NO. 32 $2.00 JANUARY 1981

MICRO™

THE 6502 JOURNAL

AIM 65 File Operations

Fun With OSI

PET Symbolic Disassembler

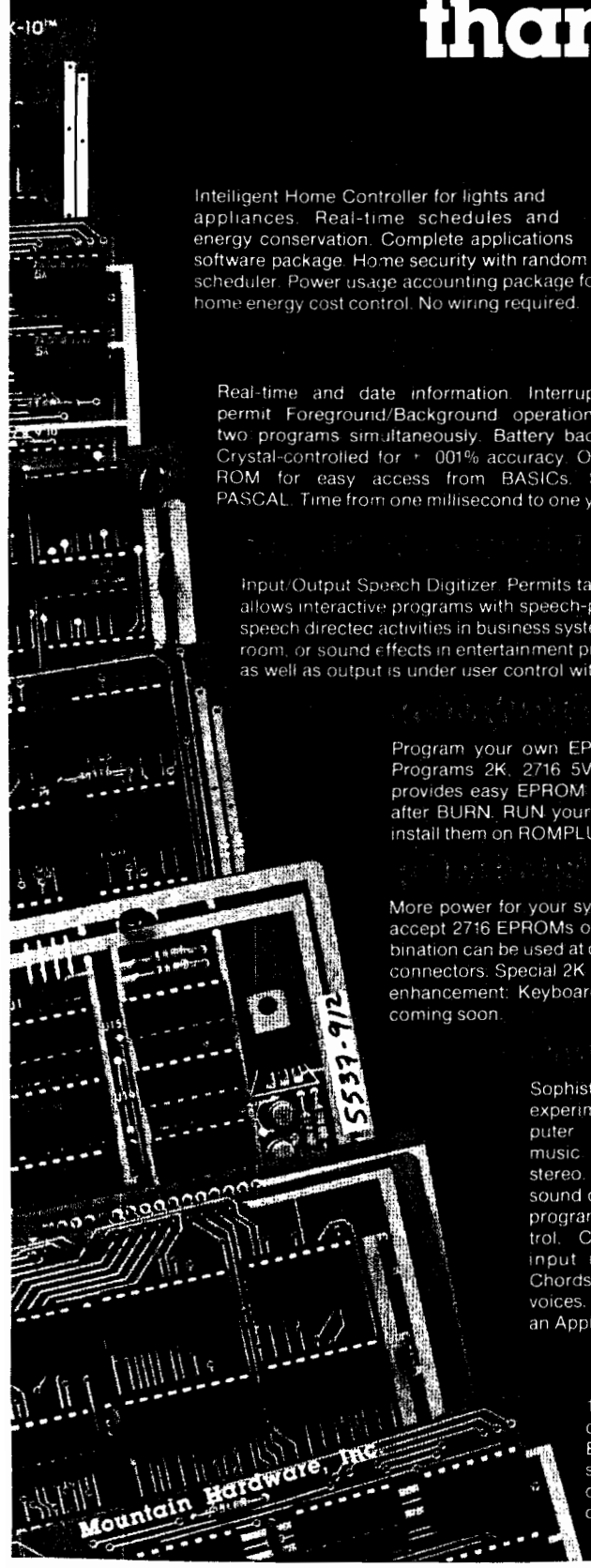
Atari Real Time

SYM Bridge Trainer

A Better Apple Search/Change

Mountain Computer makes more peripherals for the Apple Computer than Anybody.

and . . . a place to put them



Intelligent Home Controller for lights and appliances. Real-time schedules and energy conservation. Complete applications software package. Home security with random scheduler. Power usage accounting package for home energy cost control. No wiring required.

Real-time and date information. Interrupts permit Foreground/Background operation of two programs simultaneously. Battery back-up. Crystal-controlled for ± .001% accuracy. Onboard ROM for easy access from BASICs. Supports PASCAL. Time from one millisecond to one year.

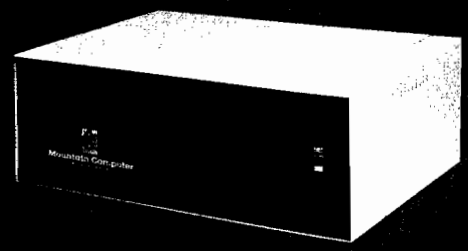
Input/Output Speech Digitizer. Permits talking programs. I/O capability allows interactive programs with speech-prompted inputs. Use output for speech directed activities in business systems, announcements in a control-room, or sound effects in entertainment programs. Easy to use because input as well as output is under user control with special software operating system.

Program your own EPROMs. Create your own firmware. Programs 2K, 2716 5V EPROMs. Disk software package provides easy EPROM programming. EPROMs are verified after BURN. RUN your programs from on-board socket or install them on ROMPLUS+.

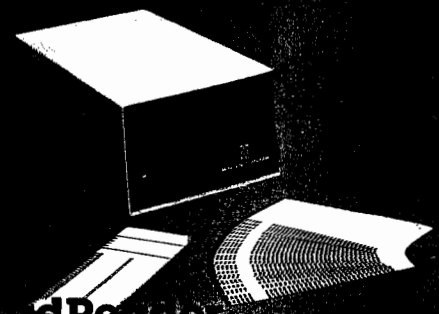
More power for your system through firmware. Six sockets accept 2716 EPROMs or ROM equivalents. Six or any combination can be used at once. Scratch-pad RAM and two TTL connectors. Special 2K ROMs available for powerful system enhancement: Keyboard Filter ROM—COPYROM—Others coming soon.

Sophistication previously available only on experimental mini and mainframe computer synthesizers. Digital instrumental music synthesizer system. 16 voices in stereo. Instrument definitions simulate the sound of real instruments—and more. Fully programmable waveforms. Envelope Control. Composition system—sheet music input using standard music notation. Chords and multi-part scoring up to 16 voices. A true instrument that anyone with an Apple can play.

16 channels analog to digital input. 16 channels digital to analog output. Eight bit resolution. Super-fast 9μ sec. conversion time. Monitor and output to the real world. All on one card.



By popular demand! Eight more slots for your Apple. Attractive sturdy enclosure. Its own heavy duty power supply. Easy to use. Address cards in Expansion Chassis the same way as in your Apple. Only one additional command to specify in Apple or in Expansion Chassis. Compatible with all Apple peripherals.



At last! An intelligent, high-quality device for data entry from user-marked cards. Implement BASIC programming, examination scoring, inventory maintenance and other applications requiring off-line data preparation for batch entry later. Connects to any computer having RS-232 interface. Software and cards are available for jobs in business, science and education.

MOUNTAIN COMPUTER has the most comprehensive line of Apple peripherals available. Anywere. From anybody. We know the Apple inside and out and are committed to providing the most innovative and unique products to expand and enhance its capabilities and use. After all, we were the first company to make an Apple peripheral—except Apple Computer.

Available at Apple Dealers worldwide.

Mountain Computer INCORPORATED

300 Harvey West Blvd., Santa Cruz, CA 95060
(408) 429-8600 TWX 910 598-4504

Apple is a trademark of Apple Computer Inc.

OSI SOFTWARE FOR OSI OSI

A JOURNAL FOR OSI USERS!!

The Aardvark Journal is a bimonthly tutorial for OSI users. It features programs customized for OSI and has run articles like these:

1) Using String Variables.

2) High Speed Basic On An OSI.

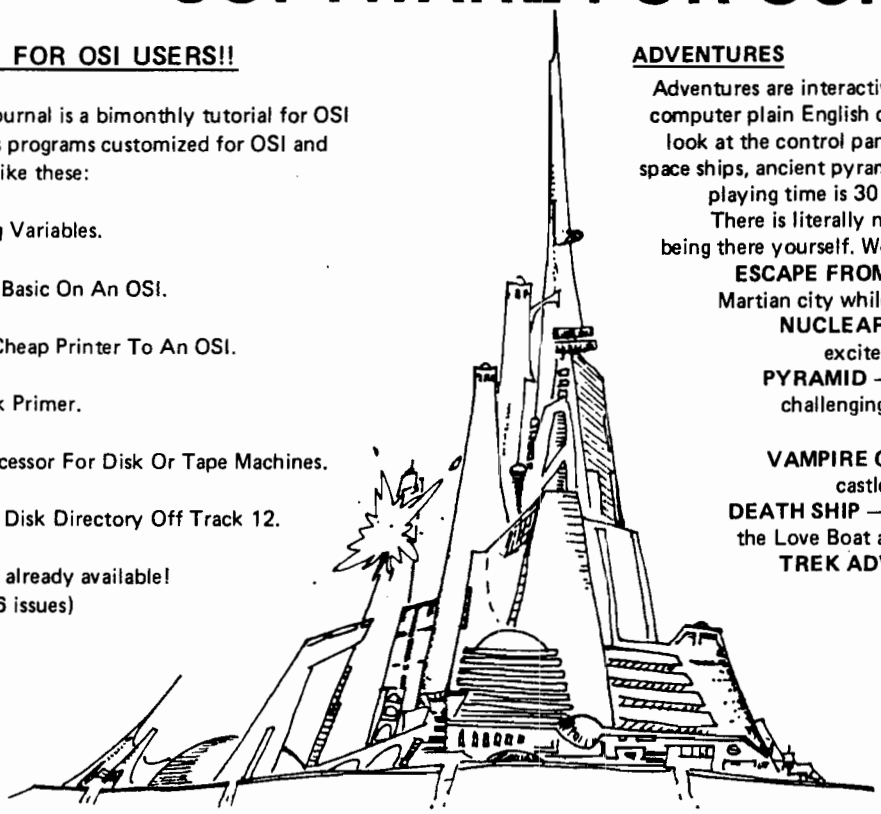
3) Hooking a Cheap Printer To An OSI.

4) An OSI Disk Primer.

5) A Word Processor For Disk Or Tape Machines.

6) Moving The Disk Directory Off Track 12.

Four back issues already available!
$9.00 per year (6 issues)



ADVENTURES

Adventures are interactive fantasies where you give the computer plain English commands (i.e. take the sword, look at the control panel.) as you explore alien cities, space ships, ancient pyramids and sunken subs. Average playing time is 30 to 40 hours in several sessions. There is literally nothing else like them – except being there yourself. We have six adventures available.

ESCAPE FROM MARS – Explore an ancient Martian city while you prepare for your escape.
NUCLEAR SUBMARINE – Fast moving excitement at the bottom of the sea.
PYRAMID – Our most advanced and most challenging adventure. Takes place in our own special ancient pyramid.
VAMPIRE CASTLE – A day in old Drac's castle. But it's getting dark outside.
DEATH SHIP – It's a cruise ship – but it ain't the Love Boat and survival is far from certain.
TREK ADVENTURE – Takes place on a familiar starship. Almost as good as being there.

$14.95 each

NEW SUPPORT ROMS FOR BASIC IN ROM MACHINES

C1S – for the C1P only, this ROM adds full screen edit functions (insert, delete, change characters in a basic line.), Software selectable scroll windows, two instant screen clears (scroll window only and full screen.), software choice of OSI or standard keyboard format, Bell support, 600 Baud cassette support, and a few other features. It plugs in in place of the OSI ROM. NOTE: this ROM also supports video conversions for 24, 32, 48, or 64 characters per line. All that and it sells for a mesly $39.95.
C1E/C2E for C1/C2/C4/C8 Basic in ROM machines.
This ROM adds full screen editing, software selectable scroll windows, keyboard correction (software selectable), and contains both an extended machine code monitor and a fix for the string handling bug in OSI Basic!! It has breakpoint utilities, machine code load and save, block memory move and hex dump utilities. A must for the machine code programmer replaces OSI support ROM. Specify system! $59.95
STRING BUG FIX (replaces basic ROM chip number 3)
All this chip does is to replace the third basic ROM and correct the errors that were put into the ROM mask. $19.95

DATA SHEETS

OS65D LISTING
Commented with source code, 83 pages. $24.95
THE (REAL) FIRST BOOK OF OSI
65 packed pages on how OSI basic works. Our best selling data sheet. $15.95
OSI BASIC IN ROM
Ed Carlson's book of how to program in basic. Now available from Aardvark. $8.95

P.C. BOARDS

MEMORY BOARDS!! – for the C1P. – and they contain parallel ports!
Aardvarks new memory board supports 8K of 2114's and has provision for a PIA to give a parallel ports! It sells as a bare board for $29.95. When assembled, the board plugs into the expansion connector on the 600 board. Available now!

REAL SOUND FOR THE C1P – and it's cheap! This bare board uses the TI sound chip to give real arcade type sound. The board goes together in a couple of hours with about $20.00 in parts. Bare board, plans, and sample program – $15.95

ARCADE AND VIDEO GAMES

ALIEN INVADERS with machine code moves – for fast action. This is our best invaders yet. The disk version is so fast that we had to add selectable speeds to make it playable.
Tape - $10.95 – Disk - $12.95

TIME TREK (8K) – real time Startrek action. See your torpedoes move across the screen! Real graphics – no more scrolling displays. $9.95

STARFIGHTER – a real time space war where you face cruisers, battleships and fighters using a variety of weapons. Your screen contains working instrumentation and a real time display of the alien ships. $6.95 in black and white - $7.95 in color and sound.

SEAWOLFE – this one looks like it just stepped out of the arcades. It features multiple torpedoes, several target ships, floating mines and real time time-to-go and score displays. – $6.95 in black and white $7.95 in color and sound.

SCREEN EDITORS

These programs all allow the editing of basic lines. All assume that you are using the standard OSI video display and polled keyboard.
C1P CURSOR CONTROL – A program that uses no RAM normally available to the system. (We hid it in unused space on page 2). It provides real backspace, insert, delete and replace functions and an optional instant screen clear.
$11.95
C2/4 CURSOR. This one uses 366 BYTES of RAM to provide a full screen editor. Edit and change lines on any part of the screen. (Basic in ROM systems only.)
FOR DISK SYSTEMS – (65D, polled keyboard and standard video only.)
SUPERDISK. Contains a basic text editor with functions similar to the above programs and also contains a renumberer, variable table maker, search and new BEXEC* programs. The BEXEC* provides a directory, create, delete, and change utilities on one track and is worth having by itself. – $24.95 on 5" disk - $26.95 on 8".

DISK UTILITIES

SUPER COPY – Single Disk Copier
This copy program makes multiple copies, copies track zero, and copies all the tracks that your memory can hold at one time – up to 12 tracks at a pass. It's almost as fast as dual disk copying. – $15.95
DISK CATALOGER
This utility reads the directory of your disks and makes up an alphabetic list off all your programs and what disks they are on. $14.95
MACHINE CODE RENUMBERER
(C2/4-MF only)
Renumbers all or part of a program at machine code speeds. – $15.95

This is only a partial listing of what we have to offer. We now offer over 100 programs, data sheets, ROMS, and boards for OSI systems. Our $1.00 catalog lists it all and contains free program listings and programming hints to boot.

Aardvark Technical Services • 1690 Bolton • Walled Lake, MI 48088
(313) 669-3110 or (313) 624-6316

DATA CAPTURE 4.0©

The most advanced and easiest to use telecommunications program for use with the MICROMODEM II™ or the Apple COMMUNICATIONS CARD™.

Q. Will DATA CAPTURE 4.0 work with my Communications Card™ and a modem?
A. It makes using the Comm. Card almost as easy as using the Micromodem II.

Q. Do I need an extra editor to prepare text for transmission to another computer?
A. No. DATA CAPTURE 4.0 gives you control of the text buffer. You can use DATA CAPTURE 4.0 to create text.

Q. Can I edit the text I have prepared?
A. Yes. You can insert lines or delete any lines from the text.

Q. How about text I have captured. Can I edit that?
A. As easily as the text you have prepared yourself. You can delete any lines you don't want to print or save to a disk file. You can also insert lines into the text.

Q. Just how much text can I capture with DATA CAPTURE 4.0?
A. If the system with which you are communicating accepts a stop character, most use a Control S, you can capture an unlimited amount of text.

Q. How does that work? And do I have to keep an eye on how much I have already captured?
A. When the text buffer is full the stop character is output to the other system. Then DATA CAPTURE 4.0 writes what has been captured up to that point to a disk file. This is done automatically.

Q. Then what happens?
A. Control is returned to you and you can send the start character to the other system. This generally requires pressing any key, the RETURN key or a Control Q.

Q. Are upper and lower case supported if I have a Lower Case Adapter?
A. Yes. If you don't have the adapter an upper case only version is also provided on the diskette.

Q. Do I need to have my printer card or Micromodem II™ or Communications Card™ in any special slot?
A. No. All this is taken care of when you first run a short program to configure DATA CAPTURE 4.0 to your system. Then you don't have to be concerned with it again. If you move your cards around later you can reconfigure DATA CAPTURE 4.0.

Q. Do I have to build a file on the other system to get it sent to my Apple?
A. No. If the other system can list it you can capture it.

Q. How easy is it to transmit text or data to another system?
A. You can load the text or data into DATA CAPTURE 4.0 from the disk and transmit it. Or you can transmit what you have typed into DATA CAPTURE 4.0.

