The independent monthly magazine for users of Timex Sinclair Computers

BUILD A JOYSTICK Complete project inside

timex sinclair user

48516
Volume 1#6
U.S. $2.95
Canada $3.50

'Clever Contest' Winners Announced and their programs listed

GETTING ORGANIZED
With File Programs

DISA-Z
An Assembler Program in Basic

T/S2068 Display Primer
by Fred Blechman

Programming Arcade Games II

Preparing for Machine Language Programming

PLUS new software, hardware & book reviews and much more!

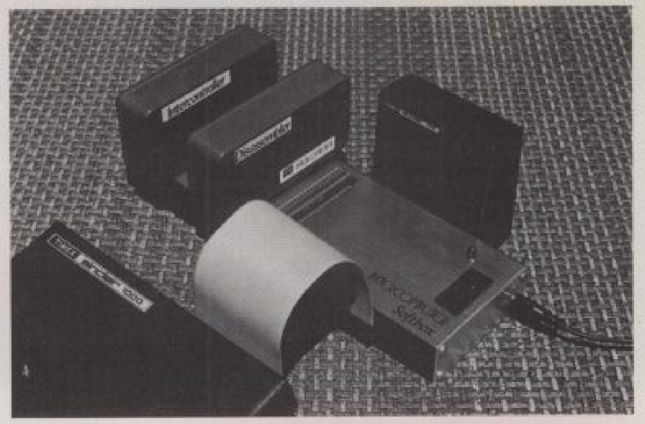
INTERCOMPUTER INC.
T.M.

Expands the use of your TIMEX SINCLAIR COMPUTERS

SOFTBOX™

- Expands the expansion port of your Timex Sinclair computer
- Has 4 gold plated, fully buffered ports
- Software selectable, any desired port/ports
- Longer life to your computer console and program cartridge
- Has LED indicator, on/off switch and DC power input jack
- Provides power to your computer and peripherals
- Eliminates the computer overloading and peripheral connection problems **#PL7012** **$59.95**

Needs flexible ribbon connector; will be available for TS1500 and TS2000 computers



DATA COMMUNICATION PACKAGE

- Now your Timex computer can communicate directly on your telephone line with other computers and data base networks such as COMPUSERVE, DOW JONES NEWS RETRIEVAL, etc.

1. High Quality Modem 0-300 BAUD **#PL7017** **$129.95**
2. RS232 Interface **#PL7006** **$99.95**
3. Connecting Cable **#PL7018** **$19.95**

PACKAGE including items 1, 2, & 3 **#PL7020** **$197.95**

PRINTER'S PACKAGE

Our printer package enables you to printout all the received information

1. 80-column Dot Matrix Printer
2. MEMOTECH Centronics Interface
3. Printer Cable **#PL7019** **$299.95**

MEMOTECH PRODUCTS

MEMOPAK 16K	**#PL7002**	**$44.95**
MEMOPAK 32K	**#PL7003**	**$89.95**
MEMOPAK 64K	**#PL7004**	**$134.95**
HIGH RESOLUTION GRAPHICS	**#PL7007**	**$89.95**
KEYBOARD	**#PL7011**	**$89.95**
MEMOCALC	**#BS2001C**	**$53.95**
MEMOTEXT	**#PL7014**	**$44.95**
MEMOASSEMBLER	**#PL7015**	**$44.95**

QSAVE

A Reliable Fast Loading System

- Operates 16 times faster than the Sinclair system (i.e. Loads/Saves 8K in 10 secs; 48K in 110 secs)
- Operates with any memory and on any program length
- Works equally well with AUTORUN programs, BASIC or MACHINE CODE
- This is what you get:
  1. A cassette interface to improve the signal quality. To make loading most reliable an AUTOMATIC LEVEL CONTROL, an OVERLOAD WARNING LIGHT, a SAVE/LOAD SWITCH, an ON/OFF SWITCH are also built into this unit.
  2. Highly sophisticated, superbly easy to use 1/2K software for up to 16K and 64K programs **#PL7001** **$37.95**

- **QSAVE** software on **Solid State cartridge** available separately **#PL7001C** **$24.95**
- Complete unit with software on **Solid State cartridge** **#PL7001B** **$57.95**

Cartridges need Flexible Ribbon Connector or Softbox

MCODER (16K RAM)

- The first real integer BASIC compiler for the TS1000 computer
- Give your BASIC program the speed of MACHINE language programs
- 75 times and in some cases up to 900 times speed improvement
- Very simple to use:
  1. Load MCODER
  2. Write your BASIC program according to the manual and compile it through the MCODER by using a single PRINT USR command

ASSEMBLER/DEBUGGER "ZXAD" (16K RAM)

- Write machine code quickly and easily
- Two pass assembler, ZILOG mnemonics, 8 pseudo-OPS
- Debugger includes examine/modifying, breakpoint with register display
- Occupies 7K, comprehensive multi-page manual **#PT4003** **$18.95**

STOPPER

- Enables you to break through any machine code software, list it on the screen, save it on another tape, or print it out
- Very simple to use, load STOPPER into your TS1000 and follow the simple instructions on the screen **#PT4002** **$18.95**

DISASSEMBLER

(Decoding Tool)
On Solid State Cartridge

- Disassemble any assembly language code on memory
- Assembly language games and other programs **#PT4004** **$34.95**

Needs Flexible Ribbon Connector or Softbox

FLEXIBLE RIBBON CONNECTOR

No More System Crashes

- Connects any RAM expansion to your TS1000
- Prevents usual system crashes caused by the computer or the RAM movement
- 9 inches long flat ribbon cable **#PL7016** **$17.95**

DEALERS/DISTRIBUTORS PACKAGE AVAILABLE.
Phone orders: (617) 738-5310 7 days a week.
Call or write to us for FREE brochure and information on our full line of ***software.***

Mail to: Intercomputer, Inc.,
358 Chestnut Hill Ave., Boston, MA 02146
Tel: (617) 738-5310
Telex: 951141COFAR

PART #	QTY.	UNIT	U.S. $	TOTAL
Shipping and Handling for MEMOTECH and QSAVE			4.95	
for EACH PACKAGE			9.95	
Mass. Residents Tax			5%	
			Total	

☐ Visa ☐ M.C. ☐ Check

Account No. ________ Exp. ____

Name ________

Address ________

City ________ State ____ Zip ____

No Shipping & Handling Charges for Cassettes

All products are compatible with TS1000 and TS1500 Computers

For more information, circle 1 on Reader Service Card.

"GET ACQUAINTED" OFFER

GET TWO TIMEWORKS™ PROGRAMS FREE

When You Buy TIMEWORKS™ "Computer Control Center"™

For Timex-Sinclair 1000 and Sinclair ZX-81 Computers.

Only $29⁹⁵

FREE

Any Two TIMEWORKS Programs Listed Below with the "Computer Control Center" (retail value $16.95 ea.)

$64.00 value
for $29.95
You save $34.05

A handsome, compact work station that consolidates work space and provides remarkably faster, easier operation!

Features:

- Effectively eliminates cluttered cables in your work area.
- ON/OFF Switch eliminates plugging and unplugging.
- Accommodates all brands of 16K, 32K, 64K RAM packs.
- Allows provision for one Printer and Tape Deck hook-up.
- Holds your Software tapes in neat, specially formed pockets.
- High impact, black molded plastic. 3⅝" high, 20" deep, 14½" wide.
- Accommodates a 13" TV screen.

