mylar underneath.

Internal Architecture

We decided to use the new high-speed TMS9995 microprocessor that runs at 10.7 MHz, a standard crystal frequency. We chose the TMS9995 over the TMS9900 because it requires fewer external control components and less circuit-board space due to its smaller package (see photo 3).

The 99/2 uses a standard address/data bus system. By using a 16-bit microprocessor and DMA (direct memory access) video processing, we attained a high-speed system performance. Initially, the system had only 2K bytes of RAM. But when we expanded the system from 16K bytes of ROM to 32K bytes of ROM to allow the software to run a full file-management system without a plug-in module, we decided our overhead use of RAM was too high. We added 2K bytes of RAM to carry the overhead of the system BASIC and still give us a reasonable amount of space for the program.

The total RAM in the computer is actually 4.2K bytes, including the 256 bytes of scratch-pad memory in the TMS9995 microprocessor. That scratch-pad memory helps improve the speed of the system. The 32K-byte ROM has a physical address space of 24K bytes, and the last 8K bytes of ROM are bank switched to preserve a 32K-byte expansion-port capability (see figure 1).

The expansion port has all system control address and data bus signals available. This allows expansion with RAM, ROM, or I/O (input/output) cartridges. The 32K-byte expansion address space can be configured in

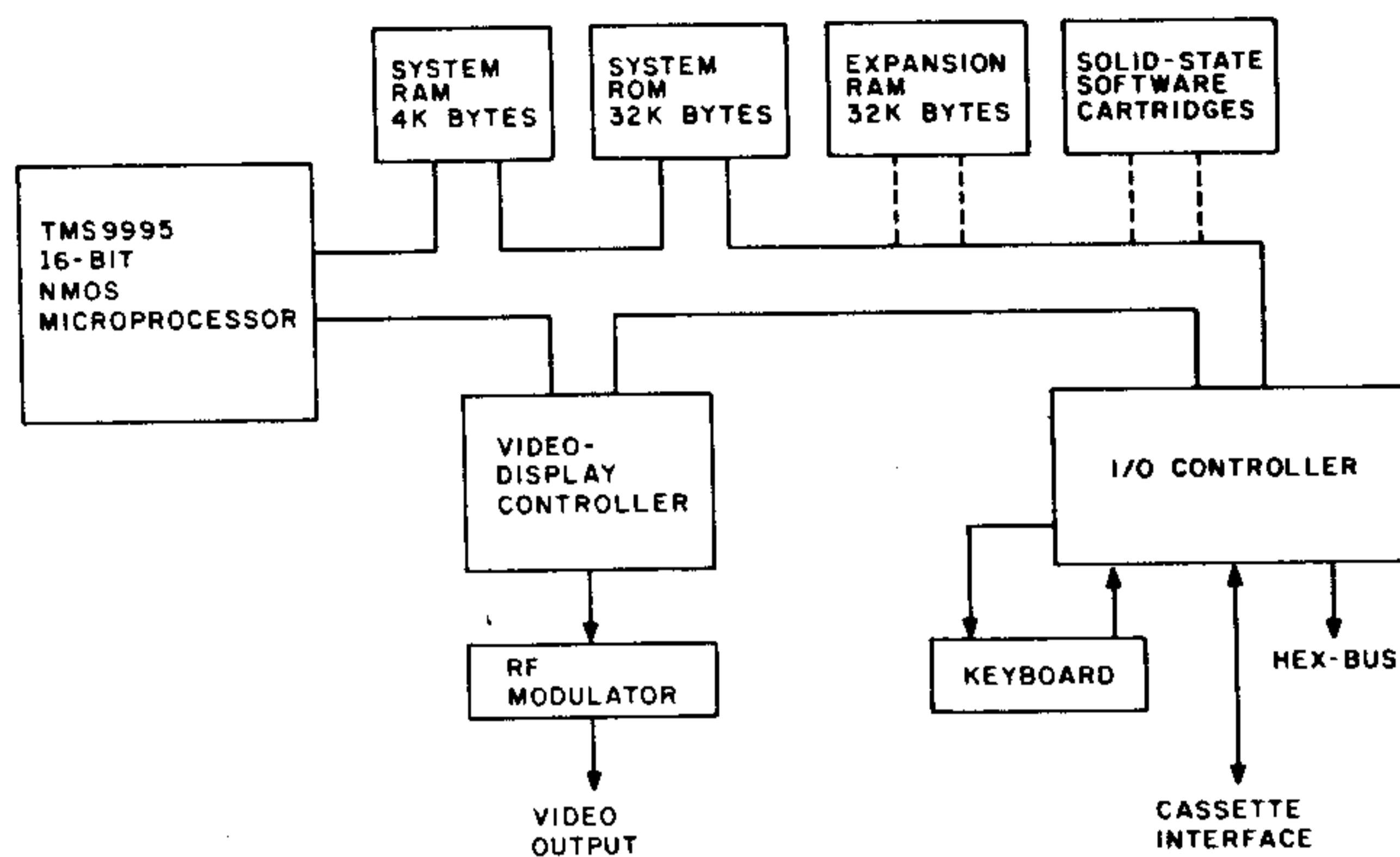


Figure 1: Internal architecture of the TI-99/2.