Q. How can I be sure the other system receives what I send it?
A. If the other system works in Full Duplex, it 'echoes' what you send it, then DATA CAPTURE 4.0 adjusts its sending speed to the other system and won't send the next character until it is sure the present one has been received. We call that 'Dynamic Sending Speed Adjustment'.

Q. What if the other system works only in Half Duplex.
A. A different sending routine is provided for use with Half Duplex systems.

Q. What if I want to transmit a program to the other system?
A. No problem. You make the program into a text file with a program that is provided with DATA CAPTURE 4.0, load it into DATA CAPTURE 4.0 and transmit it.

Q. What type files can I read and save with DATA CAPTURE 4.0?
A. Any Apple DOS sequential text file. You can create and edit EXEC files, send or receive VISCIALC© data files. send or receive text files created with any editor that uses text files.

Q. Can I leave DATA CAPTURE 4.0 running on my Apple at home and use it from another system?
A. Yes. If you are using the Micromodem II™ you can call DATA CAPTURE 4.0 from another system. This is handy if you are at work and want to transmit something to your unattended Apple at home.

Q. Where can I buy DATA CAPTURE 4.0?
A. Your local Apple dealer. If he doesn't have it ask him to order it. Or if you can't wait order it directly from Southeastern Software. The price is $65.00. To order the Dan Paymar Lower Case Adapter add $64.95 and include the serial number of your Apple.

Q. If I order it directly how can I pay for it?
A. We accept Master Charge, Visa or your personal check. You will get your order shipped within 3 working days of when we receive it no matter how you pay for it. Send your order to us at the address shown or call either of the numbers in this advertisement. You can call anytime of day, evening or Saturdays.

Q. I bought DATA CAPTURE 3.0 and DATA CAPTURE 4.0 sounds so good I want this version. What do I do to upgrade?
A. Send us your original DATA CAPTURE 3.0 diskette and documentation, the $35.00 price difference and $2.50 for postage and handling. We will send you DATA CAPTURE 4.0 within 3 working days of receiving your order.

Q. What kind of support can I expect after I buy it?
A. If you have bought from Southeastern Software in the past you know we are always ready to answer any questions about our products or how to use them.

Requires DISK II™, Applesoft II™ and 48K of Memory

DATA CAPTURE 4.0©
Copyright© 1980-Southeastern Software

* Apple™, Apple II Plus™, Disk II™ and APPLESOFT II™ are trademarks of Apple Computer Company.

* Micromodem II™ is a trademark of D.C. Hayes Associates, Inc.

* Visicalc©-Copyright by Software Arts, Inc.

We welcome your personal check. We also accept Visa and Master Charge.

Southeastern Software

Dept. MK
6414 Derbyshire Drive • New Orleans, LA 70126
504/246-8438 504/246-7937

NO. 32 JANUARY 1981

MICRO™

THE 6502 JOURNAL

CONTENTS

7 KEYBOARD ENCODING
How to connect a keyboard to your computer
By George Young

17 A BETTER APPLE SEARCH/CHANGE
Remove the length restriction of the CHANGE function
By J.D. Childress

21 VECTORS AND THE CHALLENGER 1P
Find out exactly what a vector is
By Mike Bassman

23 PET SYMBOLIC DISASSEMBLER
Disassembled programs become more understandable
By Werner Kolbe

29 AIM 65 FILE OPERATIONS
The third part of this series, using BASIC
By Christopher J. Flynn

35 ATARI REAL TIME
Drawing with graphics and several colors
By Charley and Mary Kozarski

37 FULL DISASSEMBLY LISTING ON SMALL SYSTEMS
Produce "camera-ready" listings without large memory
By Ralph Tenny

41 SYM BRIDGE TRAINER
Simple program, easily adapted for any 6502
By Len Green

53 MAKE A CLEAR PLASTIC COVER FOR YOUR APPLE
Protect your Apple, and watch while you compute
By E.J. Neiburger, D.D.S.

57 SEARCHING STRING ARRAYS
Searching a large string array becomes easier
By Gary B. Little

65 INTERFACING THE 6522 VERSATILE INTERFACE ADAPTER
This chip will allow you to add more capabilities to your system
By Marvin L. DeJong

75 FUN WITH OSI
Checker game using the polled keyboard technique
By Les Cain

DEPARTMENTS

5 Editorial — MICRO Goes to School — Robert M. Tripp
6 Letterbox
36 New Publications
51 PET Vet — Loren Wright
61 MICRO Club Circuit
79 Semi-Annual Index
87 The MICRO Software Catalog: XXVIII
90 6502 Bibliography: Part XXVIII — William R. Dial
95 Advertisers' Index

STAFF

Editor/Publisher
ROBERT M. TRIPP

Associate Publisher
RICHARD RETTIG

Associate Editor
MARY ANN CURTIS

Art Director
GARY W. FISH

Typesetting
EMMALYN H. BENTLEY

Director Marketing
JAMES ANDERSON

Advertising Manager
L. CATHERINE BLAND

Circulation Manager
CAROL A. STARK

MICRO Specialists
APPLE: FORD CAVALLARI
PET: LOREN WRIGHT
OSI: PAUL GEFFEN

Comptroller
DONNA M. TRIPP

Bookkeeper
KAY COLLINS

MICRO is published monthly by:
MICRO INK, Inc., Chelmsford, MA 01824
Second Class postage paid at:
Chelmsford, MA 01824
Publication Number: COTR 395770
ISSN: 0271-9002

Subscription rates: U.S.	$15.00 per year
Foreign surface mail	$18.00 per year
Central America air	$27.00 per year
So. Amer/Europe air	$33.00 per year
Other air mail	$39.00 per year

For back issues, subscriptions, change of address or other information, write to:
MICRO
P.O. Box 6502
Chelmsford, MA 01824

Copyright © 1980 by MICRO, INK, Inc.
All Rights Reserved

MANENT RELIEF

Word Processing pro

Apple PIE + Formatter

Apple PIE (Programma International Editor) and FORMAT (text formatter) offer full strength solutions to today's word processing problems. These versatile, powerful programs provide document preparation and word processing capabilities previously found only on much larger computer systems.

PIE is a general purpose, full screen editor that uses control keys and function buttons to provide a full range of editing capabilities such as search and replace, delete, copy, insert, move. Changes may be made directly anywhere on the screen and are shown as they are performed.

FORMAT uses simple instructions embedded in the input text to describe the desired appearance of the final document. It handles centering, underlining, indenting, page numbering, margins, headers, footers, even form letters, and includes a proofing capability.

These high-quality, cost-effective programs come with comprehensive documentation and run on a 32K Apple II. They are available through your local computer store or direct from Programma International, Inc. at the introductory price of $79.95*.

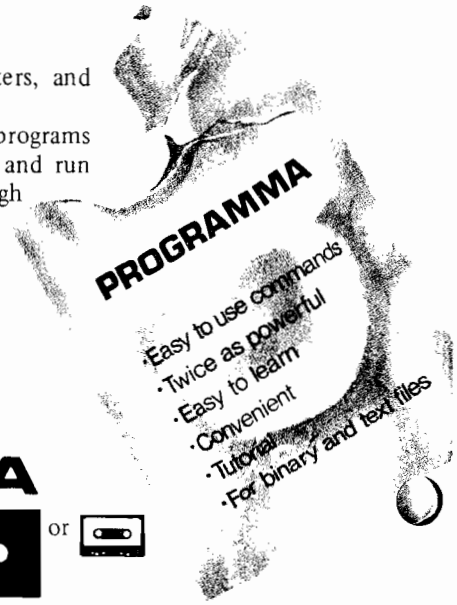
VIDEX VERSION T.M.
DOUBLE VISION T.M.
SUPR TERM VERSION T.M.
STANDARD VERSION

*December 1, $129.95.

✓11

PROGRAMMA
3400 Wilshire Boulevard
Los Angeles, California 90010
(213) 384-0579

or



Simple enough for the beginner. Versatile enough for the professional.

MICRO Editorial

MICRO Goes To School

Almost every progressive school now uses computers to teach about computers, to teach other subjects, and for administrative and other non-teaching tasks. Since a school may already own a mini- or maxi-computer, or have access to one within the town or school district, the assumption is often made that it will be cheaper to buy additional devices to tie into an existing system than to purchase new systems. This assumption may be erroneous since many of the operations can be more effectively performed by a dedicated microcomputer than by a large time-shared central computer.

A local high school rents 10 teletype terminals at over $500 each, per year, to access a medium-sized system at the regional vocational high school. One or two of the terminals are used for administrative tasks, which may require a large disk storage and/or voluminous print-outs that could be best handled by the central system. The remaining terminals are for students learning computer programming, especially programming in BASIC. Under this remote terminal approach, students can only use the teletype with its obvious limitations of speed and minimal characters, can only access the central computer at certain times, do not get the "hands-on" experience of actually operating a microcomputer, do not get the graphics and instant video response of the micro, and probably do not have the range of programs available—games and others—that they would get on a micro. Meanwhile, the school system is paying as much to rent one terminal for two years as it would cost to buy a complete 6502-based video system. This does not even include the hidden expenses of the telephone, support for the central system, special main-frame programming, and so forth.

This is absurd, but understandable. The large computer manufacturers have spent time, money and effort to convince the educational community of the value of a central time-sharing system. No microcomputer company has had the resources or time to tackle this monumental task alone. Advances to date have been mostly due to individual computer stores, "enlightend" teachers, and the occasional outside individual, getting the school systems to think along micro lines.

What better way to learn about a computer than to have the entire system at your disposal—no mysterious big black boxes in special air conditioned rooms located off-limits in another part of the building, or even across town. Probably more programs have been created for learning about computers using popular 6502-based microcomputers than were ever devised for any mainframe.

Beyond the teaching aspect of the micro in the school, there are numerous possibilities in school administration. Students could even write some of the programs. [Note: teachers had better carefully check the programs for accuracy. Some of the computer students are pretty sharp and might just "adjust" their grades!]

MICRO would like to try, with the aid of its readers, to help expand microcomputer usage in schools. There are several simple steps which you can take to help. First, become aware of what your local school system is doing in the computer field. Is it keeping up with the microcomputer revolution? If so, please let us know—in detail—how your school system is doing it. Perhaps the people responsible for the computers would be willing to write up some of their experiences, decisions, and so forth, providing assistance to other schools which are out-of-date.

If your school system is not keeping up, then offer to help. Find out who in your school system has the responsibility for the computer field and invite him or her over to see your equipment in operation. Show off some of your system's features and emphasize its low cost, its capabilities in the teaching area, and the fact that there is a lot of support through books and magazines such as MICRO. Explain that there are lots of programs available, which are generally inexpensive, covering many areas [show them the MICRO Software Catalog]. Offer to help the school system evaluate its needs and requirements. You would be surprised at how small a push it may take to get things going. If there is enough reader interest, we would like to actively pursue this topic. This could include columns on educational applications of the 6502-based systems, special software listings of educationally related programs, discussions of school system needs, and so forth.

Robert M. Tripp

Editor/Publisher

About the Cover

MICRO Goes To Town

Our cover photograph, by Loren Wright, is of Littleton, New Hampshire. As we look at towns like this, the question arises, "How can a town or city effectively utilize microcomputers?" While some data collection and processing requirements of a government probably require the use of medium to large computer systems, other functions could be more efficiently and economically handled by a dedicated microcomputer.

Our cover shows a micro being used as a processor for voter registration. This simple application could include printing labels for the mailing of voter information, updating lists as voters are added, dropped, or change their addresses, recording changes in party affiliation, and so forth. Confidential material would not have to go outside the office; updating could be kept more current, and specialized lists easily prepared as required for a special election or other need; since this task would take only a small portion of the normal business week, the micro would be available to other departments and for other uses.

Other small data bases typically kept by local governments covering property assessments, zoning regulations, building permits, dog registrations, licenses, etc. could often be done on a microcomputer. Is your town or city keeping pace with the growing microcomputer capabilities?

MICRO Letterbox

Dear Editor:

Ohio Scientific C1P has a fine BASIC, but I envy the TRS-80 crowd and their PRINT USING. Since some programs must calculate and print dollar amounts, this simple routine puts the result of the calculation off to the right side of the screen, adds any required zeros and lines up the decimal points.

The only thing it does not do is round off amounts, and that would require only adding a line, DO = DO + .005 before transferring control to the subroutine.

Note that line 100 should be early in the program so the computer does not run across it each time the subroutine is called.

Zero amounts are printed correctly, but any dollar amount with more than six significant figures will cause the interpreter to go to scientific notation, which may give gross errors. Since I never have to worry about large amounts of money, that problem does not bother me.

```
  100 Z$ = "00"
  200 REM PROGRAM
      CALCULATES DOLLAR
      VALUE--DO
  210 REM SUBROUTINE
      PRINTS
  220 GOSUB10000
  230 REM REST OF
      PROGRAM
      ...................
10000 DO$ = STR$(DO):IFDO
      = INT(DO)THEN DO$
      = DO$ + "."
10010 FORP = ITOLEN(DO $):
      IFMID$(DO$,P,1)
      = "."THEN10030
10020 NEXTP
10030 O$ = DO$ + Z$:O$
      = "$" + LEFT$(O$,
      P + 2)
10040 PRINTTAB(20 - LEN
      (O$))O$
10050 RETURN
```

Robert V. Davis
1857 South Fourth
Salina, Kansas 67401

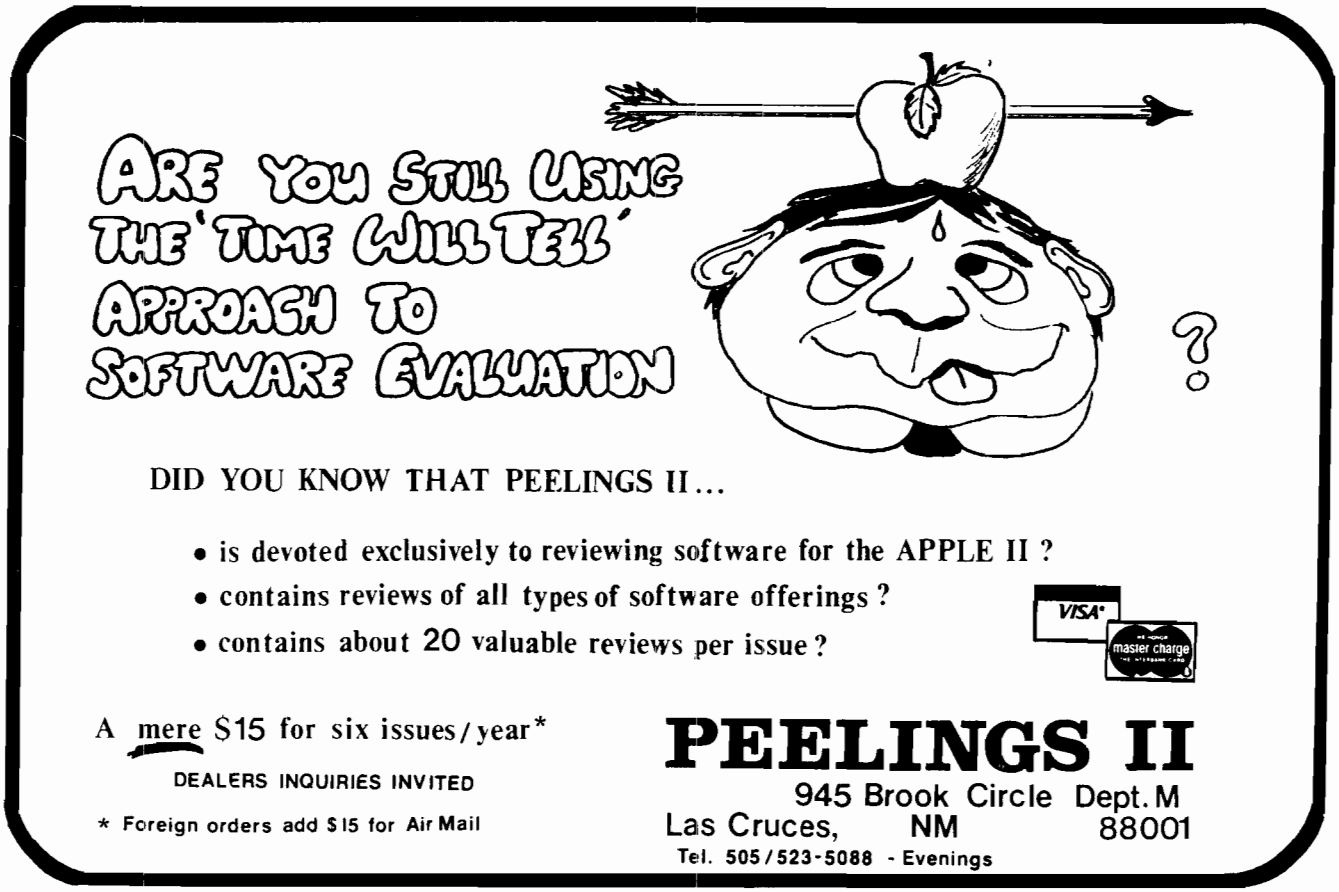
Dear Editor:

The following program will disable the "LIST" command on Ohio Scientific ROM BASIC computers. Programs can be loaded and run, but not listed. Normally, I do not like non-listable programs and accept the challenge to try to list them. Thus, I offer your readers the challenge of listing this program after it has been run. If you need a hint, read MICRO 25:15.

```
 10 FORX = 546TO586
 20 READQ:POKEX,Q
 30 NEXTX
 40 POKE11,62:POKE12,2
 50 X = USR(1)
 80 DATA230,195,208,2,230,
    196,165,195,141
 81 DATA51,2,165,196,141,52,
    2,173,20,0,201
 82 DATA153,208,2,169,128,
    76,197,0,169,76
 83 DATA133,188,169,34,133,
    189,169,2,133
 84 DATA190,96
OK
```

Tim Finkbeiner
Earl Morris
Midland, Michigan

(continued on page 76)



Keyboard Encoding

It may occur sooner, or it may occur later, but you *will* want to connect a keyboard to your computer.