Cat. No	
1101	**THE QUIZ KIT™** Educational learning system. Construct your own quizzes. 16K Req.
1201	**5-2 GAMES™** Five challenging and entertaining games. 2K Req.
1202	**STAR BATTLE™** Realistic deep space adventure to save Earth. 16K Req.
1203	**ROBBERS OF THE LOST TOMB™** Perilous adventure in search of the Sacred Tablets. 16K Req.
1204	**WALL STREET™** A competitive game of financial speculation. 16K Req.
1205	**PRESIDENTIAL CAMPAIGN™** Conduct a nationwide campaign to become the next President of the United States. 16K Req.
1207	**SCYON'S REVENGE™** Realistic deep space combat adventure. 16K Req.
1301	**THE MONEY MANAGER™** Home and business budget and cash flow system. 16K Req.
1302	**THE COLLECTOR'S COMPANION™** Cataloging and inventory recording system for all collectibles. 16K Req.
1303	**THE INSURANCE PROPERTY RECORD™** Home contents inventory recording system. 16K Req.
1304	**THE ELECTRONIC CHECKBOOK™** Check recording, sorting and balancing system. 16K Req.
1306	**FORGET-ME-NOT™** A recording and retrieval system for important occasions, events and appointments. 16K Req.
1307	**DATA MASTER™** A general information storage and retrieval system—with exclusive "X-SEARCH"™ Feature. 16K Req.
1308	**5-2K FAMILY PAK™** 5 Household programs for the basic T/S 1000 and Sinclair ZX-81 Computers. 2K Req.
1401	**PROGRAMMING KIT 1™** A practical "How-To" learning approach to Basic programming. 16K Req.

COMPLETE MONEY BACK GUARANTEE

If, for any reason, you are not completely satisfied, you may return The Computer Control Center, in good condition (freight prepaid), for a full cash refund. (Software not included).

RUSH This Coupon Today.

MBR Distributors PO Box 321 Deerfield, IL 60015

Rush me ______ Computer Control Centers @ $29.95/ea $______
Plus postage & handling ______ @ $4.70/ea $______
(Illinois residents add 6% sales tax) $______

ALSO INCLUDE TWO FREE PROGRAMS
(Additional Software @ $16.95 ea)

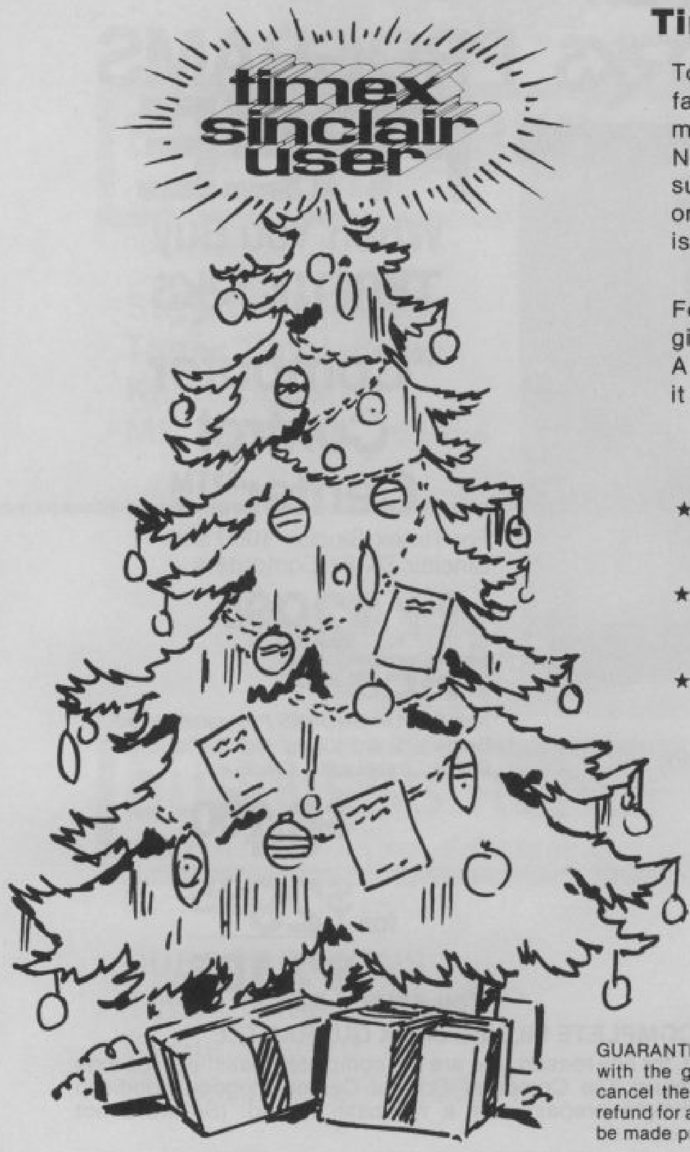
Cat. No.	Qty.		
______	______	@ no charge	$ N/C
______	______	@ no charge	$ N/C
______	______	@ $16.95/ea	$______
______	______	@ $16.95/ea	$______
		TOTAL	$______

Name ______
Address ______
City ______ State ______ Zip ______
Check or MO ☐ VISA ☐ Master Charge ☐ American Express ☐
Card No. ______ Exp. Date ______
Signature ______
Dept.

mbr DISTRIBUTORS
P.O. Box 321, Deerfield, IL 60015 (312) 291-9284

For more information, circle 4 on Reader Service Card.

AN "EARLY-BIRD" CHRISTMAS SUBSCRIPTION GIFT OFFER



Timex Sinclair User at $1.66 per issue!

Too early to think about Christmas gifts? We don't think so! In fact, by thinking about them now you can save yourself some last-minute shopping headaches and some money. From now until November 1, 1983, we are offering a special Christmas gift subscription discount of $15.43. You pay only $19.97 for 12 issues, one full year's subscription, to *Timex Sinclair User*! Only $1.66 per issue!

GIVE A GIFT AND GET A GIFT

For every gift subscription you order now, we will send you, as a gift from us, a copy of the Collector's Issue of *Timex Sinclair User*. A $2.95 value free to you. You may keep the Collector's Copy, give it as a separate gift or include it in with your gift subscription.

ORDER YOUR GIFT SUBSCRIPTIONS NOW

★ Pay only $19.97 for a 12 issue gift subscription, SAVE $15.43 off the regular price.

★ Get a gift from us of a copy of the Collector's *Timex Sinclair User* — a $2.95 value!

★ We send a Christmas card from you to the recipient of your gift subscription.

Mail to:
TIMEX SINCLAIR USER
SPECIAL CHRISTMAS OFFER
Kable Square
Mt. Morris IL 61054

GUARANTEE: If your friends are not happy with the gift you have chosen, you may cancel the subscription and receive a full refund for all unmailed issues. Refunds will be made promptly.

Please list additional gift subscriptions on a separate page.

One Collector's Copy will be sent for each gift subscription. Keep the copies yourself or give them as additional gifts.

A GIFT FOR YOURSELF

Please send me a year's subscription, 12 issues, to Timex Sinclair User plus a gift of a Collector's Copy for the special price of $19.97

Name: ____________________

Address: ____________________

City: __________ State: __________ Zip: __________

( ) Check ( ) Money Order

( ) Bill me ( ) VISA ( ) M.C.

____
card number expiry date

____
signature

A GIFT FOR A FRIEND

Please send a gift card and a one year suscription to TSU to:

Name: ____________________ Send a card

Address: ____________________ ( ) Yes ( ) No

City: ____________________

State: __________ Zip: __________

Send the Collector's Issue to: ( ) Me ( ) My friend

Billing information:

BILL ( ) Check ( ) Money Order ( ) Bill me

( ) VISA ( ) M.C.

Name: ____________________

Address: ____________________

City: __________ State: __________ Zip: __________

____
card number expiry date

____
signature

This offer expires December 1, 1983.

Contents

Programming Arcade Games on p.11

Joysticks evaluated on page 54

Next month

- Complete Christmas product listings
- Programming Arcade Games, Part 3

and much more!

timex sinclair user

President
Terry Cartwright
Editor-in-Chief
Joseph Gladstone
Managing Editor
Bill Gladstone
Art Director
Vladyana Krykorka
Associate Artists
Gwen Egan
Don Kletke
Katarina Altrichter-Raksany
Editorial Assistant
Becky Ashwell
Chairman
Richard Hease

For subscription services or information phone toll-free 1-800-435-0715 (except Illinois)

Timex Sinclair User is published monthly by ECC Publications Inc. in conjunction with C.W. Communications/Inc.