8K-byte increments of memory with ROM building up at the low end of the expansion address space and RAM building down from the high end of the expansion address space.

The system contains two gate arrays; one controls the video display, and the other handles I/O functions. Gate arrays are a low-cost method of designing custom ICs (integrated circuits). Starting with an array of logic elements on an IC, the manufacturer simply designs a mask to connect the elements in the final metalization procedure. To change the logic on a chip, you simply change the mask.

The video-display controller is a DMA processor that shares RAM and ROM with the microprocessor. The shared-memory structure improves the ability to effectively update and scroll the video display, thus allowing more time to be allotted to the central processing unit.

The I/O controller handles the Hex-bus interface, cassette interface, keyboard, I/O, clock generation, system reset, and RAM/ROM chip-select generation. This gate array allows higher system integration and therefore lowers the cost of the system.

The 99/2's BASIC language is architecturally different from the BASIC used in the 99/4A. To the user it appears the same, but to meet our goal of cost reduction we had to redesign the internal modules. For ex-

ample, the RAM and ROM are in different places in the 99/2 and 99/4A. Still, for the most part the BASIC commands produce the same responses as the 99/4A BASIC. In addition, the system is faster to execute and uses fewer parts.

We decided to go with Solid State Software to meet the need for a plug-in module that would allow the user to personalize the system through configuration software. The user could turn the system into a terminal emulator, a learning aid, or a game machine.

We didn't want the beginning user to have to hook up too many external parts.

We built the RF modulator into the system to simplify use. We didn't want the beginning user to have to hook up too many external parts.

The 99/2 uses a 128-character set. Each character is an 8 by 8 dot matrix. It takes 8 bytes, or 64 bits, of information within the ROM to define each character. There are 1024 bytes of ROM in the system committed to screen generation; these are unavailable for system BASIC.

One byte of RAM is required for every character location on the screen (768 bytes for a 24-character by 32-line screen display). The central

processing unit writes a code into each of those RAM locations. The gate array then reads that code and knows which of the 128 characters to display on the screen. The video controller chip and the microprocessor share the system in order to keep the display from flickering, which can be annoying to the first-time user.

The video controller chip has dominant control of the bus system. When it wants the bus, it gets it—no matter what the central processing unit is doing. We designed it that way to satisfy the refresh-rate requirements. The processor can compute only when the video controller is not using the system. The central processing unit can compute only during border times, when the electron beam is not drawing a trace on the screen, such as during the horizontal retrace, vertical retrace, and top and bottom border times.

When the screen is full of characters, the video controller has the bus 90 percent of the time, and the central processing unit can be used for computing only about 10 percent of the time, which slows down the computer drastically. To overcome this limitation, we designed the video controller with enough intelligence to know not to tie up the system when nothing is displayed on a line. At the end of every line that is generated, the processor writes what is called a blank end-of-line character. The video controller responds to this character by releasing the bus for the rest of that row on the line.

When the screen is blank, the video controller accesses the memory only 1/768 of the maximum computation time available. Because a line of BASIC code seldom fills an entire screen line, the computing time is greatly increased by this feature. If we need to compute quickly, the system has the ability to disable the video chip; then there are no interruptions to computing time. The software controls the screen and can be accessed through BASIC.

BASIC

To facilitate our computer-literacy goals for the 99/2, we provided three additional BASIC statements that are

a code into
ations. The
at code and
characters to
e video con-
prop- ssor
to. p the
which can be
e user.

chip has
bus system.
gets it—no
processing
d it that way
h-rate re-
or can com-
controller is
central pro-
only during
electron beam
the screen,
ntal retrace,
and bottom

is full of
ntroller has
ime, and the
an be used
t 10 percent
s down the
o overcome
ed the video
telligence to
ystem when
line. At the
nerated, the
is called a
acter. The
ds to this
bus for the
e.

k, the video
emory only
omputation
a line of
s an entire
ing time is
feature. If
ly, the sys-
disable the
e no inter-
e. The soft-
and can be

ter-...
vide...
nts that ar

not compatible with the 99/4A but that do allow the user to develop some assembly-language programming skills. These are the PEEK, POKE, and MCHL (machine-language execute) commands that let the user read/write directly into memory. By using the MCHL command, the user can execute an assembly-language program. Most applications on the 99/2 will not require the PEEK, POKE, and MCHL commands, but their availability gives users a chance to do assembly-language programming, even though it is not supported by the 99/4A console BASIC.