George Young
Sierra High School
Tollhouse, CA 93667

I will begin with a fairly simple encoder for 16 separate switches. Figure 1 gives the circuit. The 16 switches may be discrete switches or they may be 16 switches in a surplus or recycled calculator keypad.

The 16 switches are connected to the inputs of a 74150 1:16 multiplexer (MUX). Two 74161's or 74163's are used. One 74161 is used as a counter and the second is used as a register. A clock formed from ½ of a 7404 hex inverter is used to drive the 74161 clock inputs.

The outputs of the 74161 counter drive the MUX and the inputs to the 74161 register. The binary code presented to the MUX address inputs causes the MUX to scan each one of its inputs looking for a low. Closing one of the 16 switches connected to the input of the 74150 will cause this input to go low. The output of the 74150 will go high when a low is detected on one of its inputs. This high is time coincident, or time dependent, on the counter binary code at the instant of switch closure. This is the basic property of multiplexing.

The high out of the MUX is inverted with a 7404 hex inverter stage to produce a low on a key closure. This low is fed to the PE (preset enable) of the two 74161's. This low will cause the register 74161 to place the counter code that corresponds to the switch closed on its four output lines. This code will remain unchanged at the register outputs until another key is depressed. Thus, the code, or data out of the encoder is latched by the 74161 register stage.

The output code of the encoder can be seen by connecting 4 LEDs to the 4 register output lines. Figure 2 shows one way to connect the LEDs for this purpose. A logic probe may also be used to probe each output line to verify the code generated by the encoder.

When keyboards are encoded, a strobe pulse is also usually needed for the computer. Keyswitches bounce when connected to TTL chips, so a key debounce circuit is also usually required. Figure 3 shows a NAND gate section gating the clock and driving a 74122 retriggerable one-shot. This combination will debounce the key contacts and generate an active high and an active low strobe pulse.

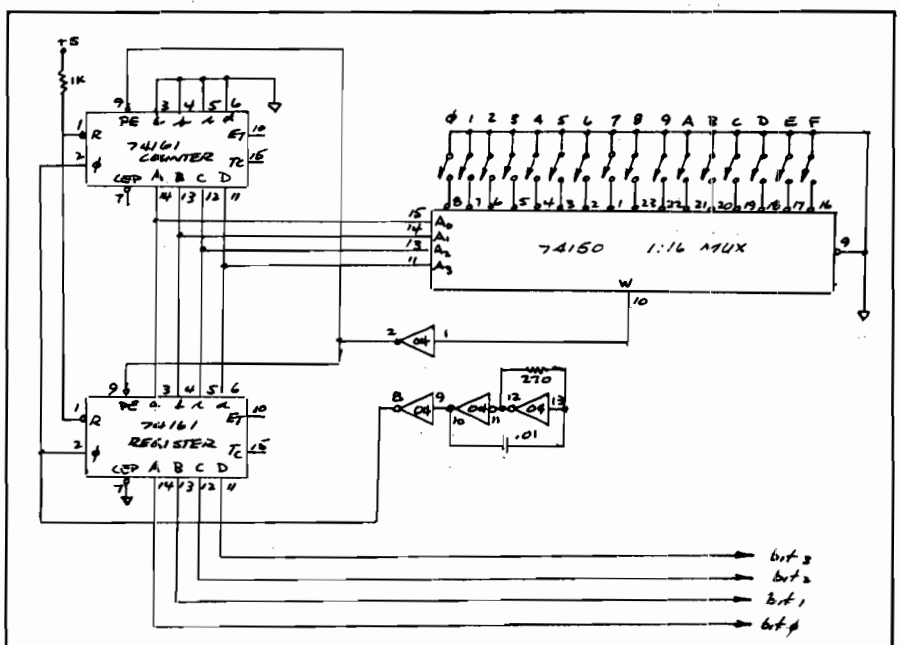


Figure 1. **Encoder for 16 SPST switches.**

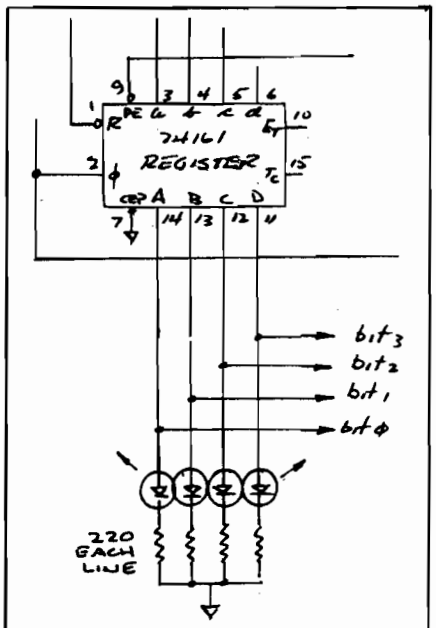


Figure 2. **LED monitor for checking output of encoder.**

To encode more than 16 keys, another 74150 would be needed with another 16 lines to the switch assembly. Additional 74161's used as counters and registers would be required to encode the added keyswitches. The major drawback to using this approach would be the size of the cable (minimum number of wires to the keyboard is 33; 32 for the keyswitches and one ground.) To circumvent this wire bundle, a different type of encoding is required.

Matrix Encoding

Only 8 wires need run from the switch assembly to the encoder circuitry to encode 16 switches, if the switches are arranged in a 4 × 4 matrix. The code is generated by a keyswitch matrix by using a 1:4 decoder to drive one set of 4 lines, which we will call the column lines, and feeding the other 4 lines to the inputs of a MUX. We will call these 4 lines the row lines.

The 74161 counter and register stages are again used in this circuit and they are again driven by the 7404 clock. Two of the counter outputs drive the two address lines of the decoder while the remaining two outputs drive the address inputs of the MUX. A 1:4 decoder (74139) or a 1:10 decoder can be used, and if the unused address lines are grounded, the decoder will function as a 1:4 decoder. A 1:8 MUX is used, operating as a 1:4 MUX again by grounding the unused address lines.

The counter code driving the address lines causes the decoder to put out sequential and mutually exclusive lows on its output pins. These lows feed the column lines of the switch matrix. The row lines feed the input pins of the MUX, and the remaining two 74161 counter outputs drive the address inputs of the MUX. Again, the MUX will look at each of its inputs to see if one of them is low. Closing a switch in the matrix causes the active low output of the MUX to go low, and the active high output to go high. This active high or low out of the MUX is again time coincident with the binary code, operating both decoder and MUX so the 74161 register outputs will again latch this code and you can examine it with LEDs on the 74161 register output lines, or with a logic probe.

Debouncing of the keyswitches and generation of strobes is again done with a NAND gate section and a 74122 retriggerable one-shot. All of this circuitry is shown in figure 4.