If you would like to contribute to Timex Sinclair User, please send programs, articles or ideas for hardware projects to:
Timex Sinclair User
49 La Salle Avenue
Buffalo, N.Y. 14214-1414
Programs should be on cassette and articles should be typed and double spaced. If you wish your material returned, enclose a stamped, self-addressed envelope. We pay $20 for each program published and $100 per 1,000 words for each article used.

ISSN 0737-0881

Typesetting
International Typesetting
Printing
Clark Franklin Kingston Press
Advertising Sales
JE Publishers Representative Company
6855 Santa Monica Blvd., Suite 200, Los Angeles, CA 90038
Los Angeles (213) 467-2266
New York (212) 724-7767
Denver (303) 595-4331
Dallas (214) 660-2253
San Francisco (415) 864-3252
Chicago (312) 445-2489
Boston (617) 437-7628

Rebuilding the Pyramids

GAMESTAPE 2

Melbourne House, 16K

I. STARFIGHTER

In Starfighter you are battling at the outer edge of the universe amidst a bevy of stars gone Nova. You must destroy as many enemy crafts as possible before your energy runs out; you destroy them by centering them in your sights and firing. The alien crafts come in three different sizes and the destruction of each is worth upwards of 10 points. Starfighter moves slowly, but don't let the pace fool you; it is challenging and fun. The excitement is nerve-wracking, as seconds after you spot an alien, he eludes your sights only to appear a few degrees away. Starfighter is lots of fun.

★★★

II. PYRAMID

Pyramid is an absorbing puzzle in which your task is to rebuild the Pyramids by carefully moving them one layer at a time. There are nine layers to be moved and three bases on which to move them. You must completely rebuild in as few moves as possible. At first glance this game looks simple, but it is very tricky. After moving the first four levels you are hard-pressed to move the fifth without squashing the first four layers. Pyramid is billed as a "thinkers game", and is as difficult to play as a Rubik's Cube. It's a real challenge. ★★★

III. ARTIST

Originally designed as a graphic designer's aid, the special cursor control in Artist allows you to "draw" simple pictures, graphs, and charts. Your computer doodles can be saved on cassette for future use, or copied to a printer. Artist has a memory mode with 10 memories to store your drawings. There is also a write mode which functions as a memory, and finally a Read mode which recalls previously stored drawings. The possible uses of this game are fantastic, especially where the novice is concerned. Both children and adults with or without computer experience will enjoy and learn from this non-competitive game.

★★

Overall rating: ★★★

For more information, circle 5 on the reader service card.

Search and Destroy

SPACETREK

Beam Software, 16K

The object of Spacetrek is to seek out and destroy Drakons. This is done by travelling to different space sectors in search of them. There are five levels of play in Spacetrek, ranging from level one (easy) to level five (impossible). This game takes hours to play and is lots of fun. It requires strategy rather than a quick hand or eye.

Playing Spacetrek will introduce you to some almost familiar characters; Mr. Scotch and Mr. Spark. While travelling through the different space sectors you must keep a watchful eye on your fuel level, but Mr. Scotch will let you know when your shields are dangerously low, and if you have trouble navigating, you can get expert advice from Mr. Spark. ★★★

For more information, circle 6 on reader service card.

Attn. software manufacturers! Got a product you'd like to see reviewed on these pages? Send it to *Timex Sinclair User, Software, 49 La Salle Avenue, Buffalo, New York 14214-1414.*

Software

In The Fast Lane

TIMEBLASTERS

Robotec, 16K

Timeblasters loads easily but at a higher volume than many other T/S video games. In Timeblasters you are at war with aliens who have the technology to control time. When you fire at and miss an enemy ship, or if one slips past you, time accelerates. On the other hand, blowing up alien space stations (by hitting them dead center) will slow time down. You must destroy as many alien spacecraft and space stations as possible in order to survive. There are six levels of play in Timeblasters from Novice to Master, and three cursor control keys. It is one of the fastest games I've played on the T/S,ZX. The Master level of play happens so fast it's almost a blur. You really have to be in control to succeed at this super-paced, exciting game. ★★★★

For more information, circle 7 on reader service card.

Fringe Benefits

ZXAK-MAN

Pleasantrees Programming, 16K

The opening graphics are marvelous. The game is patterned after — can you guess? — PAC MAN. The graphics are well done throughout. There are nine levels, and four ghouls.

The rules are readily understandable and the play is quite simple. The four arrow keys direct ZXAK around the mazes.

The game itself is written in machine code, so it progresses smoothly. The folks down at Pleasant Programming have given us another real advantage with this one. The accompanying slip of paper is small, so don't lose it because it suggests ways of modifying the game. That's right! If you want it faster, harder, easier, whatever, they'll tell you how.

Also included are instructions to make your own joystick at an estimated cost of $2.

All in all, the game alone is worth the money. Toss in the side benefits and it's really a pretty good investment. ★★★

For more information, circle 8 on reader service card.

'Rithmetic

SUPERMATH

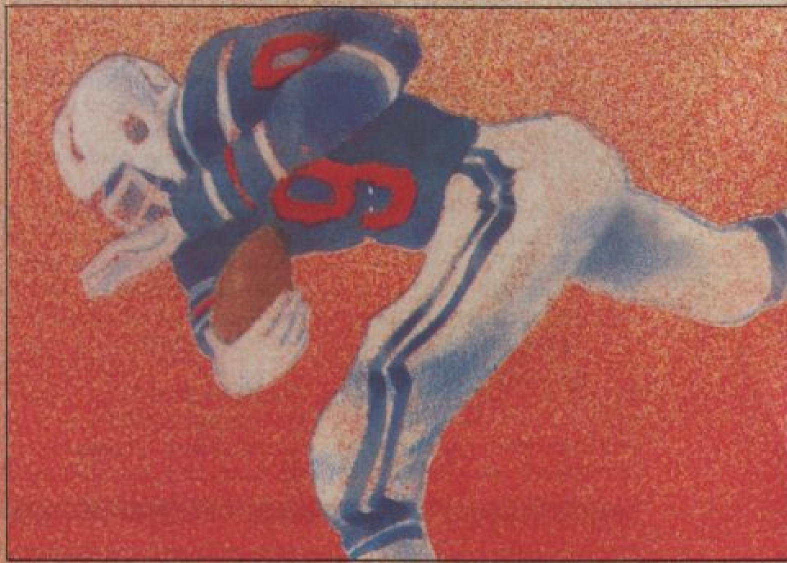
Timex, 2K

Supermath contains five levels of difficulty in each of the four math operations — addition, subtraction, multiplication and division.

This program is useful to drill the basic math skills. This would normally be useful for children in grade school. However, at the upper levels, questions must be copied on to paper and worked out or done on a calculator). The questions are generated at random and are very challenging.

The screen is used dramatically when your solution is correct, but the message when incorrect is somewhat tedious. The graphics and screen set up at the beginning could have been much better. ★★

For more information, circle 9 on reader service card.



For Football Fans

PICK 'EM

Stuart Software, 16K

I guess I'd better make it clear before I begin that although I do follow football I rarely make money betting, not even in the office pool. This program is therefore perfect for me because it says right on the cover that it is "not to be used for any illegal activity." It is supposed to predict the winners and point spread.

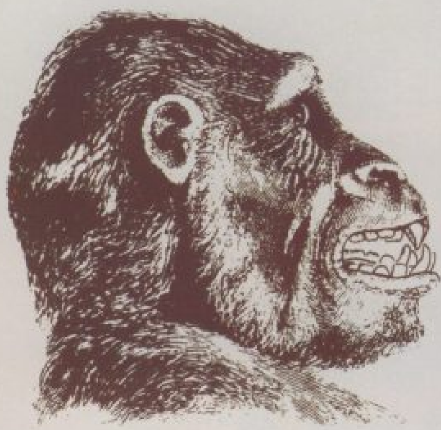
Eagerly I tore open the plastic zip-lock bag. The booklet fell apart — someone had forgotten to staple it. Never mind. With sweating palms and a glazed look I started to read: *Pick home team from list.* Okay. *Same for visitor.* Okay. Now *Points for and against.* Okay. *Number of games played?* Okay. *Any injuries?* was next. Okay. Finally, *Did each team win its last game?* Okay.