The 99/2 users manual explains any of the incompatibilities. It is important for the user to know these when writing a program that uses the assembly-language capabilities of the 99/2. Those instructions will be flagged as errors on the 99/4A. We felt that anyone who can program in assembly language will understand how to change the program to run on the 99/4A. Anything written totally in BASIC will be compatible.

System Software

The system software of the 99/2, we decided, would provide power-up initiation and system configuration definitions and would allow users to select operation in BASIC or to use an application module that had been previously inserted into the expansion port. We decided to implement a subset of the TI-99/4A console BASIC to satisfy the upward software compatibility users need to migrate to another computer system.

Because of architectural limitations, the 99/2 does not support several features—CALL COLOR, CALL SOUND, CALL JOY, CALL CHAR\$, and CALL SCREEN. The cost of the additional hardware required to support these features would have caused us to exceed our pricing goals.

In addition, the system would not use the TI-99/4A GROM language because of the cost of the hardware required to support GROM chips. We had to develop a completely new BASIC interpreter for the 99/2 that was optimized to use the speed advantages of our architecture yet pro-

vide a meaningful subset of TI-BASIC. We included a full file-management system in the system software to take full advantage of the Hex-bus peripheral system.

Within the time frame we gave ourselves, we didn't have sufficient resources to develop a complete software system for the 99/2, so we decided to go to an independent third party for the software generation. We needed a group that was willing to undertake a task of this magnitude in

the time allotted.

We contacted the University of Southwestern Louisiana (USL) in Lafayette, Louisiana, because its computer science department had worked with TI before and came highly recommended. After we discussed our concepts and needs with them, the USL team accepted the challenge. During the course of the four and a half months in which we prepared prototypes for the January 1983 Consumer Electronics Show




EXPOTEK

2723 W. Windrose Suite 3
Phoenix, Arizona 85029
1-800-528-8960




All Prices Subject To Change

COMPUTERS	PRINTERS
<p style="text-align: center; margin: 0;">ALTOS</p> <p>5-15D - \$2120 5-5D - \$3985 580-10 - \$4695 586-10 - \$5698 580-14 - \$9395 8600-12 - \$8950</p> <p style="text-align: center; margin: 0;">NORTHSTAR</p> <p>Advantage - \$2250 280A - \$1950 5m Byte - \$3350 15m Byte - \$4499</p> <p style="text-align: center; margin: 0;">TELEVIDIO</p> <p>802 - \$2599 802H - \$4450 806 - \$4950 800A - \$1250 803 - \$1890 1603 - \$2695</p> <p style="text-align: center; margin: 0;">APPLE-LOOK-A-LIKE CALL</p>	<p style="text-align: center; margin: 0;">OKIDATA</p> <p>M92A - \$480 M93A - \$830 M82A - \$385 W/Tractor & Grap. - \$465 M84P - \$965 M84S - \$ CALL Pacemark 2350P - \$1999</p> <p style="text-align: center; margin: 0;">CITOH</p> <p>F10 40 cps - \$1170 F10 55cps - \$1580 1550P - \$659 1550CD - \$709 8510P - \$399 8510BCD - \$499</p> <p style="text-align: center; margin: 0;">DATASOUTH</p> <p>DS120 - \$595 DS180 - \$1175</p> <p style="text-align: center; margin: 0;">DIABLO</p> <p>630RO - \$1749 620 - \$895</p> <p style="text-align: center; margin: 0;">NEC</p> <p>8023 - \$425 7710 - \$2045 3510 - \$1375 3550 - \$1834</p> <p style="text-align: center; margin: 0;">TRANSTAR</p> <p>140 - \$1290 130P - \$715</p>
<p style="text-align: center; margin: 0;">APPLE CARDS</p> <p>16K RAM - \$78 Z80 CARD - \$235 Videx Card - \$227 Viewmax-80 - \$175 Microsoft Prem. Pk - \$475</p>	<p style="text-align: center; margin: 0;">TERMINALS</p> <p style="text-align: center; margin: 0;">ADDS</p> <p>A1 - \$485 Viewpoint A3 - \$485</p> <p style="text-align: center; margin: 0;">HAZELTINE</p> <p>1500 - \$995 ESPRIT - \$498</p> <p style="text-align: center; margin: 0;">TELEVIDEO</p> <p>910 - \$569 925 - \$718 970 - \$1040 950 - \$899</p> <p style="text-align: center; margin: 0;">ZENITH</p> <p>Z19 - \$670 Z89 - \$2129</p>
<p style="text-align: center; margin: 0;">SOFTWARE</p> <p>All Major Brands 25% off List \$ CALL</p>	<p style="text-align: center; margin: 0;">MONITORS</p> <p>Amdek 300 - \$135 Color I - \$295 Amdek Color II - \$625 Amber - \$149 BMC Green - \$85 USI Amber - \$149</p>
<p style="text-align: center; margin: 0;">DISK DRIVE</p> <p>Microsci Apple Dr. - \$265 Rana Elite I - \$275 Rana Elite III - \$550 Rana 1000 - \$375</p>	<p style="text-align: center; margin: 0;">COMPUTER IC'S COMPLETE LINE</p> <p>SPECIAL 4164 (150ns) \$6.50ea.</p>
<p style="text-align: center; margin: 0;">MODEMS</p> <p>HAYES - MICROMODEM - \$263 HAYES - SMARTMODEM - \$199 HAYES - 1200 Baud - \$530</p>	<p style="text-align: center; margin: 0;">DISKETTES/BOXES</p> <p>Elephant - \$19 Scotch - \$25 Dysan - \$35</p>
<p style="font-size: 2em; font-weight: bold; margin: 0;">GUARANTEED LOW PRICES</p>	
<p style="font-size: 1.5em; font-weight: bold; margin: 0;">Customer Service 602-863-0759</p>	