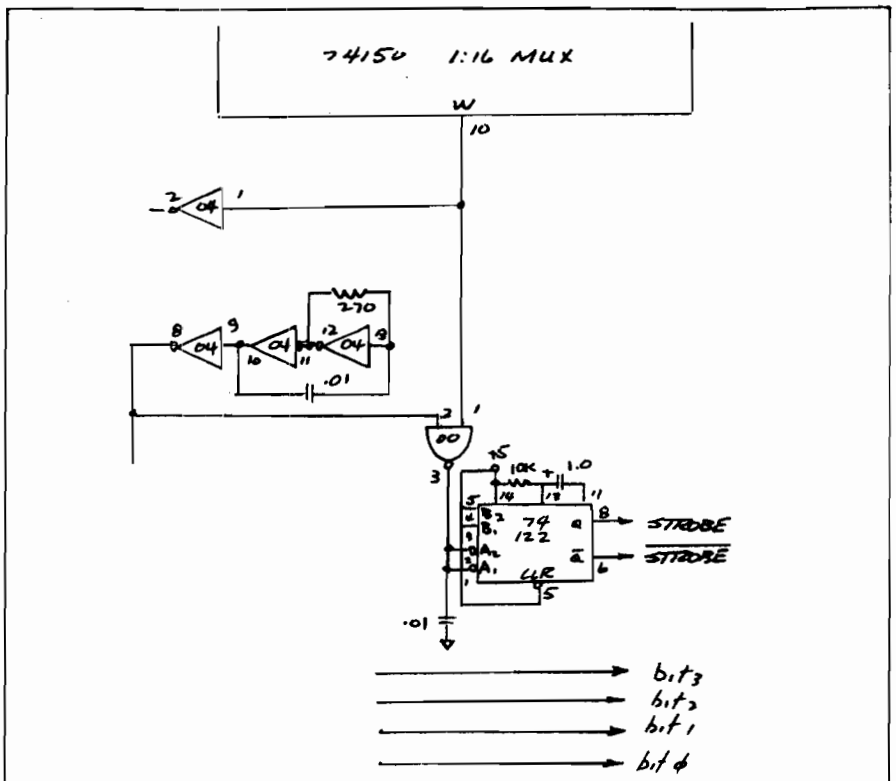


Figure 3. **Key contact debounce/strobe generator.**

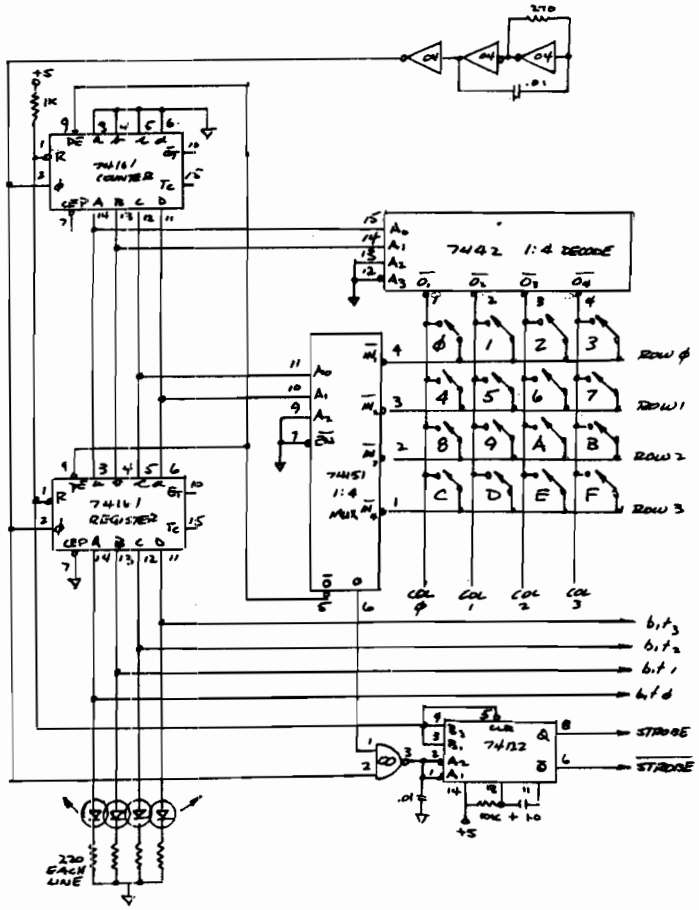


Figure 4. **Basic matrix encoder for 16 switches.**

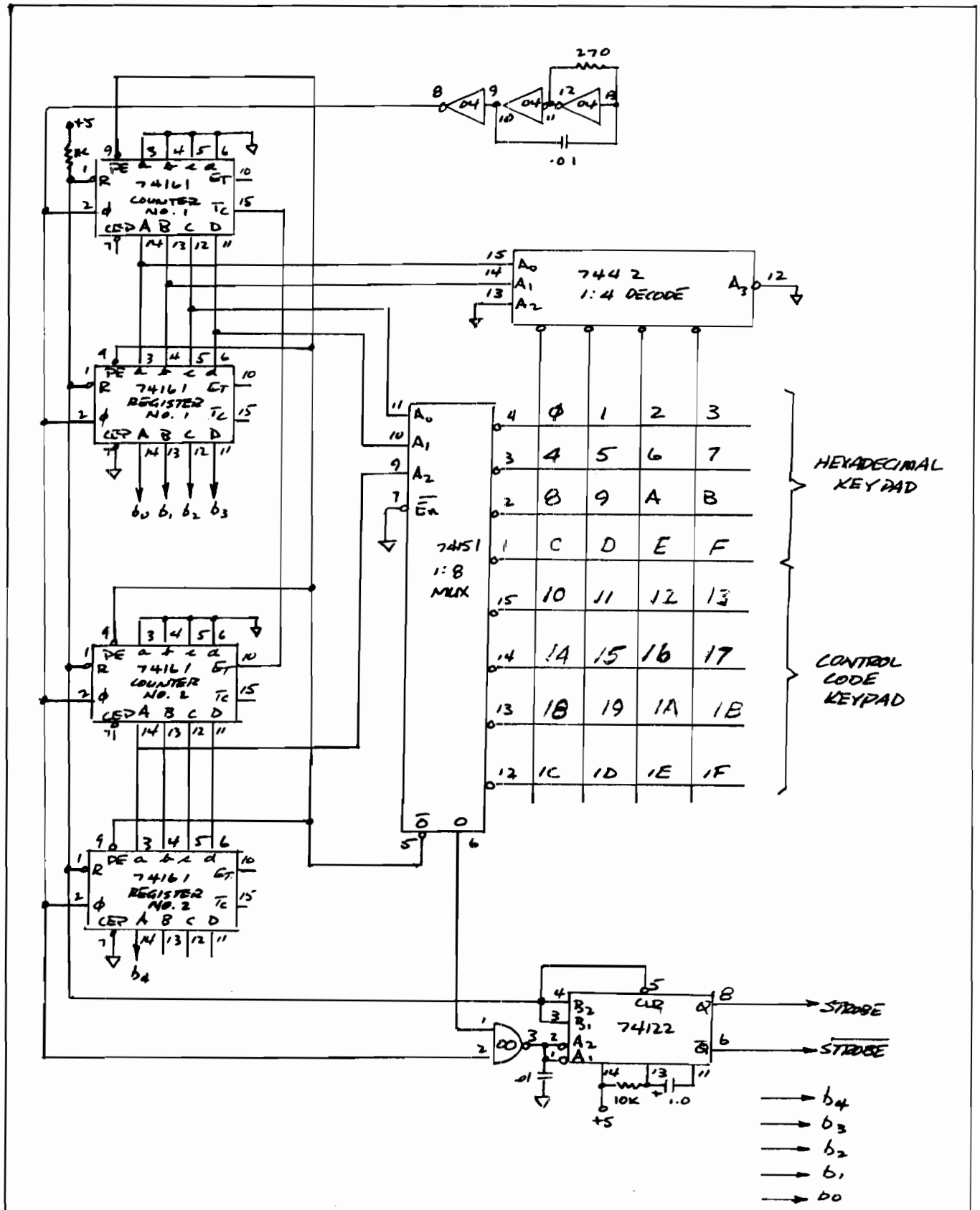


Figure 5. **32 key matrix encoder.**

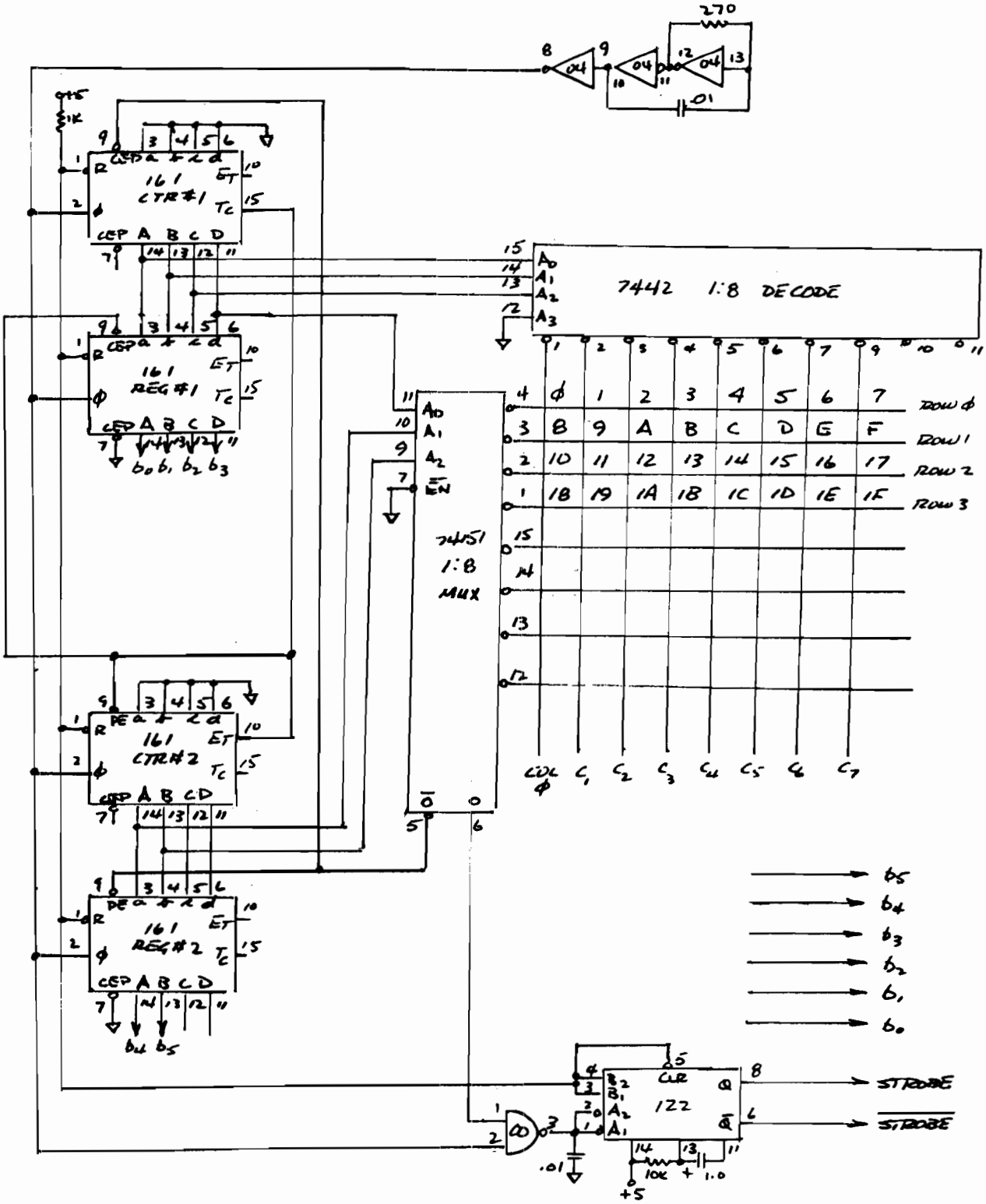


Figure 6. **Encoding up to 64 keys on an 8 × 8 matrix.**

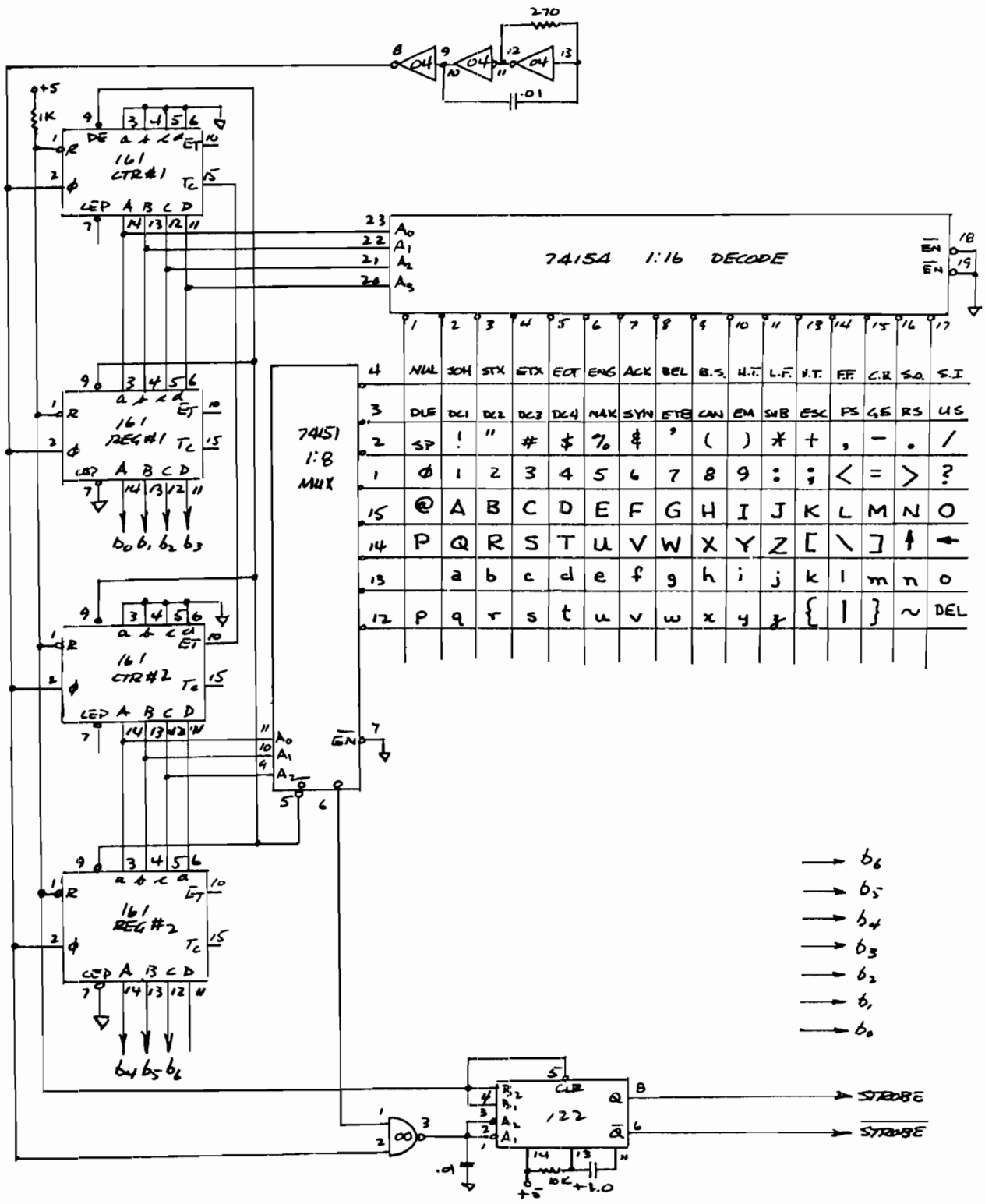


Figure 7. **128 key matrix encoder.**

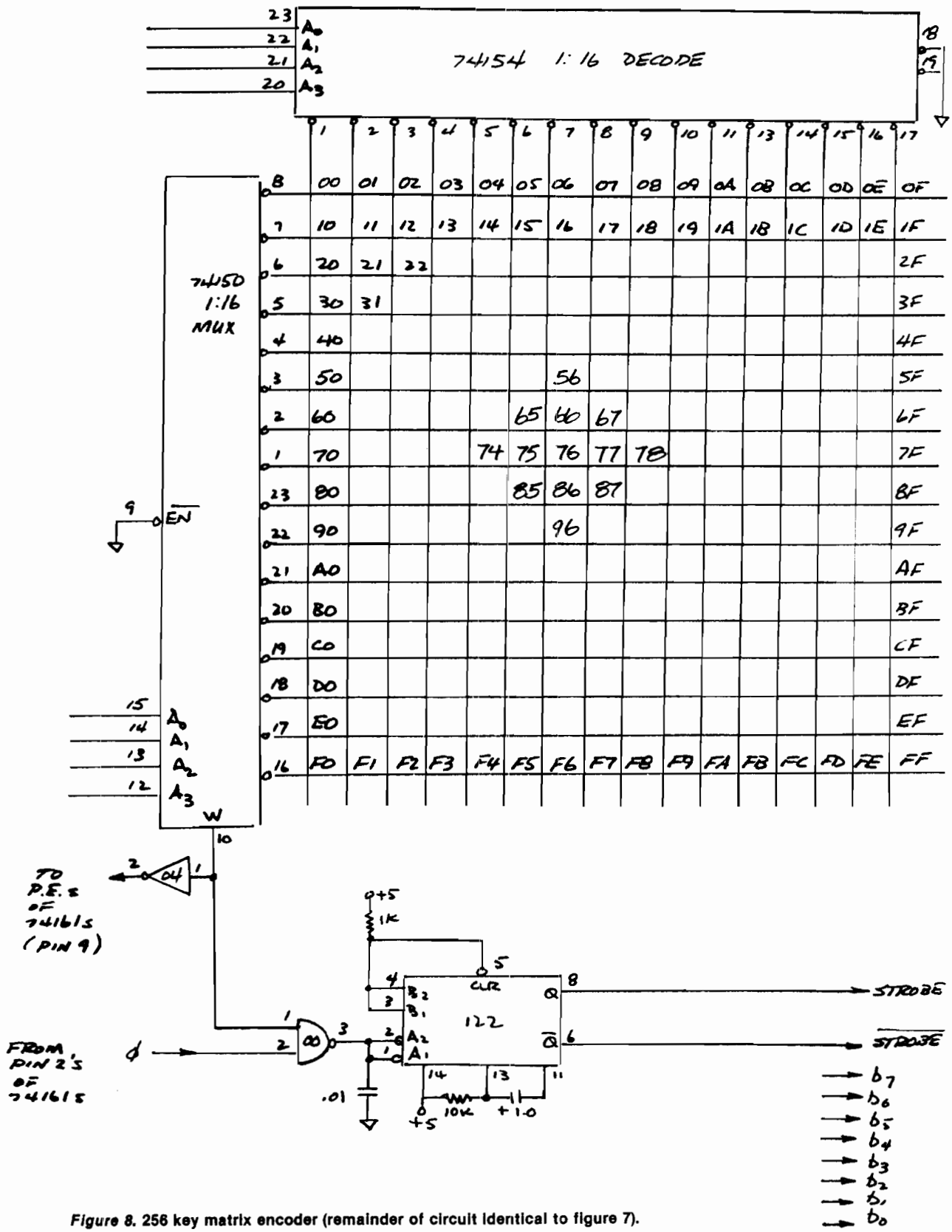


Figure 8. 256 key matrix encoder (remainder of circuit identical to figure 7).

Expansion

To encode 32 keys in a matrix requires only 4 additional lines from the keyswitches to the encoder circuitry. An additional 74161 counter stage, and an additional 74161 register stage must be added to the basic circuit of figure 4. I will first place the matrix in a 4 × 8 configuration. This circuit is shown in figure 5. The top keyswitch matrix will generate the hexadecimal code from 00 to 0F. The lower keyswitch matrix will generate hex codes from 10 to 1F. The keyswitches can also be arranged in an 8 × 4 matrix to encode the keyswitches. This circuit is shown in figure 6. Figure 6 also reveals that the encoding capability of the 4-74161's and 1:8 decoder and 1:8 MUX is only partially used. Sixty-four switches in an 8 × 8 matrix requiring 16 wires from switches to encoder circuitry can be encoded by the circuitry of figure 6.

The basic circuit is very flexible. Let us next encode a keyboard that has over 100 keys. We must change the decoder to a 74154 1:16 decoder, and place the keyswitches in a 16 × 8 matrix. In the configuration of figure 7, 128 switches can be encoded with the basic circuitry used earlier. The ASCII character set has been placed in position in the matrix to illustrate how the circuitry can be used to encode an ASCII keyboard. The builder will find, however, that he must make special provisions for the ASCII keyboard Shift and Control keys.

The circuit of figure 7 will encode up to 128 keys placed in a 16 × 4,5,6, 7 or 8 matrix and therefore can be used to encode almost any surplus keyboard. If the key in the matrix generates a different code than desired, it is a fairly simple matter to use the power of the computer itself to convert the code to ASCII, EBCDIC, Baudot (Murray), or whatever other code the builder may desire.

Thirty-three lines were used earlier to encode 32 discrete keyswitches. If the decoder is a 75154 and the MUX is a 74150, the 32 lines from the keyswitch matrix to the encoder will encode 256 switches (if you can find that many to encode!). The matrix is a 16 × 16, and the encoder capability is twice as much as the full ASCII character set. Extra keypads, cursor-controls, computer controls, and so on, can be encoded and the hexadecimal codes generated run from 00 to FF. figure 8 shows this circuit.

Computer Encoding

Keyboards may be encoded with a computer as well as with a hardware encoder. In this case we usually find ourselves restricted to 64 keys of encoding. This is caused by the 8-bit operation of our machines. We can use a PIA (Peripheral Interface Adapter) chip and tell 8 of the port lines they are outputs, and 8 more that they are inputs. With software, we then essentially duplicate the decoder function and drive each of the output lines low one-at-a-time. These correspond to the column lines of the hardware encoder. The keyswitches are placed in an 8 × 8 matrix so that the input lines (row lines) feed the input port. The keyswitch code generated in this fashion does not resemble hexadecimal in any sequence that you can dream up, but each key in the matrix will generate a discrete code. These discrete codes are then converted by software to the desired code. Figure 9 gives the basic circuit.

A full ASCII keyboard does not have 128 keys. The keys are made to do "double duty" with the Shift key. The Shift key on an 8 × 8 computer encoded matrix can be used to set one bit high and essentially double the number of codes generated. Special circuitry and software must be employed to handle the ASCII keyboard Control key. Thus, although the basic software encoder seems only capable of encoding either upper or lower case characters, it can, with some special circuitry encode almost all of the full 128 characters of the ASCII character set. We will have to give up a few of the ASCII codes, since two of the keys (Shift and Control) will need to perform special functions.

The basic problem of the computer encoded keyboard is that keyboards do not easily fit into a square matrix. Keyboards are rectangular and are usually 16 keys (or less) wide by 5 or 6 rows high. Consequently, if the foil traces must be cut to get the key-

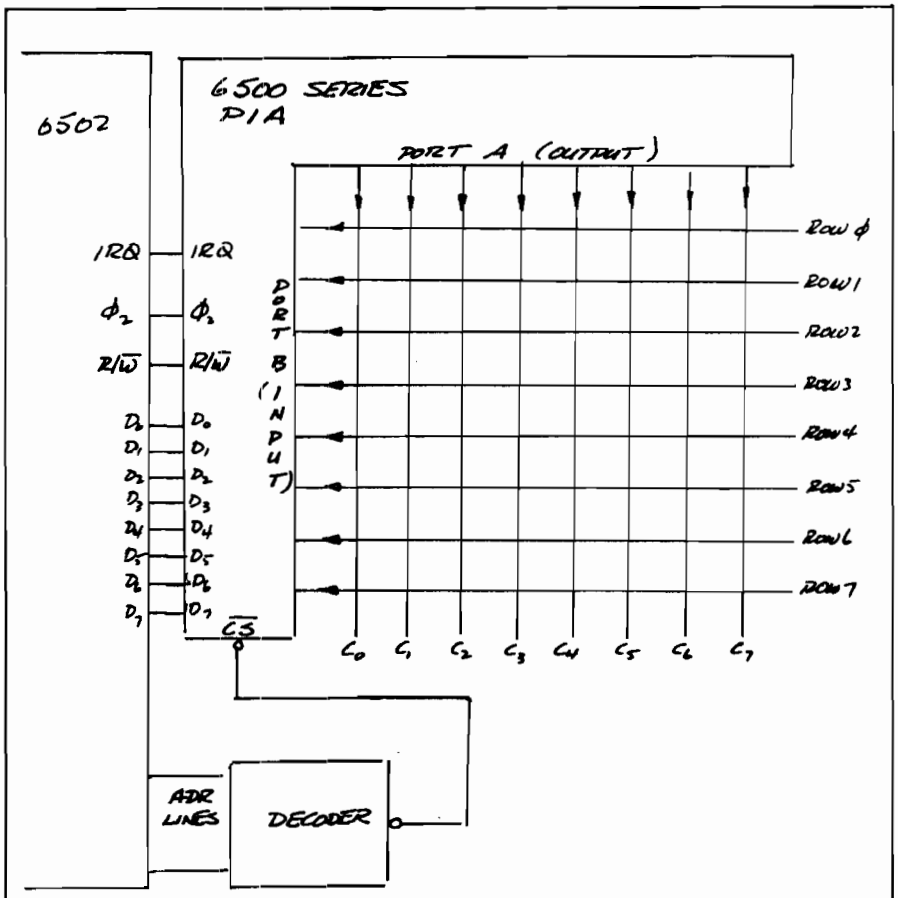


***Figure 9.* Basic circuit for matrix encoding using PIA chip.**

switches into a matrix, the hardware 16 × 5 or 6 encoder will usually prove simpler to rewire than figuring out how to square things up and get the keys into an 8 × 8 matrix.

The Input Port

Once the keyboard has been encoded in hardware the builder must find a way to get its codes into the machine. One very definite advantage of the software encoded keyboard using a PIA chip is that this detail need not concern the builder.