Then came a cloud of doubt. The last question after the prediction was "Predict anther *(sic)* game? Who are these Anthers? Are they a sure thing? How did I fail to hear about them?

Anyway I phoned by book . . . no — my accountant, and proceeded to — never mind!

It's a good program — a little overpriced at nearly $20, but with good judgement — and that is recommended all through the manual and on the tape. This could be an interesting aid for a football addict. ★★★

For more information, circle 10 on the reader service card.



Angry Ape

KRAZY KONG
Intercomputer, 16K

Those who appreciate a slow-paced yet challenging game will like this one. In Krazy Kong, you must penetrate the maze-like jungle (Tarzan style) to rescue Jane who is being held captive by (who else?) Kong, while at the same time avoiding the rocks and drums Kong throws at you. Once Jane is rescued, the screen changes and you have to maneuver away from Kong's barrage of rocks and drums.

Krazy Kong could use a bit of arcade flash to dress it up, but it's an absorbing game and in spite of the slow pace, it's not a game which is easily won. ★★★

For more information, circle 11 on reader service card.

Reviews by ***George Miller*** *and* ***M.K. Wilson.***

Technical Reviews

Book-cassette package teaches programming

PROGRAMMING FOR REAL APPLICATIONS
Dilithium Software, 16K

This book-cassette package offers a great way to learn programming. Practice and theory are combined. You get a book with a variety of program listings backed up by explanations and theory, then you purchase the cassette which contains the programs listed in the book. Aside from easing the task of program entry, the cassette contains full-fledged programs which might be useful on their own.

Dilithium offers several book-cassette learning tools for various computers. The principle of teaching this way is not new. Many software companies include models and tutorials on disk with their documentation. Educational institutions also use such methods. For learning to program on the Sinclair, however, it's a first.

Randle Hurley, the author of both the book and the software presents the owner of a TS/ZX with a learning program that is biased towards business applications. In the first chapter, titled "Aims and Assumptions," he tells of how financial restrictions pushed him to develop programs and to modify his TS/ZX so as to create a true business machine. The ways and means of squeezing every last drop of power out of Sinclair BASIC result from necessity and the author proposes to give you the benefit of his experience. There are nine programs that cover the general gamut of business applications: Cash Handling, Personal Finances, Banking, Bulk Storage, Rank Order, Examination Result Analysis and two versions of Word Handler, a rudimentary word processing program.

Before I get into the pros and cons of this work, I can say that on the whole the program tape and book are worthwhile acquisitions for anyone interested in learning advanced BASIC. They are clear, well-structured and contain a variety of useful information and original approaches to the many problems associated with Timex Sinclair computers. The main restrictions are speed of execution and input-output limitations.

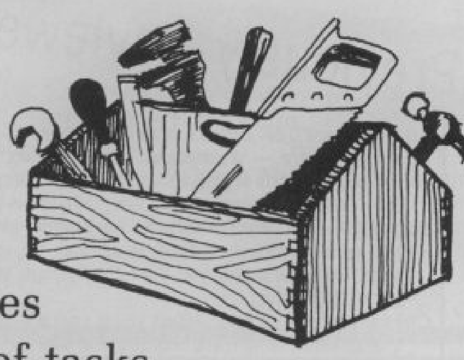
For those who are hell-bent on changing this machine into something useful for serious applications, the general procedure is to buy or make hardware and use machine language for the software. Even with this, it is very difficult to get around the Sinclair logic chip which controls the microprocessor as well as the peripherals. I was certainly intrigued by the authors' claim to have developed real "workhorses" without leaving BASIC. To prove this, Hurley wrote the fourth chapter of the book using his Word Handler program. All I can say after using it myself is that he is either a masochist or amazingly adept at using his own program.

As a learning tool, this program is excellent and uses several original methods of screen manipulation, memory access and storage. It's a valiant attempt that doesn't quite make it. The other programs involving storage and math do not require a constant interactive environment as with word processing and come much closer to the mark. The floating point and string handling routines are the forte of Sinclair BASIC. If we put aside problems of magnetic storage and a membrane keyboard, these programs can be considered "workhorses". Though this tutorial on BASIC programming will not change your Sinclair into an IBM PC, (which is the impression the advertising leaves), it will help you to exploit the possibilities of Sinclair BASIC to the maximum. The book contains excellent programming tips and a great deal of information about the system. The cassette is a perfect complement to the book.

Programming for Real Applications might be best suited for people like the author: short on cash, longing to put the Sinclair to work and not quite ready for machine language programming. Hopefully, Hurley's sequel on machine language will be ready soon. If he manages to exploit machine code to the same extent as he does BASIC, we may expect some real "workhorse" programs.

— André Roussil

For more information, circle 12 on reader service card.



Tape does variety of tasks

ZX-TOOLBOX

Melbourne House, 1K

The programs on this tape were designed to work with 1 or 16K of RAM. Toolbox, in machine code with five graphic and three programmer utilities, locates itself above RAMTOP and is accessed through USR statements from BASIC. Records and Directory are short BASIC programs with machine code routines located in REM statements and were both written by Dr. Ian Logan, whom many consider as the foremost independent expert on the Sinclair machines. Records and Directory are examples of sophisticated programming which bypass the system. Records allows you to create, save and load data files independent of program files. In other words, you can store data on tape without a program attached, and can retrieve the data without overwriting the program that calls it up. Directory reads and lists titles from a cassette and greatly facilitates finding programs. Both are transportable to your own BASIC and aside from being useful, they can be educational to the novice and advanced programmer.

On the other hand, Toolbox is of doubtful utility. Programs of this type are generally four to eight kilobytes long, which might be more memory than you are willing to spare. This one is only 144 bytes long and takes a dozen seconds to load. It's hardly noticeable in memory. This is its main quality, but it has so few features that it is questionable whether you need it at all. The graphic routines may be impressive to those unfamiliar with machine language, and are quite simple to write. They are not directly transportable to your programs and if you managed this, you could probably have written them in the first place. The three programmer utilities (Renumber, Search & Replace and Free) can only be used in specific ways for a single task. Renumber, for instance, renumbers all the program lines without changing the GOTO and GOSUB destinations. This might be tolerable for a 1K program but is more trouble than it is worth for longer ones. If it were not for the two programs supplied by Dr. Logan I could not recommend this tape to anyone including those who have only 1K of RAM. For those who have ever had trouble finding a program on tape or have been frustrated by the whole tape filing system of the Sinclair computer, these two small utilities offer some relief. If you are interested in machine language, do what I did — take them apart to see how they work. — ***A. R.***

For more information, circle 13 on reader service card.

Software

Previously Reviewed

All tapes are 16K unless otherwise noted.

Beam Software — **Wallbusters**. This fast, smooth 1K Batting game offers seven speeds of play, though to change speed you must reload the tape. Despite this programming flaw, the game is pleasing. ★★½

Bug Byte — **Mazogs**. Superlative maze game, playable at three levels that correspond from difficult to suicidal. On your way through the large, complex maze, you fight the Mazogs you encounter with a sword. Exhilirating. ★★★★

Chessmaster — **Chess**. Allows you to save unfinished games for later, change sides, resign, and choose one of seven levels of difficulty. Of all chess programs, our reviewer gave this one top marks. ★★★★

International Publishing & Software — **Flashcard**. In the classroom, a flashcard has a problem on one side and an answer on the other. This program lets you input the problems and answers, poses questions at random, and gives a score at the end. ★★★

International Publishing & Software — **Galactic Invaders**. Seven alien ships fly over your laser base; when you destroy them, the next regiment appears. You get to choose the speed. Annoyingly, some of the aliens take long rests on the ground, but the game is still a must for the video game connoisseur. ★★★

International Publishing & Software — **Home Money Manager**. Lets you keep track of your finances on a monthly basis, then do a month-by-month or full year balance sheet. Well-documented, well-conceived, user-friendly. ★★★½