(CES) in Las Vegas, Nevada, the USL team members provided very useful feedback on the 99/2 system architecture.

Development Schedules

The entire 99/2 system was conceived, designed, and developed in time to introduce working prototypes at the CES show. The full development cycle, from concept through production, including qualification and certification, was approximately eight months long. This achievement required the cooperative dedicated efforts of many TI groups both inside and outside the Consumer Group.

Our Semiconductor Group developed two new gate array ICs, from layout to working prototypes, in two months. Logic design of the gate arrays was done in Lubbock, Texas, by the IC design organization of the Personal Computer Division in approximately one month.

The mechanical design team of the Personal Computer Division designed and developed the console, AC

adapter, cassette/video connections, and the keyboard in time to have prototypes at the CES show, also in about four and a half months. Development included implementation of advanced assembly techniques to allow reduced manufacturing time and lower cost. Our Industrial Design team provided the styling of the console.

The full development cycle, from concept to production, was approximately eight months.

Other members of the TI Consumer Group also contributed greatly to the successful introduction of the TI-99/2. Our packaging design and artwork team worked many extra hours to provide the appropriate packaging material for the unit, while our manual-writing staff developed a series of four manuals for the 99/2 that gradually progress in complexity

to handle the range from novice to computer hobbyist.

Final Thoughts

The TI-99/2 is not just another low-cost, entry-level computer. It is a computer system designed for education and personal use. Users will find it an inexpensive introduction to microcomputers. With the use of Solid State Software cartridges such as Learn to Program and Learn to Program BASIC, individuals and schools can easily obtain instruction in computer programming.

By introducing peripherals and inexpensive software simultaneously with the computer, Texas Instruments believes this system will grow with an individual, inspire confidence in the use of computers, and foster the continued growth of the computer revolution. ■

About the Authors

Harry Littlejohn was program manager and project engineer and Mark Jander was project design engineer for Texas Instruments' TI-99/2 design group.

COMPUTER DEALERS

THIS AD WRITTEN FOR YOU ...

BYTE INDUSTRIES WANTS TO BE YOUR COMPUTER CONNECTION

Byte Industries is your one STOP shopping place for over 50 lines of microcomputers, terminals, printers and various other computer peripherals and accessories. Byte won't leave you at a STOP waiting for equipment to complete a sale. If you call in your order by 1:00 PST, we'll ship it out THE SAME DAY!

Byte Industries is searching for qualified Computer Dealers in the business of reselling the product we stock. If you're such a Dealer, you may have just found the best wholesale "Computer Connection" available.

BYTE INDUSTRIES IS THE AUTHORIZED COMPUPRO DISTRIBUTOR

I am in the business of reselling computers and related products. I would like a sales representative to contact me and tell me how I can receive my free Wholesale Pricelist and open a Dealer Account.

Name: _____
 Store Name: _____
 Address: _____
 City: _____ State: _____ Zip: _____

Return to: Byte Industries, Inc. (415) 783-8272
 21130 Cabot Blvd., Hayward, CA 94545