To get into the machine, additional hardware is required. If a serial port is desired, this is usually handled with a UART. A parallel port is very easy to construct, and usually much less expensive to get into operation than the UART. (Figure 10 gives the circuitry.) The port is placed wherever the builder desires, with a decoder. The circuit of figure 10 will place the port at 7FF7, or about half-way through the address range. The keyboard code, which must be latched, is transferred to the accumulator with a LDA, Absolute Mode at address 7FF7. If the software is interrupt-driven, the IRQ latch may also be reset with this command. If the software is NMI-driven, then resetting of the IRQ latch is unnecessary. A STA, page zero or STA, Absolute will then place the keyboard code in memory and a Return From Interrupt can follow. The code can then be processed with a subroutine to convert it to whatever code you desire.

A bonus of the circuit of figure 10 is that not only one port is provided for, but 8 active lows are available from 7FF0-7FF7. This means that you can have 8 input or output ports at these locations. Or the signals generated by either reading or writing these memory locations can be used for handshakes or software generated turn-on or turn-off signals to the outside world. By placing a 1:16 decoder (74154) here instead of a 1:8 or 1:10 decoder, 16 ports, handshakes, etc. can be obtained.

Summary

The one basic circuit is quite flexible. It can be used to hardware encode almost any keyboard. Placing the keyboard in a matrix reduces the number or interconnecting wires needed between keyswitches and encoder circuitry. Since only one fundamental circuit was given here, and there are many, many ways to do the job, this article has, in reality, barely "scratched the surface" of encoding keyswitches.

George Young has been into computer electronics and programming for only three years. He has designed and built his own 6502-based homebrew computer in nine Phases, and this HB includes everything but floppies. Articles for publication are prepared on the text editing portion of this 6502 homebrew, and printing is via an interfaced Selectric I/O Writer.

The author's main interest is education He will teach *anything* to *anybody* who will take the time to listen to him or ask the questions and listen to the answers. He enjoys writing since it is an extension of his teaching (and enables him to have a larger number of students in his 'classroom').

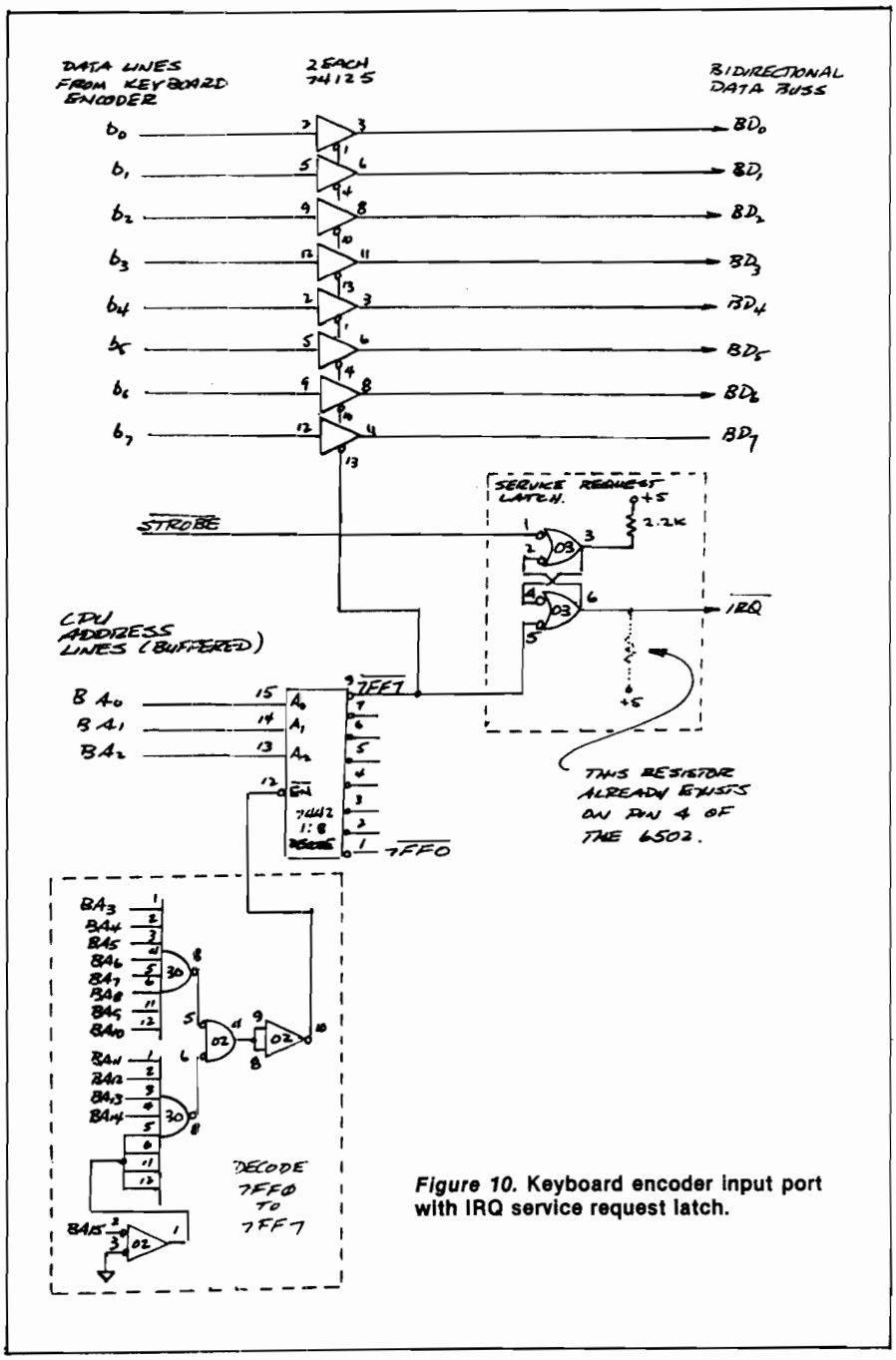
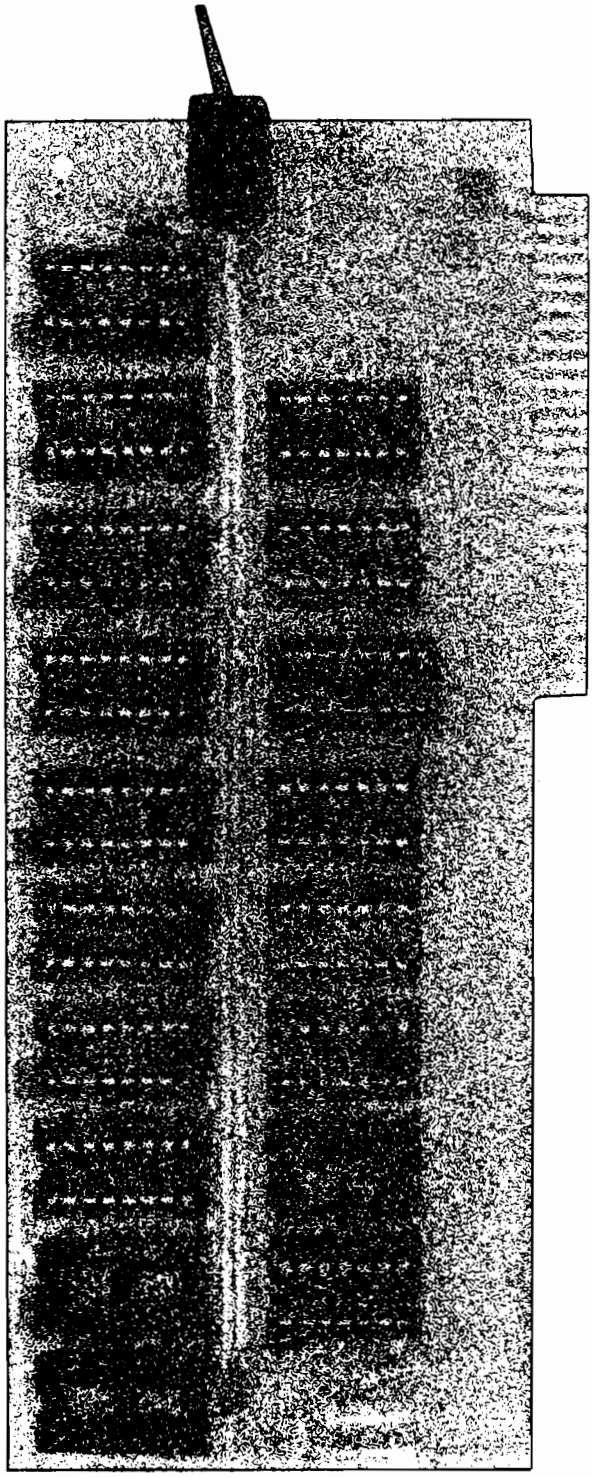


Figure 10. Keyboard encoder input port with IRQ service request latch.



16K RAM Expansion Board for the Apple II*

$195.00

- expands your 48K Apple to 64K of programmable memory
- works with Microsoft Z-80 card, Visicalc, LISA ver 2.0 and other software
- eliminates the need for an Applesoft* or Integer Basic ROM Card
- switch selection of RAM or mother board ROM language
- includes installation and use manual
- fully assembled and tested

Visa and MasterCard accepted
Shipping and handling will be added unless the order is accompanied by a check or money order
N.C. residents add 4% sales tax

*Apple II and Applesoft are trademarks of Apple Computer, Inc.

ANDROMEDA
INCORPORATED**
P.O. Box 19144
Greensboro, NC 27410
(919) 852-1482

**Formerly Andromeda Computer Systems

Computer Camp

This summer, youngsters can again sign up for an overnight camp in Moodus, Connecticut, where the main activity will be COMPUTERS. This unique recreational and educational experience is directed by Dr. Michael Zabinski, Professor at Fairfield University. It is the original summer, overnight, computer camp currently offered in the USA.

The 1981 National Computer Camp will feature two action-packed one-week sessions: first session July 19-24; 2nd session July 26-31. The campers, ages 10-17, will enjoy small group instruction and mini and microcomputers for ample "hands-on" experience. Dr. Zabinski will be assisted by elementary and secondary school teachers.

The camp is for kids of all levels of experience, including no experience whatsoever. In addition to computers, the campers will enjoy the recreational facilities of the Grand View Lodge, which include swimming and volley ball.

For further information, contact Micahel Zabinski, Ph.D., at (203) 795-9069, or write to:

Computer Camp
Grand View Lodge, Box 22
Moodus, Connecticut 06469.

Apple Disk Fixer

APPLE II
32K, DISK **13 OR 16 SECTOR**

If you care enough to back up critical programs and files, Disk Fixer™ will give additional peace of mind. This powerful utility for experienced Apple users is a tool kit for manipulating, repairing, and protecting all data on disk.

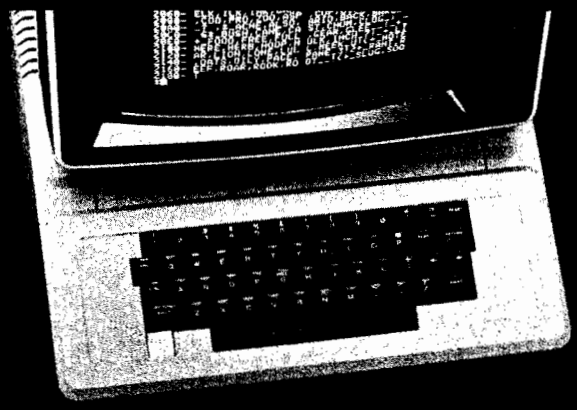
Use the high-speed full screen editor to examine and easily change any portion of a disk, correct space usage within files, and save money by locking out bad tracks on disks. Directories are alphabetized, if you choose.

The display and search capabilities show where specific HEX or ASCII data is located and you can modify any data including binary files. DOS 3.2, DOS 3.3 & LANGUAGE SYSTEM DISK

Written by Jeffrey P. Garbers
©1980 The Image Producers, Inc., All Rights Reserved

IMAGE COMPUTER PRODUCTS™
615 Academy Drive
Northbrook, IL 60062
312/564-5060

Apple Monitor Extender



APPLE II 16K, CASSETTE

This utility program works in complete harmony with the Apple monitor to extend your computer's capability and help you use the full power of machine language programming.

Screen display shows memory in HEX, ASCII or BINARY. Move data anywhere in memory without regard to direction or overlapping and read or write any sector on disk. Insertions may be in HEX or ASCII so you can easily format high speed text displays without conversions.

Study, modify or disassemble any program, complete with labels. Several programs may be combined, and the entire disassembled text file stored on disk/tape for later assembly.

The slow listing feature steps through listings with ease.

Copyright 1980 Glenn R. Sogge, All Rights Reserved.

IMAGE COMPUTER PRODUCTS™
615 Academy Drive
Northbrook, IL 60062
312/564-5060

MUSICAL COMPUTER I AND II

Learn How to Read Music!

Written by an M.A. educator with over 20 years of music experience. This two program cassette provides an *alternative* to music education.

- Treble & Bass Note Reading
- Telling Time
- Notes and Rests
- Sharps and Flats
- Signs and Symbols
- Tempo Definitions

$34.95 plus $1.00 for postage and handling
Check or Money Order Please
(Mi. residents add 4% sales tax)

Check: ☐ Apple II 32K ☐ TRS-80 Level II 16 K

Name: .....

Address: .....

.....

COMPUTER
APPLICATIONS
TOMORROW

P.O. Box 605
Birmingham
Mi. 48012

A Better Apple Search/Change

This improved version of the SEARCH/CHANGE program removes the length restriction on the CHANGE function.

J.D. Childress
5108 Springlake Way
Baltimore, Maryland 21212

Lately I've made much use of the SEARCH/CHANGE program published last January in MICRO [20:55] and have become quite impatient with its shortcomings. I first made some minor modifications but then decided the real need was to remove the length restriction on the CHANGE function. The complete program, with this modification, is given in the program listing. The discussion following assumes familiarity with the earlier paper.

An urgent warning is issued here. This program is like the girl with the curl: When it's good, it's very, very good but when it's bad, it's horrid. If it misfires, your program likely will be lost. The wise will always keep a current backup copy.

Minor Modifications

For those not wanting to do much retyping or needing only the range control feature [references here are to the line numbers in the SEARCH/CHANGE listing of the January paper]: delete lines 63100, 63280, 63300, and 63310. These were written and kept in an abundance of caution. Change 63300 to 63320 after the GOSUB in line 63150. Add a line, say 63035, for input of BL, the beginning line for the change, and EL, the ending line; then insert OR LM < BL OR LM > EL in line 63350. The foregoing produces a shortening of the program, along with a capability for changing only a specified part of the program.

Another warning is in order here. The feature of listing the found lines works with a search for the LIST in line 63270 backwards from the end of program memory. If anything with a LIST in it is placed in higher line numbers, this search will be fouled up. And again note: somehow leading zeros in line 63270 can get removed; I suggest that a few, say 5, colons be inserted before INPUT.

Overview of Major Modification

Insertion of a CHANGE item longer or shorter than the SEARCH item requires that spaces either be added or deleted. This is accomplished by a shift of the program in memory and corresponding changes of all the next-address pointers. Needless to say, the part of memory space being used by SEARCH/CHANGE must not be jiggled, else its operation will be clobbered. So that the SEARCH/CHANGE program can remain fixed in memory and all the Applesoft operational pointers functional, spacers—colons—are added in line 62999.

Memory Move

The heart of this modification is the memory move call from Applesoft. [See lines 63360 and 63370 in the listing of this paper.] The memory move call given in *CONTACT 5, 5* [June 79] works only for integer BASIC; a call to the Apple HOTLINE produced the information that the move call had to be routed through a short machine language routine. The routine supplied by Apple is the following: POKE 768,160: POKE 769,0: POKE 770,76: POKE 771,44: POKE 772,254. The corresponding call is then CALL 768. This location is $300-$304. I use that space for my SLOW LIST utility, so changed to location $342-$346 with no ill effects. [See line 63010.]

One block of memory cannot be moved into a second block overlapped by the first. This is because one byte would be moved into another before that byte's content had been read. Thus a two-step procedure is required. Line 63360 moves the memory block to the top of memory just below HIMEM and line 63370 moves it back to the desired end location.

Next-Address Pointers

Recall how Applesoft stores BASIC in memory. The end of each line is indicated by a zero byte. The next two bytes contains a pointer, low byte first, to the next line's first byte, the low byte of its next-address pointer. When a branch is executed, the program skips along these pointers from the first until the indicated address is found. If any next-address pointer points to a wrong address, all gang a-gley. So until all these pointers are put aright, the program being searched and changed is simply hidden from the operating program. Line 63160 POKEs the next-address pointer for line 63000 into the pointer location of line 1; after the dust settles, line 63230 restores the original pointer.