International Publishing & Software — **80 Hours Around Europe**. A refreshing change from space games. You visit 12 European cities in 80 hours with 1,500 British pounds to spend on souvenirs — one from each city. There's little time for sightseeing as you contend with snowdrifts, exchange rates, and the constant ticking away of time. ★★★

International Publishing & Software — **ZX Scramble**. Hit enemy installations before they hit you, while maneuvering to avoid crashing into treacherous terrain. The game is slow, and the terrain repeats itself like a chase scene in a poorly animated cartoon. ★★

Melbourne House — **Combat Flight**. You fly a fighter ship over a mountain range, with enemy ships approaching. Combat Flight is fast, exciting and like numerous games on the market that have a similarity of concept. ★★★

Mindware — **Graphic Golf**. An absorbing game requiring skill and good judgement. Choose a club (wood, iron, putter) and tee off, taking wind direction and obstacles like trees, ponds and sand traps into account. The computer is the caddy. ★★★

Mindware — **The Fast One**. A "computerized filing cabinet," ideal for setting up lists of phone numbers, recipes, addresses, inventories, and any other listable information. Excellent documentation. ★★★½

Mindware — **Gulp**. You move around any of six mazes at any of nine speeds eating dots while a hunter pursues you. The keys that control movement are too close together, and after you eat the last dot you have to be captured before the game ends, but it's still enjoyable. ★★★

Mindware — **Labyrinth**. You may choose to see an overview of the complex, 3D-style maze before you wander through it, and if you get lost and give up you get to see the whole picture again. With practise, the trek becomes simpler. An interesting game but not a classic. ★★½

Orbyte Software — **Vault of the Centaurs**. For budding Jedis. The precious fuel Zykon, which your planet needs for survival, is stored deep in a vault guarded by ruthless, highly-skilled fighters known as Centaurs. Package includes a cardboard keyboard overlay that puts the six movement keys at your fingertips while covering the others. ★★★

Softsync — **Mothership**. Fast arcade-type game. Protect your Starlight Fighter from the attacking Mothership as you race down the Zarway space corrior. Excellent graphics. ★★★

Stuart Software — **Caves of Zulu**. The object of this combination word, adventure and maze game is to collect treasure hidden in a maze, but the action is slow and the programming flawed — the game falls into a loop on certain moves. It's a good idea not realizing its full potential. ★★½

Reston Publishing — **Invasion Force**. Shoot lasers through a moving force field to destroy alien ships, and watch out for bombs. The game is fun, the graphics distinctive. ★★★½

SoftSync — **TS Destroyer and Space Raid**. Fast action duo in 2K. In TS Destroyer, you must dodge or destroy floating objects as a spaceship takes potshots at you; in Space Raid, you must shoot a hole through a spaceship and hit the left foot of an alien. Neither game prints a score. ★★★½

Timex — **The Coupon Manager**. Before you go shopping, you can flip through your file of coupons and pick out the ones you need. Keeps track of amount, store and expiry date. A must for the compulsively organized. ★★★

ZX-Panding — **Checkbook, Tax and Budget Organizer**. Helps you manage your finances. Comes with a clear, four-page booklet but gives no clear instructions on how to get going after loading. An excellent save frame lets you return to the menu in case of error. ★★★

Ø/C

Letters

Computerized Grandchildren

With my third issue of *Timex Sinclair User* I know that this magazine is great. The Software Blues article hit the nail right on the head. You should have mentioned that some books written for Timex Sinclair computers are not any better. Some books and tapes are very good, but some are a disgrace. My 12-year-old grandson, for whom I set up a Timex, loves it. You ask if computers will be as common as television: of course they will. Of my five grandchildren, three want to work with computers and are waiting for me to set them up.

Salvatore Curcio
Brooklyn, New York

Thanks for the kind words and your warning regarding books. I'm sure many of our readers agree with you. It sounds like you're getting your grandchildren off to a flying start on computers — good for you. The photos you sent us of your set up (not shown here) are very impressive. Keep us informed as to how and what your grandchildren do on and with their Timex machines.

Treasure Hunt

In the program "Treasure Hunt" (Issue 4) lines 11, 13 and 15 are solid. Was there writing in the original copy? Also, can you tell me where I can order the chess game Chessmaster that you gave a four-star rating? I love your magazine and was delighted to run across it at my local store. Your reply will probably win you another subscription — mine!

Carol Barnes
Lakeland, Florida

Get that subscription order filled out 'cause here's your reply. Lines 11, 13 and 15 were solid as shown (actually 13 was a graphic character). There was no printing under the lines. We're glad you're enjoying "our magazine" and look forward to you making it "your magazine."

Wonderful Winky

I would like to commend you on your review of the Winky Board 2. It is fantastic! And the service of G. Russell Electronics is more than excellent. No sooner had I mailed my check when I found it at my door! I received my board in five days. Thanks for that review. Continue this superb tradition.

Brian K. Kautz
Camp Hill, Pennsylvania

Glad the Winky Board 2 is working for you. We agree it's a fantastic device.

For more information, circle 14 on reader service card.

For more information, circle 32 on Reader Service Card.

The articles in this series are excerpts from the book, "Programming Arcade Games," soon to be published by Reston Publishing of Reston, Virginia

Preparing For Machine Language Programming

In the second part of his series on programming arcade games, Bob Fraser discusses the programmer's most powerful tool

USING your current knowledge of BASIC as a foundation, you are now ready to learn machine language programming. Using machine language is exciting and challenging, and your programs will have a speed unmatched in BASIC. Your arcade games will be faster, more exciting, more fun to play. First you must become familiar with these concepts:

1) Organization of computer memory
2) Storage of numbers
3) Computer terminology — for example, registers and addressing

This article explains these concepts. You will need this information to take full advantage of subsequent articles in this series.

Advantages of Machine Language Programming

A good arcade game requires fast, fluid movement. This is best accomplished using machine language because instructions are executed faster and more directly than in BASIC. A high execution speed allows the computer to do more in less time, which becomes extremely important in arcade games involving a lot of movement.

BASIC is slow executing because it is actually an intermediate language between humans and computers. We can easily translate our thoughts into BASIC; then the computer translates our thoughts into machine language using a

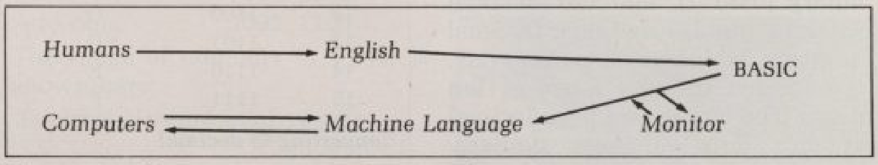


Figure 1. *How we communicate with computers.*

monitor program consisting of many machine language routines stored permanently in the computer. The monitor program interprets and executes each BASIC statement.

For a simple instruction such as PRINT AT 5,4;"EXAMPLE" the computer must (1) "Read" the PRINT command and go to the machine language routine that handles printing; (2) "Read" the AT statement and call a subroutine that sets print positions; (3) "Read" the 5,4, print coordinates, verify their validity and then set the print position (if the coordinates are invalid, execution is terminated and an error message printed); (4) Move each letter of the word EXAMPLE to the screen, one at a time.

As you can see, the computer

ILLUSTRATION • Don Kletke

does a lot of work executing a BASIC statement. Machine language bypasses the translation steps and moves the word EXAMPLE directly to the screen, thus executing fewer machine language statements and therefore reducing the overall execution time. Executing more statements in less time, machine language is the serious programmer's most important tool.

Three Number Systems

To use machine language instructions effectively, you should understand the decimal (base 10), binary (base 2), and hexadecimal (base 16) number systems. Decimal numbers, which you use every day, are formed through a set of ten digits. The binary and hexadecimal systems form numbers similarly but with a different number of digits. Figure 2 shows the digits used by each system.

Digits used by the three number systems.

Binary	0,1
Decimal	0,1,2,3,4,5,6,7,8,9
Hexadecimal	0,1,2,3,4,5,6,7,8,9, A,B,C,D,E,F

Figure 2. *The three number systems.*

Very often in machine language programming, you will convert a number from one base to another. This is made easier using the table in Figure 3.

Bits and Bytes

Bits and bytes are terms specifying units of memory. A bit is the smallest unit of memory accessible to the programmer, and contains a binary digit, either 0 or 1. 1+ is called set when it equals 1, reset when it equals 0.

Bit	*Bit*
[0] = *Reset*	[1] = *Set*

A byte is a group of eight bits. The number stored in a byte can range from 0 to 255.

Byte		*Hex*		*Decimal*
0000 0000	=	00	=	0
0000 0001	=	01	=	1
0000 0010	=	02	=	2
•		•		•
•		•		•
•		•		•
1111 1110	=	FE	=	254
1111 1111	=	FF	=	255

Counting to 15 in the three systems, with conversion examples.

Decimal	*Binary*	*Hexadecimal*
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F

Converting to decimal

1. Converting base 2 to decimal:

$$1111 = 1 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 = 8 + 4 + 2 + 1 = 15$$

2. Converting base 16 to decimal:

$$FF = 15 \times 16^1 + 15 \times 16^0 = 240 + 15 = 255$$

Converting binary to hexadecimal

Computers use the base 2 system (binary) for storing numbers in memory. Any number from 0 to 255 is stored as a group of eight binary digits. These binary digits can be converted to hexadecimal quite easily using the chart.

11111111 = [1111] [1111] = FF

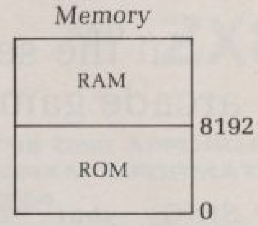
Each group of four binary digits represent one hexadecimal digit.

1. 10100111 = [1010] [0111] = A7
2. 01001111 = [0100] [1111] = 4F
3. 10001010 = [1000] [1010] = 8A

Figure 3. Conversions.

Organization of Computer Memory

The memory of the computer is divided into two main areas, ROM (read only memory), and RAM (read access memory). ROM is located first in memory. The start of ROM is at address location 0 and is 8192 bytes in length.

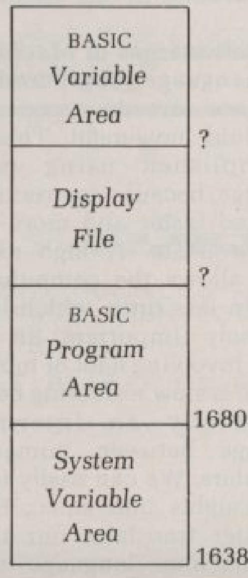


ROM is a permanent area of memory, the location of all the monitor routines.

RAM is an area of memory free to the programmer, where the BASIC program, BASIC variables and display area are located.

ROM is permanent in memory because it contains all the monitor routines that allow the computer to function. This information is retained in the computer even when the computer is unplugged.

RAM is not permanent in memory, and its information is lost every time the power is turned off. There are four sections of RAM that are very important to machine language programming. They are shown here:



The system variable area stores variables used by the monitor routines. The location of these variables in memory is always the same. Some of them supply useful information to the programmer such as the address location of the display file or the BASIC variable area.

The BASIC program area is the area of memory where the BASIC program is stored. Its starting address is always 16509, but its final address varies depending on the size of the BASIC program.

The character codes for each printable location on the screen are stored in the display file. The address location of the display file varies depending on the size of your BASIC program.

Space is reserved in the BASIC variable area for each variable used by your BASIC program. The address location of this area also depends on the size of your BASIC program.

Representation of Numbers

A byte of memory is made up of eight bits. Together, those eight bits can form a number from 0 to 255 decimal. For numerical purposes, such as adding and subtracting, the computer uses the leftmost bit to indicate the sign of the number.

0 = positive
1 = negative

```
BIT  7  6  5  4  3  2  1  0
   --------------------------
   :  :  :  :  :  :  :  :  :
   :  :  :  :  :  :  :  :  :
   --------------------------
```

With only seven bits representing a number, the largest positive number is 127 (64 + 32 + 16 + 8 + 4 + 2 + 1).

+127 = 7F = 0111 1111

Negative numbers are represented differently using notation called two's complement. To form the two's complement of a number, reverse all bits in the number — 0s become 1s, 1s become 0s — and add 1 to the number.

For example, −100 is changed to 2's complement like this:

Binary:	0110 0100
Reverse bits:	1001 1011
Add one:	1001 1100

The range of numbers in a byte is shown here:

Decimal		*Hexadecimal*		*Two's Complement*	
+127	=	7F	=	0111	1111
+126	=	7E	=	0111	1110
+125	=	7D	=	0111	1101
:		:			:
:		:			:
+2	=	02	=	0000	0010
+1	=	01	=	0000	0001
0	=	00	=	0000	0000
−1	=	FF	=	1111	1111
−2	=	FE	=	1111	1110
:		:			:
:		:			:
−125	=	83	=	1000	0011
−126	=	84	=	1000	0010
−127	=	85	=	1000	0001
−128	=	86	=	1000	0000

Range of numbers in a byte or register:

+127 <−−> −128

The example below shows how the computer adds two numbers.

```
  0000 1010   (+10)
+ 0001 1000   (+24)
-----------------
= 0010 0010   (+34)
```

Registers

A register is a single byte location inside the computer. The T/S1000,ZX81 contains 24 registers, eight of which are frequently used in machine language instructions. These eight registers are lettered A to F.

A register holds a number from 0 to 255. Machine language instructions manipulate the data in these registers in various ways. Here are three examples of machine language instructions using the registers:

Instruction	*Explanation*
LD A,4	— *Load the A register with 4 (let A = 4)*
LD B,A	— *Load the B register with the contents of the A register*
INC A	— *Increment the value in the A register by one*

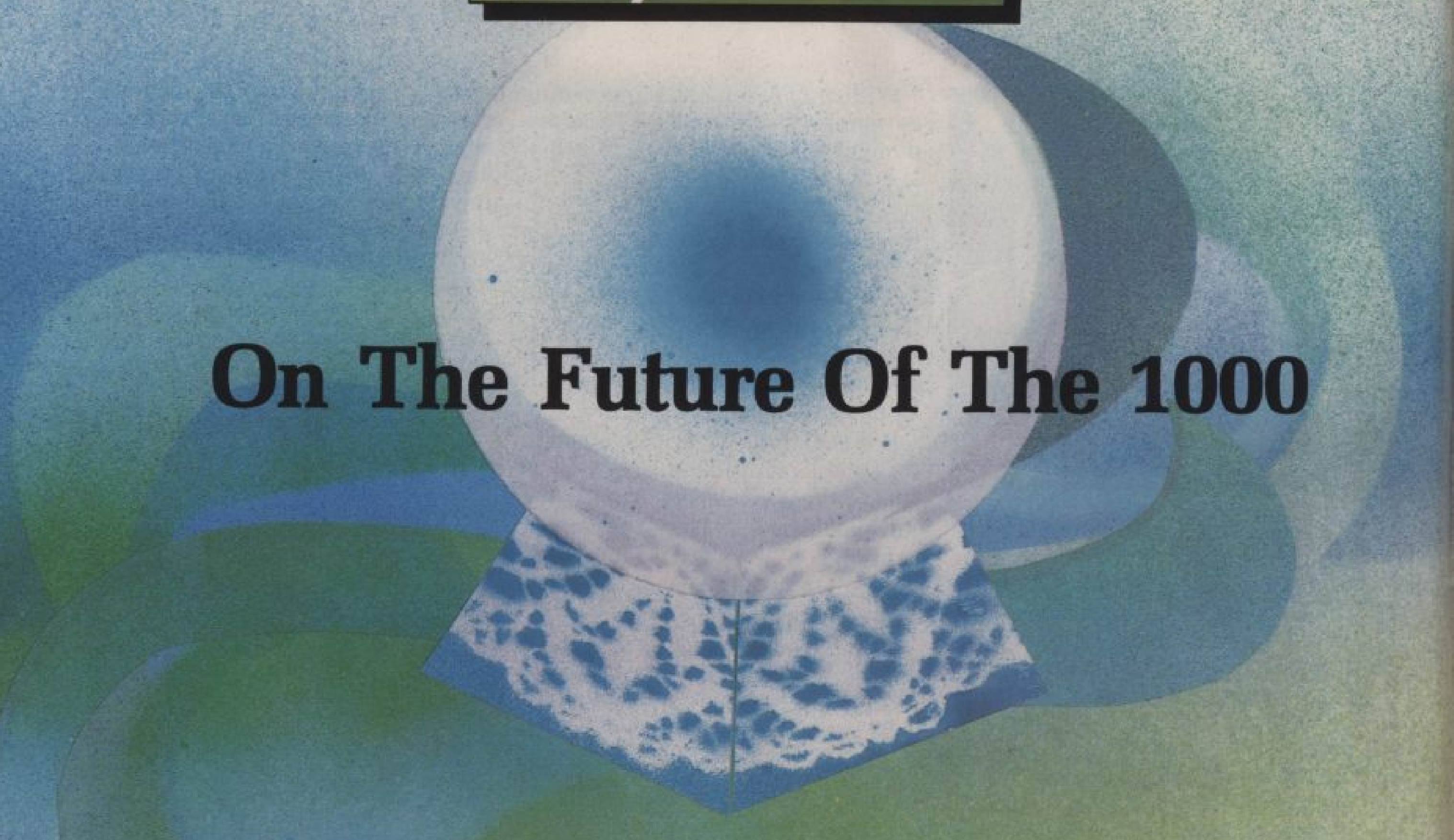
Some instructions use register pairs. A register pair can be used to store large numbers such as addresses. A number stored in a register pair can range from 0 to 65536.