Other Matters

A search is made from the end of program memory to find two things: the location of the LIST in line 63310, and the location of the beginning of line 62999. This search is done by line 63020. Two numbers, 540 and 1730, are set for SEARCH/CHANGE exactly as written [lines 63000 and following] in the listing. The first number causes a skip from the end of the program to the immediate neighborhood of LIST, the second, a skip from LIST to near the end of line 62999. One effect of this search is a delay after the return following RUN 63000. A too-short delay should alert the user that line 62999 might not have enough colons for substantial changes. If the colons are depleted, line 63350 halts the change operation and prints a message to that effect.

```
62999  END ::::::::::::::::::::::
     ::::::::::::::::::
63000  DIM SK(100),NT(100),L(100)
     :START = 256 *  PEEK (104) +
      PEEK (103):FINI = 256 *  PEEK
     (106) +  PEEK (105)
63010 HM = 256 *  PEEK (116) +  PEEK
     (115): POKE 834,160: POKE 83
     5,0: POKE 836,76: POKE 837,4
     4: POKE 838,254
63020  FOR WW = FINI - 540 TO STA
     RT STEP  - 1: IF X = 0 AND  PEEK
     (WW) = 188 THEN W = WW:X = 1
     :WW = WW - 1730
63030  IF 256 *  PEEK (WW + 1) +
      PEEK (WW) <  > 62999 THEN  NEXT

63040 NL = 256 *  PEEK (WW - 1) +
      PEEK (WW - 2):CO = NL - WW -
     10: HIMEM: HM - WW - 100
63050  IF 256 *  PEEK (START + 3)
      +  PEEK (START + 2) <  > 1 THEN
      PRINT "YOU MUST ENTER YOUR
     SEARCH ITEM AS LINE": PRINT
     "1 BEFORE YOU RUN 63000.": HIMEM:
     HM: END
63060  LIST 1,2: PRINT : PRINT "P
     LEASE VERIFY IF THE COMPUTER
      TAKES": PRINT "THIS AS YOU
     INTENDED. DO YOU WANT": INPUT
     "TO CONTINUE (YES OR NO)? ";
     Y$: PRINT : IF Y$ <  > "YES"
      THEN  HIMEM: HM: END
63070  PRINT "THE CHANGE ENTERED
     WILL BE MADE IN THE": PRINT
     "RANGE OF LINES CHOSEN. ENTE
     R": INPUT "    THE BEGINNING
      LINE   ";BL: INPUT "    THE
      ENDING LINE      ";EL: PRINT
63080 NF = 256 *  PEEK (START + 1
     ) +  PEEK (START)
63090  FOR I = 0 TO 255:SK(I) =  PEEK
     (START + 4 + I): IF SK(I) <
      > 0 THEN  NEXT
63100 N = I - 1
63110  IF SK(0) = 34 THEN  FOR I =
     1 TO N:SK(I - 1) = SK(I): NEXT
     :N = N - 1:SQ = 1
63120 M = START + N + 6 + SQ:INC =
     0:CH = 0: IF 256 *  PEEK (M +
     3) +  PEEK (M + 2) <  > 2 THEN
     CH = 1: GOTO 63170
63130  FOR I = 0 TO 255:NT(I) =  PEEK
     (M + 4 + I): IF NT(I) <  > 0
      THEN  NEXT
63140 NN = I - 1:ADD = NN - N:M =
     M + NN + 6:WW = WW - ADD(ADD
      < 0) + 8
63150  IF NT(0) = 34 THEN  FOR I =
     1 TO NN:NT(I - 1) = NT(I): NEXT
     :NN = NN - 1:ADD = ADD - 1
63160 X =  INT (NL / 256):Y = NL -
     256 * X: POKE START + 1,X: POKE
     START,Y
63170 LM = 256 *  PEEK (M + 3) +
      PEEK (M + 2):NA = 256 *  PEEK
     (M + 1) +  PEEK (M): IF LM >
      = 62999 THEN 63230
63180  FOR I = M + 4 TO M + 255: IF
      PEEK (I) <  > 0 AND  PEEK (
     I) <  > SK(0) AND  PEEK (I) <
      > 34 THEN  NEXT
63190  IF  PEEK (I) = 34 THEN SQ =
     SQ + 1 - 2 * (SQ = 1)
63200  IF  PEEK (I) = SK(0) AND S
     Q <  > 1 THEN  GOSUB 63330
63210  IF  PEEK (I) = 0 THEN NA =
     NA + INC:X =  INT (NA / 256)
     :Y = NA - 256 * X: POKE M +
     1,X: POKE M,Y:M = I + 1: GOTO
     63170
63220  NEXT
63230 X =  INT (NF / 256):Y = NF -
     256 * X: POKE START + 1,X: POKE
     START,Y
63240  PRINT : PRINT : PRINT "THE
      ITEM": PRINT "  ";: LIST 1:
      PRINT : PRINT "IS FOUND IN
     THE FOLLOWING LINES:": PRINT
     : IF L(1) = 0 THEN  PRINT "
          NONE.": HIMEM: HM: END
63250  FOR I = 1 TO K: PRINT L(I)
     ,: NEXT : PRINT
63260  PRINT : INPUT " DO YOU WAN
     T THESE LINES LISTED (YES OR
      NO)? ";Y$: IF Y$ = "NO" THEN
      HIMEM: HM: END
63270  PRINT : PRINT "THERE WILL
     BE A WAIT AFTER EACH LINE": PRINT
     "UNTIL YOU HIT RETURN TO CON
     TINUE.": PRINT
63280  FOR I = 1 TO K: IF L(I) =
     L(I - 1) THEN 63320
63290 L$ = "0000" +  STR$ (L(I)):
     L$ =  RIGHT$ (L$,5)
63300  FOR J = 1 TO 5: POKE W + J
     ,48 +  VAL ( MID$ (L$,J,1)):
      NEXT
63310  LIST 00050::::::::: INPUT
     "";Y$
63320  NEXT : HIMEM: HM: END
63330  FOR J = 0 TO N: IF  PEEK (
     I + J) <  > SK(J) THEN  RETURN

63340  NEXT
63350  IF CO - INC < ADD AND CO% =
     0 THEN CH = 1:CO% = 1: PRINT
     "THE SUPPLY OF COLONS IN LIN
     E 62999 IS": PRINT "DEPLETED
     .  THE CHANGE HAS BEEN MADE"
     : PRINT "THROUGH PART OR ALL
      OF LINE ";LM: PRINT
```

```
63360 K = K + 1:L(K) = LM: IF CH <
      > 0 OR LM < BL OR LM > EL THEN
      RETURN
63370 ZS = I + N:X =  INT (ZS / 2
      56):Y = ZS - 256 * X: POKE 6
      1,X: POKE 60,Y:X =  INT (WW /
      256):Y = WW - 256 * X: POKE
      63,X: POKE 62,Y:ZH = HM - 10
      0 - WW + ZS:X =  INT (ZH / 2
      56):Y = ZH - 256 * X: POKE 6
      7,X: POKE 66,Y: CALL 834
63380  POKE 61,X: POKE 60,Y:X =  INT
      ((HM - 100) / 256):Y = HM -
      100 - 256 * X: POKE 63,X: POKE
      62,Y:ZS = I + NN:X =  INT (Z
      S / 256):Y = ZS - 256 * X: POKE
      67,X: POKE 66,Y: CALL 834
63390 WW = WW + ADD:INC = INC + A
      DD
63400  FOR J = 0 TO NN: POKE I +
      J,NT(J): NEXT :I = I + NN
63410  RETURN
```

The perceptive reader will note a number of small numbers in various lines. These are finagle factors adjusted (but probably not optimized) to make the program run. For example, the 10 in line 63040 prevents the appearance of multiple line 62999's for a change item shorter than the search item.

I have also changed the choices for search and the way they work. Experience has shown no need to search both inside and outside strings at the same time; it's an either/or situation. To eliminate a needless question/answer routine, the program now works as follows: the SEARCH/CHANGE is made outside strings only, unless the first character of line 1 is the quotation mark. In that case, the SEARCH/CHANGE is made inside strings only. For example,

```
1 TOTAL
```

would search for TOTAL outside strings but

```
1 "TOTAL
```

would search for TOTAL inside strings.

A quotation mark can be used with line 2 in a similar way to sneak "forbidden" words past the interpreter. This should be used with care in changes outside strings; the interpreter has a way of exacting its revenge on sneaky things.

Operation

Except for the search mode selection change, operation of the better SEARCH/CHANGE is essentially the same as the original. Append SEARCH/CHANGE to the program to be searched. Enter the search item as line 1 and the replacement item as line 2. Note that anything that will list as line 1 (or line 2) can be entered. Execute with a RUN 63000.

As mentioned earlier, things can go wrong. If the worst happens, try entering a new line, or deleting a line, or both. That sometimes will save part, or almost all of the program. A sensible precaution is to check line 62999 often and keep it well-stocked with colons.

Another good idea is to use a SLOW LIST utility with SEARCH/CHANGE. (I recommend the one supplied with S-C ASSEMBLER II.) Then if a LIST command produces endless junk, the listing can be aborted without the additional hazard of a RESET.

Another idea is to know your HIMEM. If something goes wrong, it's possible for the HIMEM setting to be ratcheted down to a low value.

A problem I encountered on occasion and, I trust, eliminated, is the DOS 3.2 renumbering program (which I keep in memory while programming) or the DOS buffers being clobbered, probably by an occasional encroachment into the space above HIMEM.

MICRO

APPLES NEED ROMS FOR ROMPLUS

Now you can realize the full power and usefulness of those now empty sockets on your **ROMPLUS** board.
If you have **MH's** 'Keyboard Filter' then you know how easy, quick, and convenient it is to activate and run the 'Filter' program. Can you visualize having other programs, on a ROM, plugged into those empty sockets on your Romplus board and activating them with the same ease and convenience? If you can visualize it then you can realize it!

APPLESOFT RENUMBER/ MERGE IN ROM

This invaluable program, made famous by Apple Computer, needs no explanation for the serious programmer. When activated, RENUMBER/MERGE IN ROM will not distrub any part of a program in memory. Now you can take advantage of this powerful utility without the inconvenience associated with getting it up and running prior to a programming session. Requires 48K, with or without Disk II **$35.95**

DISK COPY/DISK SPACE IN ROM

A resident disk copy program will make it easier, more convenient, and will encourage you to make back-ups on your valuable diskettes.

- Copies diskettes, from either SINGLE or DUAL drive, single or dual controller, 13 or 16 sector and with or without VTOC.
- Options - Gross copy, Active Sectors only copy, more than one copy, DOS overwrite for faster copy time, auto matic boot of copy disk, free space on disk in sectors and kilobytes.
- Init and volume number change are selectable.
- All disk error reporting routines with option to abort.

Requires a minimum of 32K **$35.95**

BASICS IN ROM

Eliminates the added steps of locating, installing, running and rebooting with a BASICS diskette when the need arises to go from 16 to 13 sector disks. Just put a 13 sector disk in your disk drive and address the Basics ROM. The Basis ROM is addressable on coldstart (without AutoStart ROM) or warmstart at any time and will save wear and tear on your disk drives. Requires a minimum of 16K **$35.95**

'YOUR' PROGRAM LINE EDITOR (PLE) IN ROM

If you have this invaluable program in software, send us a Diskette which has only your predefined PLE greeting program on it (Int or FP version). You will get back your diskette and a ROM chip that has 'YOUR' PLE burned into it. The ROM will also have a routine to erase 'YOUR' PLE. 'YOUR' PLE will install and initialize itself in memory and will not affect any Applesoft programs in memory. Please specify if your disk is 13 or 16 sector and be sure no program steps have been added to the original PLE. Allow 2-3 weeks delivery. Only one chip per disk please. Requires 32K or 48K, 3.2 or 3.3 **$45.95**

All Roms, after they have been activiated, will shut down your Romplus board and reconnect your Apple to the I/O hooks of DOS or Basic. All ROM chips can be returned for updating to later versions for a nominal charge. All chips will be shipped UPS.

SOFT CTRL SYSTEMS
BOX 599
WEST MILFORD, N.J. 07480

BASIC THAT SCREEMS

Is the sedate pace of your OSI BASIC taking the fun out of your programming?

Then turn your system on to FBASIC.

Now you can compile your programs with FBASIC and take full advantage of your computers potential.

FBASIC is fast. Not just 5 or 10 times as fast as OSI BASIC. FBASIC is OVER ONE-HUNDRED TIMES FASTER! Allowing you to do unheard of things in BASIC. Things that used to require assembly-language. With no need to learn a new language.

The secret to this incredible speed is that FBASIC produces native 6502 machine code. With no run-time interpreter to get in the way of all-out machine performance.

FBASIC is good for any application from wordprocessors to video games. Whatever tickles your fancy. FBASIC accepts standard BASIC source files, and produces executable disk-based object files. It includes many of the features you have always wanted: hex constants, convenient machine language calls, optional user selection of array locations, and more.

As an example of its power and performance; FBASIC was first written in OSI BASIC, which took 6 hours to compile itself. With many features added since then, it now compiles itself in a little over 4 minutes!

So let that pent-up performance out. Find out what your machine is really capable of. Feed it some FBASIC and stand back!

FBASIC runs under OS-65D and requires 48K.

Available on 8" diskette for $150.

Please include $5 for shipping & handling.

PEGASUS SOFTWARE
P.O. Box 10014, Dept. M-2,
Honolulu, Hawaii 96816

OHIO SCIENTIFIC

THE ALL NEW
SERIES 2 COMPUTERS
ARE ON DISPLAY

WE STOCK THE COMPLETE
LINE OF AARDVARK

FOR SAN FRANCISCO AREA
READERS WE HAVE A
USERS GROUP MEETING THE
FIRST SUNDAY OF EVERY MONTH

SEND $1.00 FOR CATALOGS

SUNSET ELECTRONICS
2254 TARAVAL ST.
SAN FRANCISCO, CA 94116
(415) 665-8330
OPEN 7 DAYS

American Data

The world's largest distributor of
Ohio Scientific Microcomputers

ANNOUNCES

The availability of the new Ohio C2-D and C3-D (10 MB 8" hard disk units) and the C1P Series 2 converted by American Data for European power.

American Data is the exclusive distributor of OSI microcomputers to Europe. Dealer inquiries invited: Contact David O'Brien (301) 840-9540. Telex: 64405 (USA).

Vectors and the Challenger 1P

Vector is one more computer buzzword usually found in conjunction with confusion. This article will try to clear up what vectors are and will show how to use them effectively.

Mike Bassman
39-65 52nd Street
Woodside, New York 11377

Computers have subroutines for every command and for every other necessary function. Computers also need places in memory to look up the address of a particular subroutine. A vector is a place in memory in which the computer finds the address of a subroutine. The vector will consist of two bytes containing any address from $0000 to $FFFF in low-byte, high-byte format.

I'll try to clear this up with an example. Let's say you type in a SAVE command. The C1P must know where to go to find the SAVE subroutine. The C1P looks at the SAVE vector, which is at $220 and $221 (544 and 545 decimal), and in it finds $96 and $FF, which is $FF96, the address of the SAVE routine.

What good are vectors? They are used if you want to add to BASIC or change it or any of its commands. Vectors have been used to create shorthand (see MICRO 24:25), do a true backspace, and ensure program security. Let's take program security as an example. A good way to prevent copying of tapes is to disable the SAVE command. We want the SAVE command to coldstart BASIC if someone tries to save a program. The SAVE vector must be changed from pointing at the SAVE routine at $FF96 to the coldstart address at $BD11. Therefore, the SAVE vector at 544 and 545 must point at $BD11. These two poke commands will take care of it:

```
POKE 544,17
POKE 545,189
```

Now, if you type SAVE, the C1P will respond with "MEMORY SIZE?". Although the C1P has vectors for every command, a large portion of them are in ROM rather than in RAM. Since only RAM and not ROM can be changed, only those vectors residing in RAM can be used. Vectors in RAM, their addresses, and the address that they point at initially are listed below.

Vector	Address		Initial Value	
	Hex	Decimal	Hex	Decimal
SAVE	220,221	544,545	FF96	65430
LOAD	21E,21F	542,543	FF8B	65419
CTRL C	21C,21D	540,541	FF9B	65435
OUTPUT	21A,21B	538,539	FF69	65385
INPUT	218,219	536,537	FFBA	65466
WARM START	001,002	001,002	A274	41588

The first two vectors, the SAVE vector and the LOAD vector are used whenever the SAVE and LOAD commands are executed. The Control-C vector is somewhat more subtle. For every line of BASIC executed, BASIC checks the keyboard for a Control-C. If one has been typed, program execution ends. This vector is the one that is used for every line executed. Therefore, we can use this vector, if our function is one that would be executed once for every line, just as we would use a program tracer. The output vector is used each time BASIC wants to type a character. The input vector is used each time BASIC wants to input a character.

Let's try a more involved example. This time, we'll change the cursor from the underline to, say, a tank character. The program will require use of the input vector, because we will have to change the cursor each time a character is inputted. The program will be in the free part of page 2, starting at $222 (546 decimal). BASIC stores the cursor location at $200 (513 decimal). This is used as an index from screen location $D300 (54016 decimal). Since BASIC resets the input vector after carriage return is hit, we will set the input vector to $222 after every character. The program follows.