Register Pairs

	High	*Low*
AF =	A	F
BC =	B	C
DE =	D	E
HL =	H	L

A register contains 16 bits of information. The rightmost eight bits comprise the low order register.

continued on page 48



On The Future Of The 1000

IN THE PAST we have "gone on" a bit about how Timex has chosen to introduce new machines and its lack of support for present ones. We have suggested that the amount of confusion created was not needed — in fact that it was quite detrimental to all involved . . . both users and producers. Many of you have written in support of our position. While no accurate count has been kept, the number of writers who agree with us has been overwhelming.

Our aim has been not to create more problems for Timex but rather to act as a conduit between Timex and its customers, voicing the concerns we had heard. Now that the new machines are starting to be shipped, some of those concerns become immediate needs. I would like to quote from two letters that show the range of opinions being sent to us.

Thomas B. Woods of Jefferson, New Hampshire, writes: "Your comments about 'Computers and Confusion' in TSU 1:4 parallel my own thoughts very closely. It seems that no sooner do I have the products and the competition 'figured out' than the rug is pulled out from under me from a different direction: the computer I work with changes. A new computer may come out that's ten, 100 or even 1,000 times better than the T/S1000, but that doesn't make our computers any less valuable. I agree wholeheartedly with you that the future of the T/S1000 will be in the area of dedicated tasks.

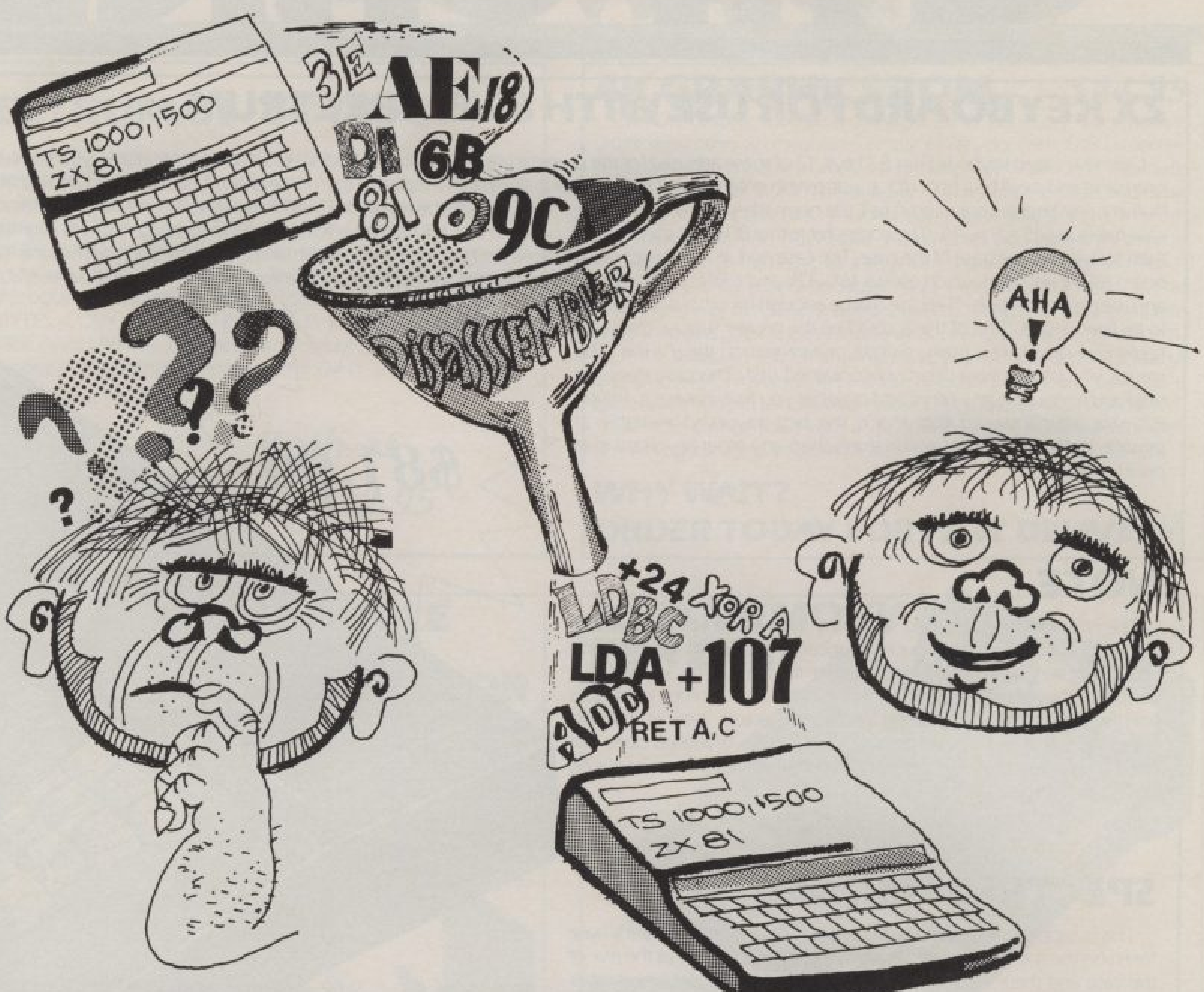
"In my business the satisfied customer will return to purchase my new works. I value their judgement and if they tell me they want more and better software for the 1000, then by George, that's exactly what I intend to give them."

Mr. N.H. Nucho of Arcadia, California, writes: "In your last two issues you have mentioned the new 2000 series without giving any 'guidance' as to how users of the 1000 will be able to use their present software with the new machines. If we buy the 2000, will all the money we spent on software go down the drain? Do we have to rekey all those programs into the 2000? That could take a lifetime! How this problem is approached I am sure could determine the success or failure of the new 2000 series in the marketplace."

Producers and users both face problems because Timex cannot seem to provide information to those who need it on a regular and consistent basis. There are a lot of users out there who want to remain loyal to Timex, if only Timex would let them!

As to us here at *Timex Sinclair User*, we certainly guarantee that our magazine will always reflect the fact that the 1000 and 1500 are out there. We will continue to publish articles, software reviews, projects and program printouts for those machines.