Using these techniques, Ohio Scientific C1P users can customize BASIC to their convenience.

```
                    BASSMAN:VECTORS AND THE CHALLENGER 1P
0222                        ORG   $0222
0222 48                     PHA           SAVE ACCUMULATOR
0223 8A                     TXA           PUT X IN A
0224 48                     PHA            AND SAVE
0225 AE 00 02               LDX   $0200   GET CURSOR LOCATION
0228 A9 FA                  LDA   #$FA    GET TANK CHARACTER
022A 9D 00 D3               STA   $D300,X STORE WITH INDEX
022D 68                     PLA           GET X FROM STACK
022E AA                     TAX           AND PUT INTO X
022F A9 22                  LDA   #$22    LOW BYTE--INPUT VECTOR
0231 8D 18 02               STA   $0218   AND SAVE
0234 A9 02                  LDA   #$02    HIGH BYTE--INPUT VECTOR
0236 8D 19 02               STA   $0219   AND SAVE
0239 68                     PLA           GET A
023A 4C BA FF               JMP   $FFBA   AND RETURN
```

MICRO

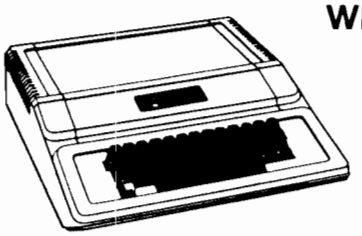
THE STAR MODEM

From Livermore Data Systems

RS232 MODEM	SALE $135
IEEE 488 MODEM	SALE $245
RS232 CCITT	$170
IEEE 488 CCITT	$280

STAR Modem is the price performance leader with a full 2 YEAR FACTORY WARRANTY.

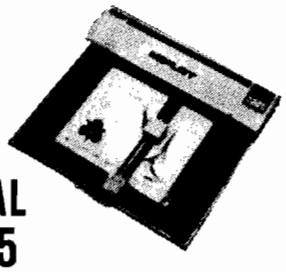
WE CARRY THE BLACK APPLE



BELL & HOWELL

apple computer inc.

MIPLOT Intelligent Plotter
by Watanabe Instruments (Digiplot)



SPECIAL $1095

- Incorporates all intelligent functions required for producing graphs and drawings including 8 vector and 4 character commands.
- Distance accuracy within 1%, repetition accuracy within .01 inches, programmable step size .004 inches, internal interpolation in .002 steps.
- Solid and broken line types can be specified.
- Character generator for letters, numerals and symbols. Characters can be enlarged and rotated in four orientations.
- Coordinate axes can be drawn by simply specifying the graduation interval and number of repetitions.
- Self test mode automatically draws complete test pattern.
- Printer mode outputs character data in 16 sizes with 4 orientations.
- Connection to any microcomputer using parallel 7-bit ASCII code.
- Can use simple cable to parallel port, or special interface to IEEE or other.
- Modular control circuit and mechanical construction.
- Uses any hard fiber-tip pen.
- Uses 11 x 17 paper.

Includes power supply, I/O connector, 2 pens, 50 sheets of paper, and complete manual.

EBS Business System for PET/CBM $795
Extremely comprehensive package for small business. Fully integrated inventory and accounts receivable system including invoices, packing slips, mail labels, statements, bank deposit slips, and 17 reports. Allows any of 10 standard letters to be merged with customer record info on either a selective or complete file basis. Demo disk and system description available for $3.00.

OZZ Data Base System for CBM 8032 $335
Flexible file handling and report writing package for such applications as: Inventory Control; Management Information; Mailing List; Scheduling; Medical Record Keeping; Accounting.

KMMM Pascal for PET $75
Subset of standard Pascal with true machine language translator for faster execution. 16K with tape or disk.

EARL for PET (disk file based) $65
Editor, Assembler, Relocater, Linker to generate relocatable object code.

fullFORTH+ for PET/CBM $65
A full-featured FORTH with extensions conforming to Forth Interest Group standards. Includes assembler, string processing capabilities, disk virtual memory multiple dimensioned arrays, floating point and integer processing.

6502	7.45	10/6.95	50/6.55	100/6.15
6502A	8.40	10/7.95	50/7.35	100/6.90
6520 PIA	5.15	10/4.90	50/4.45	100/4.15
6522 VIA	6.45	10/6.10	50/5.75	100/5.45
6532	7.90	10/7.40	50/7.00	100/6.60
2114-L450		3.95	20/3.75	100/3.45
2114-L300		4.45	20/4.25	100/3.95
2716 EPROM		13.45	5/12.75	10/11.85
2732 EPROM (TMS2532)				29.50
4116-200 ns RAM (NEC)				**8 for 35.00**
S-100 Wire Wrap $2.65				Solder Tail $2.15

CASSETTES – AGFA PE-611 PREMIUM
High output, low noise, 5 screw housing, labels.

C-10	10/5.65	50/25.00	100/48.00
C-30	10/7.30	50/34.00	100/66.00

All other lengths available. Write for price list.

ATARI 800 $777
All Atari Modules 20% OFF
SPECIAL–purchase ATARI 800, receive extra 8K memory FREE.
EDUCATIONAL PLAN - buy 2 ATARI Computers, receive 1 ATARI 400 FREE!

DISKS
(write for quantity prices)

SCOTCH (3M) 8"	10/3.10	50/2.85	100/2.75
SCOTCH (3M) 5"	10/2.95	50/2.80	100/2.70
Maxell 5" Double Dens	10/4.25	50/4.10	100/3.95
Maxell 8" Double Dens.	10/4.65	50/4.50	100/4.35
Verbatim 5"	10/2.40	50/2.35	100/2.30
(add .75 for plastic storage box for Verbatim 5")			
Verbatim 8" Dbl. Dens.	10/3.35	50/3.25	100/3.15
BASF 5"	10/2.60	20/2.50	100/2.40
BASF 8"	10/2.40	20/2.35	100/2.30
Diskette Storage Pages			10 for 3.95
Disk Library Cases		8" - 2.85	5" - 2.15

commodore

CBM-PET SPECIALS

FREE Up to $235 free merchandise with purchase of one of following CBM-PET items!

		FREE
8032 32K - 80 Column CRT	$1795	235
8016 16K - 80 Column CRT	$1495	205
8050 Dual Disk Drive-1,020,000 Bytes	$1795	235
CBM Modem - IEEE Interface	$395	50
CBM Voice Synthesizer	$395	50
8N Full size graphics keyboard	$795	100
16K Full Size Graphics or Business Keyboard	$995	150
32K Full Size Graphics or Business Keyboard	$1295	205
2040 Dual Disk Drive - 343,000 bytes	$1295	205
2022 Tractor Feed Printer	$795	100
C2N External Cassette Deck	$95	12
Used PETs (8, 16, and 32K)		CALL

WRITE FOR SYSTEM PRICES

****EDUCATIONAL DISCOUNTS****
Buy 2 PET Computers, get 1 FREE

CBM Full Size Graphics Keyboard	$ 74
WordPro III - 32K CBM, disk, printer	$170
WordPro IV - 8032, disk, printer	$255
VISICALC for PET, ATARI	$170
BPI General Ledger, A/P, A/R for PET	$270
Programmers Toolkit - PET ROM Utilities	$34.90
PET Spacemaker Switch	$24.90
Dust Cover for PET	$6.90
IEEE-Parallel Printer Interface for PET	$65.00
IEEE-RS232 Printer Interface for PET	$149.00

EPSON MX-80 Printer	$545
C. Itoh Daisy Wheel Printer	$1575
Centronics 737 Proportional Spacing Printer	$795
NEC Spinwriter - parallel	$2450
XYMEC HI-Q 1000 Intelligent Daisy Wheel	$1975
Leedex Video 100 12" Monitor	$129
ZENITH DATA SYSTEMS	
Zenith Z19 Terminal (factory asm.)	$735
Zenith Z89 with 48K	$2150
SYM-1	$209
SYM BAS-1 BASIC or RAE-1/2 Assembler	$ 85
KTM-2/80 Synertek Video Board	$349
KIM-1 (add $34 for power supply)	$159
Seawell Motherboard - 4K RAM	$195
Seawell 16K Static RAM - KIM, SYM, AIM	$320
KL-4M Four Voice Music Board and Visible Music Monitor (4 Voice) for PET	$59.90

FLEXFILE Database-Report Writer by Michael Riley
Flexible file handler for PET/CBM

MICRO-REVERSI for PET by Michael Riley $9.95
Machine language version—you can't win at Level 5.

PAPER-MATE 60 Command PET Word Processor $29.95
Full-featured version by Michael Riley

A P Products 15% OFF
A P Hobby-Blox 15% OFF

ALL BOOK and SOFTWARE PRICES DISCOUNTED

The 8086 Book (Osborne)	$12.75
Z8000 Assembly Language Programming	$10.60
PET Personal Computer Guide (Osborne)	$12.75
PET and the IEEE-488 Bus (Osborne)	$12.75
6502 Assembly Language (Osborne)	$ 9.90
Programming the 6502 (Zaks)	$10.45
6502 Applications Book (Zaks)	$10.45
6502 Software Cookbook (Scelbi)	$ 9.45
CP/M Handbook (w/ MP/M) Zaks	$11.85

115 E. Stump Road
Montgomeryville, PA 18936 **215-699-5826** **A B Computers**

WRITE FOR CATALOG.
Add $1.25 per order for shipping. We pay balance of UPS surface charges on all prepaid orders. Prices listed are on cash discount basis. Regular prices slightly higher.

PET Symbolic Disassembler

Most disassemblers output only absolute addresses. This symbolic disassembler generates labels and symbols for these addresses, making the disassembled program much more understandable.

Werner Kolbe
Hardstrasse 77
CH5432 Neuenhof
Switzerland

If you want to understand how programs written in machine code work, you need a clear and comprehensive listing. For this purpose I developed a disassembler (listing 1) which produces a listing nearly like one from an assembler—only the remarks are missing. Because I used the program mainly to get a complete listing of the PET ROM routines, the program also contains a short machine routine to overcome the PEEK limitation. The program automatically creates names for all memory locations used by the machine program. It also calculates the absolute addresses of all branches and assigns labels automatically. Listing 2 shows an example of a listing produced by the disassembler.

Using the Disassembler

Because the program is very simple to use, explanations are not included, in order to save space. The program first asks for the output device number, which should be 3 for the screen or 4 for the printer. Then it asks for the start address (origin) of the program to be disassembled. The address has to be entered in hexadecimal without a leading $ sign. Leading zeros are not necessary. Program operation can be stopped by pressing the SPACE key and continued by pressing the key a second time. By pressing the HOME key, the program jumps back to the start, asking again for the output device and the start address. However, it still holds all names created thus far. To get a complete listing it is necessary to go through the machine code a second time, because all labels created by backward jumps can only be resolved in a second pass. Tables of the labels and variable names are printed when, instead of a start address, the following keywords are entered:

- PM all labels are listed
- PL only the labels created by a branch (they start with L) are listed
- PJ only the labels created by a JMP or JSR are listed (they start with J)
- PZ a listing of all zero-page memory places used is given (their names start with Z)
- PW a listing of all memory places above page zero used by the program is given (names starting with W).

By entering "ENTRY" instead of a start address, you can name your own *variables*. This must be done before the program has created a name, because a variable cannot be defined twice. Provision for entering your own *label* names has not been made. You can leave the "ENTRY"-mode by entering END as a name. Addresses in "ENTRY"-mode must be hexadecimal without a leading $. The program itself occupies less than 4K bytes, but to provide enough space for all addresses and names that have to be stored, you should have at least about 16K bytes of memory available.

Program Description

Line No.	Description
5	determines the dimension of arrays L,L$, and P,P$ which contain the labels (L) and the page >0 (P) names. If you have a new ROM and enough memory you could set DM and PM to higher values.
10...15	loads a short machine routine which overcomes the PEEK limitation in the old ROMs.
20	dimensions the arrays. M$ = mnemonic code, Z% = zero-page addresses, Z$ = zero-page names, L, P (see above)
25	the addresses 0 are assigned to provide the headings for the listings.
35	reads mnemonic code, including addressing index.
45	assigns output device.
50...72	FL is a "text"-flag. If FL = 0, then text mode is assumed until a zero is found. The "address" in E$ is investigated in the following lines to detect the keywords.
75...90	subroutine 280 changes E$ into a number in E. P is a pointer to the current address. Subroutine 325 searches for a label under the address in P. S = 1 tells the subroutine not to create a label if none is found. Subroutine 300 performs a decimal to hexadecimal conversion. Subroutine 295 performs an "E = PEEK(P)" and the hexadecimal of E is returned in E$.

92 separates mnemonic code and addressing index.

95 jumps to the subroutines according to the addressing index.

105 the start/stop feature using the SPACE key.

130 prints the mnemonic code at the correct place. As TAB does not work on my printer, a counter K and the SPC function is used.

135...160 handles implied addressing and text.

165 immediate addressing mode.

170 zero-page addressing mode.

175 fetch operand; print it and mnemonic; look for entry in the address table and create name, if not found (subroutine 380); print name plus addressing string.

180 zero-page, X addressing mode.

185 zero-page, Y addressing mode.

190 absolute addressing.

195...205 same task as 175, only for absolute addressing. If no new names can be created because the arrays P,P$ are full, the address is printed in hexadecimal.

210 absolute, X addressing.

215 absolute, Y addressing.

220...230 relative addressing. If the label was created anew and the branch is backward, the address is printed in hexadecimal. The same occurs if the arrays L,L$ are full.

235 (indirect), X addressing.

240 (indirect), Y addressing.

250 absolute indirect addressing; the operand is printed in hexadecimal. It seemed to be unnecessary to create a table of two-byte pointer names only for this seldom-used addressing mode.

260 accumulator addressing.

265...275 JMP and JSR absolute. Similar task as 220...230.

280...285 converts hexidecimal in E$ to decimal in E.

290...320 increments code pointer P, PEEKS the memory location P, and returns the value in E and ES.

325...375 looks for the address E in the array L; if found, returns corresponding name in L$; if not, creates a new name and sorts the table. The new name starts with "L" or "J" and continues with a current number which is the "end of table L" pointer LL. A binary search is executed to improve program speed.

380...415 same as above but for zero-page addresses.

420...465 same as above but for addresses above page zero.

470...540 mnemonic code and assigned addressing index.

545...605 prints address tables and enables entry of addresses and corresponding names.

Werner Kolbe is a German computer enthusiast who has been living in Switzerland since 1977, working as an electrical engineer in the field of power system protection. In 1978 he bought one of the first PETs available in Europe. With some programming experience in FORTRAN gained on a large IBM computer, using his PET is really enjoyable.

DISASSEMBLER

```
5 DM=255:PM=50
10 FORI=1011TO1017:READA:POKEI,A:NEXT
15 DATA173,0,0,141,250,3,96
20 DIMM$(255),L$(DM),L(DM),Z%(255),Z$(255),P(PM),P$(PM)
25 DM=DM-1:PM=PM-1:Z$(0)="ZERO":L$(0)="LABEL":P$(0)="PAGE >0"
30 FORI=0TO255:READM$(I):NEXT
45 CLOSE1:INPUT"OUTPUT DEV. #";D:PRINT:OPEN1,D,1
50 FL=1:INPUT" ORG";E$:PRINT:IFE$="PM"THENV$="":GOTO545
55 IFE$="PL"THENV$="L":GOTO545
60 IFE$="PJ"THENV$="J":GOTO545
65 IFE$="PZ"THEN560
70 IFE$="PW"THEN570
72 IFE$="ENTRY"THEN600
75 GOSUB280:P=E-1
80 P=P+1:E=P:S=1:GOSUB325:IFL$<>" "THENFL=1
85 GOSUB300
90 PRINT#1,SPC(5-LEN(E$))E$;:GOSUB295:PRINT#1," "E$;:K=5
92 M$=LEFT$(M$(E),3):B=VAL(MID$(M$(E),4))
95 ONB*FL+1GOSUB135,165,170,180,185,190,210,215,220,235,240,250,260,265
105 GETE$:IFE$<>""THENV$=" ":GOSUB115
110 GOTO80
115 IFE$=V$THENGETE$:V$="":GOTO115
120 IFE$="S"THENPRINT#1:GOTO45
125 RETURN
```

MICRO wants to obtain more

Foreign and Domestic Dealers and Distributors

To learn why you should carry MICRO products, (MICRO - the 6502 Journal, The BEST OF MICRO Series, and additional special interest publications) write or call:

James Anderson
Marketing Director

MICRO wants to tell advertisers more about why they should advertise in forthcoming issues.
For details, please write or call:

Cathi Bland
Advertising Manager

MICRO
P.O. Box 6502
Chelmsford, MA 01824
(617) 256-5515

PROGRESSIVE COMPUTER SOFTWARE
405 Corbin Rd., York, Pa. 17403
(717) 845-4954

PCS

SOFTWARE-HARDWARE-SYSTEMS
CUSTOM PROGRAMMING

APPLE & HDE

ANNOUNCING!

★ **DISK HEAD CLEANING KIT** ★

Designed to professionally clean your HDE or APPLE 5-1/4" disk unit heads without disassembling the drive. Available soon for HDE 8" drives.

PCS01DC **HDE 5-1/4"**
PCS02DC **APPLE 5-1/4"**

Complete kit for many cleanings — order now!

$24.95

Send SASE for complete catalog or PCS/HDE pgm. library information.

sophisticated routine can always be substituted if desired. FILLPK is doubtless very wasteful on time, but quite frugal on memory. It merely goes round the previous cards selected and rejects any valid trio-decimal smaller than 3D, which has been selected previously. Despite its time inefficiency, it only takes a fraction of a second to shuffle and deal out the whole deck. The rest of the program, apart from the deliberately built in delay and pauses, is virtually instantaneous.

The bidding program runs from maximum points downwards, finishing with no-bid in default of any valid bid. Routines for no-trumps, long suit, clubs etc., are used more than once. The bidding program should be clear from the comments in the listing, and the previously mentioned bidding criteria. The only aspect which possibly warrants any comment is the no-trump bidding.

On the possibly debatable hypothesis that a valid no-trump call takes precedence over others, the criteria adopted for no-trumps are the three that I mentioned earlier. There is at least partial overlapping with some of the other routines essential to the program, and therefore several different possibilities of programming for no-trumps. The method I have chosen is not necessarily the shortest or best. In addition to the short suit tests for voids and singletons and doubletons, I have employed a test for 15 or 21 high card points. No additional tests are necessary since 14 high-card points, for example, together with the mandatory minimum of 16 total points, indicate at least 2 distribution points, and hence one 6-card suit, or two five-card suits etc. It should be evident that all of these are inconsistent with the short suit criteria for no-trumps, and hence would in any case be invalidated later on, on that count. Incidentally, once any distribution points are discovered, there must obviously be at least one biddable suit present. It is then superfluous to check for a one club call, and so the routine skips LONGSU and branches direct to FILLSU. LONGSU is the general routine which compares the total points with the high-card points. If these are equal, there are no distribution points, no long suit, and one club is called. If they are unequal, FILLSU transfers the longest and strongest suit to DISBUF.

SYM has not been programmed to bid the ACOL or Precision systems or the like, but simply a homely hybrid of amateur conventions. Experienced bridge players may be particularly shocked that SYM calls one club with any good 19 to 21 point hand, and also with a large range of hands strong and weak containing no long suit, in addition to the regular genuine one club hand. But all of this can be modified in the programming to conform with *any* conventions according to personal tastes and individual requirements.

If desired, the program can be easily be altered to continue round all four hands after valid bids, to redisplay a hand, to bid preselected or problem hands, and a scoring and timing system could be added to increase tension and interest. If an extra page or so of RAM is available, SYM's onboard 'scope interface can be exploited to display each complete hand much more aesthetically, incorporating conventional card graphics for the four suits if so desired.

This routine certainly does not cover all eventualities; I don't know whether there are any conventions which do. The other day SYM's North "picked up", Spades—void; Hearts—A,K,Q,J,8,7,5,4,2; Diamonds—void; Clubs—T,9,8,4:—and naturally "1.H." appeared on the display. My shouting that this was an obvious four or five hearts, or at the very least one strong club, in order to keep the bidding open for one round, was, of course, a waste of breath. At least nobody screamed back! To obviate similar anomalies, it might be worth adding bonus distribution points on an increasing sliding scale for 8-card suits and longer.

I should like to take this opportunity to thank the several experienced SYMmers in the United States, England, and New Zealand who have assisted and encouraged me by corresponding over the past year. I should also like to express my appreciation to the authors of *The First Book of KIM*. Apart from generally acting as a stimulating catalyst and learning tool, I have also modified and built upon basic ideas contained in their Blackjack, Card Dealer and Sort routines.

Len Green was born and educated in London, has travelled extensively in Europe, and now resides in Israel, where he has lived for nearly 30 years. He enjoys experimenting with his SYM, which he purchased in 1978, and he still considers himself to be a "newcomer" to hobbyist computing.

MICRO

Bad News — Good News

About MICRO Subscriptions

The *bad* news — **MICRO** must raise subscription rates, to cover increased costs of envelopes, postage, and handling.

The **good** news — the increase is only $3.00, and the cover price will remain unchanged! Subscribers will be saving 25% over single-copy purchases.

More Good News—you can still enjoy **MICRO** for the next two years *without* a price increase *because* — **MICRO** will accept one- and two-year subscriptions, or renewals at the *present* rates until March 15, 1981!

Send your order c/o Circulation Department, MICRO, P.O. Box 6502, Chelmsford, MA 01824.

Type of Subscription	Rates	
	Current	New
U.S.	$15.00	$18.00
Foreign surface mail	18.00	21.00
Central America — Air	27.00	30.00
South America/Europe — Air	33.00	36.00
Other air mail	39.00	42.00

EXCERT, INCORPORATED
AIM-65 SYSTEMS

EXCERT specializes in AIM-65 System integration. In addition, we sell industrial quality power supplies from Condor, cases from the Enclosures Group, and expansion products from Seawell, MTU, The Computerist, Cubit, Optimal Technology and we integrate these items into the exact System *YOU* want! These Systems are completely *Assembled* and *Tested* (whenever possible and at no additional charge) and *Warranteed* for six (6) months. All *YOU* have to do is plug in the line cord.

Here are several of the more popular configurations:

"BASIC" System	4K AIM w/BASIC	A65-4B	$510
	Case & Power Supply	ENC4	100
	Assembly & Test	N/C TOTAL	610
"A/D" System	4K AIM	A65-4	$420
	8 A/D in-2 D/A out, 500 conv/sec	ADC1	115
	Cable	CABLE	25
	Case & Power Supply	ENC5A	129
	Assembly & Test	N/C TOTAL	689
"32K" System	0K AIM	A65-0	$360
	32K DRAM Plus™ -EPROM-I/O	MEB4	395
	Cable	CABLE	15
	Case & Power Supply	ENC3A	119
	Assembly & Test	N/C TOTAL	889

For more information, call or write for our complete Product Sheet:
EXCERT, INC
P.O. Box 8600
White Bear Lake, MN 55110
(612) 426-4114

Skyles Electric Works

BASIC Programmer's, Toolkit™, Disk-O-Pro™, Command-O™

For PET™ Owners Who Want More Fun And Fewer Errors with Their Programs

Here are thirty-five commands you'll need, all on dual chips you can install in two minutes without tools, **on any PET or PET system.** 2KB or 4KB of ROM firmware on each chip with a collection of machine language programs available from the time you turn on your PET to the time you shut it off. No tape to load or to interfere with any running programs.

For PET/CBM 2001-8, -8N, -16N/B, -32N/B, 3016 and 3032

BASIC Programmers Toolkit™ commands

AUTO[ed] DELETE[ed] RENUMBER[ed] HELP[ed] TRACE[ed]
STEP[ed] OFF[ed] APPEND[ed] DUMP[ed] FIND[ed]

BASIC Programmers Disk-O-Pro™

CONCAT[B80] DOPEN[B80] DCLOSE[B80] RECORD[B80] HEADER[B80] COLLECT[B80]
BACKUP[B80] COPY[B80] APPEND[B80] DSAVE[B80] DLOAD[B80] CATALOG[B80]
RENAME[B80] SCRATCH[B80] DIRECTORY[B80] INITIALIZE[BS] MERGE[BS] EXECUTE[BS]
SCROLL[ed] OUT[ed] SET[ed] KILL[ed] EAT[ed] PRINT USING[BS] SEND[BS] BEEP[BS]

```
RUN

?DIVISION BY ZERO ERROR IN 500
READY.
HELP
 500 J = SQR(A*B/C)

READY
```

```
APPEND "INPUT"

PRESS PLAY ON TAPE #1
OK

SEARCHING FOR INPUT
FOUND INPUT
APPENDING

READY
```

```
RUN

READY.
DUMP
A1 = 10
BW = -6.1
C$ = "HI"

READY.
```

NOTES:

ed — a program editing and debugging command
B80 — a BASIC command also available on Commodore CBM™ 8016 and 8032 computers.
BS — a Skyles Electric Works added value BASIC command.
BASIC Programmers Toolkit™ is a trademark of Palo Alto IC's.
BASIC Programmers Disk-O-Pro™, Command-O™ are trademarks of Skyles Electric Works.
PET™, CBM™ are trademarks of Commodore Business Machines.

AVAILABLE: USA/CANADA: Please contact your local dealer
England: Calco Software Lakeside House, Kingston Hill, Surrey KT2 7QT
GERMANY: Unternehmensberatung, Axel Brocker Lennebergestr 4, 6500 Mainz
Japan: Systems Formulate, 1-8-17 Yaesu Shinmaki-cho Bldg. 11F Chuo-ku, Tokyo JAPAN 103

Phone or write for information. We'll be delighted to answer any questions and to send you the complete information package.

Skyles Electric Works | **231 E South Whisman Road Mountain View, CA 94041 (415) 965-1735**

Skyles Electric Works

BASIC Programmer's, Toolkit™, Disk-O-Pro™, Command-O™

For CBM™ Owners Who Want More Fun And Fewer Errors with Their Programs

Here are nineteen commands you'll need, on a single chip you can install in two minutes without tools, **on any CBM or CMB system.** 4KB of ROM firmware on each chip with a collection of machine language programs available from the time you turn on your PET to the time you shut it off.

For CBM 8016 and 8032; BASIC 4.0

BASIC Programmers Command-O™

AUTOed **DUMP**ed **DELETE**ed **FIND**ed (improved) **HELP**ed **KILL**ed **OFF**ed **TRACE**ed (improved) **RENUMBER**ed (improved) **INITIALIZE**BS **MERGE**BS **MOVE**BS **EXECUTE**BS **SCROLL**ed **OUT**ed **SET**ed **SEND**BS **PRINT USING**BS **BEEP**BS

```
  100 GOSUB 180
  105 PRINT USING C$, A, B$
  130 INPUT "TIME", D$
  131 INPUT "DAY", E$
  160 IFB-  -C THEN 105
  180 FOR X -IT09
  183 PRINT Y(X) NEXT
  184 RETURN
  200 I - X/19
READY

RENUMBER 110, 10, 105-184

READY
LIST
  100 GOSUB 150
  110 PRINT USING C$, A, B$
  120 INPUT "TIME", D$
  130 INPUT "DAY", E$
  140 IFB-  -C THEN 110
  150 FOR X -IT09
  160 PRINT Y(X) NEXT
  170 RETURN
  200 I - X/19
READY
```

```
MERGE D1 "BUY NOW*"

SEARCHING FOR BUY NOW*
LOADING

READY
RENUMBER 100, 10

READY
FIND B$
110 PRINT USING A$, [redacted] - C$ - D$
280 [redacted] "NOW IS THE TIME"

READY
```

```
  580 BA  BA  1
  590 RA  123*5X.92 - BA*10
  600 IF BA  143 THEN 580
  610 RETURN
  620 C$  "PROFIT $#, ###.## DAILY"
  630 PRINT USING C$, PI
  640 D$  "LOSS  $#, ### ## DAILY"
  650 PRINT USING D$, LI

RUN

PROFIT $1, 238.61 DAILY
LOSS  $     0.00 DAILY

READY
```

NOTICE

When you order **Command-O,** we will **loan** you a **Toolkit** until we deliver **Command-O.**

NOTICE

PRICES:

BASIC Programmers **Toolkit™** (chip only)	$40.00
BASIC Programmers **Disk-O-Pro™** (chip only)	$75.00
BASIC Programmers **Command-O™** (chip only)	$75.00
Interface boards (needed sometimes)	$20.00-$50.00
Instruction Manual (with redeemable $5.00 coupon)	$5.00

Shipping and handling $2.50 USA/Canada, $10.00 Europe/Asia

California residents please add 6% or 6-1/2% sales tax as required

*Reserve your **Disk-O-Pro, Command-O** today*

Toolkit™ immediate delivery, Disk-O-Pro delivery in December, Command-O delivery in January

VISA, MASTERCHARGE ORDERS CALL (800) 538-3083 (except California residents)

CALIFORNIA ORDERS PLEASE CALL (408) 257-9140

Skyles Electric Works | 231 E South Whisman Road, Mountain View, CA 94041, (415) 965-1735

APPLE & PET

MAE

The Most Powerful Disk-Based Macro Assembler/Text Editor Available at ANY Price

Now includes the Simplified Text Processor (STP)

For 32K PET, disk 3.0 or 4.0 ROMS or 8032 (specify) — OR — 48K APPLE II or APPLE II + and DISK II

MAE FEATURES
- Control Files for Assembling Multiple named source files from disk
- Sorted Symbol table — Up to 31 chars./label
- 27 Commands, 26 Pseudo-ops, 39 Error Codes
- Macros, Conditional Assembly, and a new feature we developed called Interactive Assembly
- Relocatable Object Code
- String search and replace, move, copy, automatic line numbering, etc.

STP FEATURES
- 17 text processing macros
- Right and left justification
- Variable page lengths and widths
- Document size limited only by disk capacity
- Software lower case provision for APPLE II without lower case modification

ALSO INCLUDED
- Relocating Loader
- Sweet 16 macro library for APPLE and PET
- Machine Language macro library
- Sample files for Assembly and text processing
- Separate manuals for both APPLE and PET

PRICE
- MAE, STP, Relocating Loader, Library files, 50 page manual, diskette — $169.95

SEND FOR FREE DETAILED SPEC SHEET

EASTERN HOUSE SOFTWARE
3239 LINDA DRIVE
WINSTON-SALEM, N. C. 27106

(919) 924-2889 (919) 748-8446

THE MAILING LABEL AND FILING SYSTEM

From Avant-Garde Creations **only $24.95 ppd.**

This unique system will handle both your filing needs and your mailing label needs.

It's uniqueness starts with user-determined variables (up to 10 options) and continues with a special COUNT/SORT routine that allows the user to sort up to 9 VALUES for each of any 9 (out of 18) variables. It will print mailing labels, do a regular print-out or just display the criteria-meeting records while it counts them. It will also range-sort for 3 particular variables.

It makes an alphabetized directory of names and record numbers. You can find records by name or by numbers in seconds. If you don't know the exact spelling there's a quick-find option for directory-reading.

You can customize your labels and print up to 6 lines of your variables on them.

It includes special quick-copy and backup programs.

An easy to use system, brimming with options and dynamics, which ends the need for separate filing and mailing label programs.

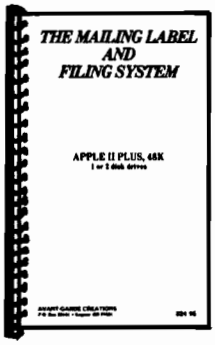
Includes 40-page program manual and disk. APPLE II PLUS, 48K, one or two disk drives.

$24.95 ppd.
We accept VISA/Mastercharge

Write for our detailed brochure and more information:

Avant-Garde Creations
P.O. Box 30161
Eugene, OR 97403

Dept. m



Winchester Backup

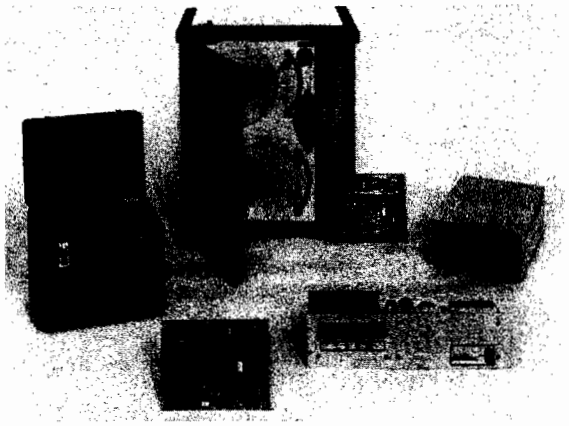
ALLOY offers COST-EFFECTIVE Tape Sub-systems boasting File-Oriented Backup. Using this technique, a Maintenance backup need only contain Active or Changed Files. Furthermore, a single file may be restored without disturbing other modified files. The Cartridge-Tape products support the "FUNNEL" 6400 bpi product by Data Electronics (**DEI**). The 9-TRACK products support most Industry Compatible Formatted tape units including the new **Cipher** Microstreamer.

Cartridge-Tape Subsystems with up to **13.4 Mbytes** are available for:

- Data General **microNOVA and NOVA**
- Ohio Scientific **Challenger C2/C3**
- **S-100** under both **CP/M™ & MP/M**
- Intel **Multibus** under **ISIS**
- **Z-80** processor piggy-back 5" x 5"
- General Purpose **RS-232 & PARALLEL**

9-Track Tape Subsystems with up to **45 Mbytes** are available for:

- Ohio Scientific **Challenger C2/C3**
- **S-100** under both CP/M™ **& MP/M**



ALLOY ENGINEERING COMPANY, INC.

Computer Products Division

85 Speen Street
Framingham, MA 01701
(617)620-1710
TWX: 710-380-7624