In this issue we take a good look at joysticks through the hardware review and our project. Bob Fraser's "Programming Arcade Games" continues with the second of five parts. Different file programs are looked at in Focus and of course Nigel Clark continues to keep us informed as to what's happening in the U.K. In all, another great issue! 0/0



DISA-Z: A Z80 Disassembler

SINCLAIR'S 8K ROM for the T/S1000 is generally easy to poach upon in machine code. There is almost always a handle — an address you can call — to get the better part of most operating-system functions done for you. Even the floating-point operations are easily accessed. All you need is a good labelling disassembler and/or an annotated listing for the Sinclair ROM, and of course a little machine or assembly programming savvy.

The T/S2000 will be a much more complex machine, with 24K of ROM on board, memory bank switching, and several new BASIC functions. The system includes a function dispatcher that can be called by machine programs to access most BASIC and Input/Output functions, thus removing the burden of knowing your ROM in order to program in Z80 code. Presumably, going through the function dispatcher will give the machine programmer some assurances that future changes in the ROM will not destroy his programs, as the switch to the new ZX81 ROM did back in 1981. Good programming practice will dictate using the function dispatcher for nearly all system calls.

Nevertheless, as a machine programmer, I still want to know my way around a computer's ROM before I decide whether to fall in love with it, much in the way a hardware hacker wants a schematic. I want an instant disassembler that I can key into a new T/S2000 without the usual wait for the assembly language tools and books to appear. The cure for that is a BASIC program that can be keyed into a brand new 2000 on the first day to churn out an assembly

dk'tronics

ZX KEYBOARD FOR USE WITH 80/81 SPECTRUM

Our new cased keyboard has 52 keys, 12 of these are used for the numeric pad. The numeric pad offers some useful features, you can cursor with one hand and it will be a boon for anyone who enters a lot of numeric data. The pad is a repeat of the 1-9 keys plus it has a full stop and a shift key. The numeric pad keys are coloured in red, the normal keyboard keys are grey, with the case being black which makes the whole thing very attractive. The case measures 15 × 9 × 2½. The computer (either 80/81 or spectrum) fits neatly inside. You will have to remove the computer from its original case, it is then screwed to the base of the case. The case had all the bosses already fitted and the screw holes are marked. Also fitted inside the case is a mother board (81 model only) which allows 16K, 32K and 64K to be fitted in the case. All the connections are at the rear of the case i.e. Power, Mic, Ear, T.V. and the expansion port. The case is large enough for other add ons also to be fitted inside. One of these could be the power supply, then you could very quickly fit a mains switch, or a switch on the 9V line. This means you have a very smart self-contained unit. This case does not stop you from using any other add-ons that you may have e.g. Printer etc. We are convinced that this is the best keyboard available at present. It offers more keys and features than any other keyboard in its price range.

$81.00

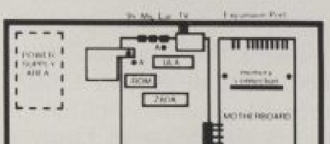
NOTE

The case can be purchased separately with the keyboard aperture uncut, so if you have one of our early uncased keyboards, or in fact any other suppliers' keyboards, these could be fitted. The keyboard is connected to your computer by a ribbon cable and this has connectors fitted which simply push into the Sinclair connectors. It is a simple two minute job and requires no electronic skills. This keyboard does not need any soldering. Please specify on order whether you require the ZX 81 or Spectrum case.

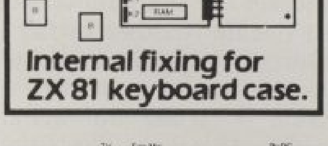
SPECTRUM MODEL

This is supplied with spectrum legends, and a slightly different base for fitting the spectrum inside, again all the connectors are at the rear of the case and there is plenty of room for the power supply (and other add-ons). Should you want to change, we can supply both the Spectrum legends and details of updating your case which will enable modification from the ZX 81 to spectrum. PLEASE specify on your order whether you require the ZX 81 or spectrum case.

KEYBOARD (SPECTRUM/81)



Internal fixing for ZX 81 keyboard case.



Internal fixing for spectrum keyboard case.

SPECTRUM LIGHT PEN

The pen enables you to produce high resolution drawings on your own television set, saving a vast amount of time over using basic programming statements, such as Plot, Draw etc. You can erase, modify and save drawings and it comes complete with software program. The superb light pen is available from D.K. tronics.

$49.95

SPECTR

16K Memory Ex
The 16K uses 4116
Static Ram and
using the Static R
speed low power
the computer. A
which comes to y
where. Position i

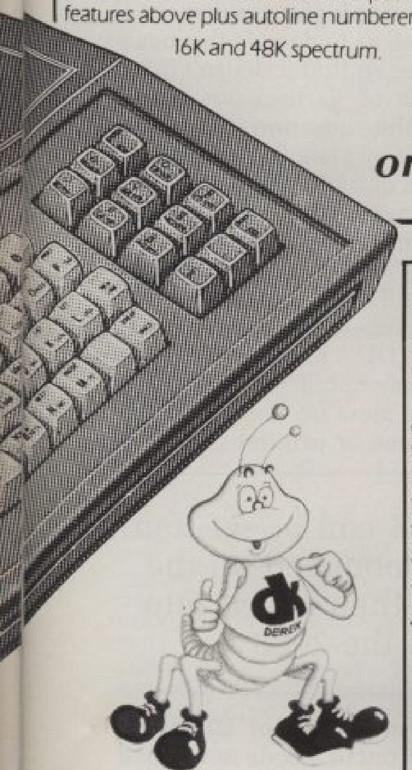
64K Memory Ex
All the above inf
advantage lies in
56K of usable me
the use of other a
8192-65536. The
Spectrum Memo
Upgrade your Spe
it is simply slipped
are supplied, and
time. The fitting re
same as Sinclair's

ZX 80~81 Spectrum HARDWARE

SPECTRUM/81 TOOLKIT

This is the toolkit which won acclaim in the feature in the August 1982 issue (pages 29 and 30) of Sinclair User. "It is the most impressive program, fast in execution with clear and full instructions...it stands out from the rest of the field." The ZXED is a powerful editor for use on the expanded ZX81. It is intended for use by the serious BASIC programmer and offers several useful and time saving features most helpful during all stages of program development. The facilities provided are as follows: ALTER, BYTES, COPY, DELETE, FIND, HELP, INSERT, KEEP, MOVE, RENUMBER AND VERIFY. The Spectrum Toolkit contains most of the features above plus autoline numberer and append, and will run in the 16K and 48K spectrum.

Both at only $12.95



4K GRAPHICS ROM *$44.95*

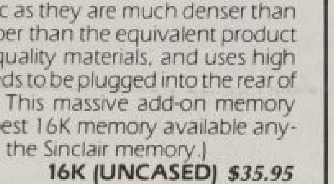
The DK Graphic module is our latest ZX 81 accessory. This module unlike most other accessories fits neatly inside your computer under the keyboard. The module comes ready built, fully tested and complete with a 4K graphic ROM. This will give you an unbelievable 448 extra pre-programmed graphics, your normal graphic set contains only 64. This means that you now have 512 graphics and with their inverse 1024. This now turns the 81 into a very powerful computer with a graphic set rarely found on larger more expensive machines. In the ROM are lower case letters, bombs, bullets, rockets, tanks, a complete set of invaders graphics and that only accounts for about 50 of them, there are still about 400 left (that may give you an idea as to the scope of the new ROM). However, the module does not finish there, it also has a spare holder on the board which will accept a further 4K of ROM/RAM. This holder can be fitted with a 1K/2K/RAM and can be used for user definable graphics so you can create your own custom character sets.

WHY WAIT?
ORDER TODAY FOR FAST DELIVERY

FLEXIBLE RIBBON CONNECTOR

If you have ever had whiteouts or system crashes this could be the answer. It stops the movement between the computer and the RAM expansion, it is supplied with a ribbon, 6 inches long, with a male connector at one end and a female at the other, at only

$35.95



16/64 MEMORY FOR ZX 81

16K Memory	*$41.95*
16K (uncased)	*$35.95*
64K Memory	*$135.00*
64K (uncased)	*$132.00*

The above illustration shows the casing for the 16K or massive 64K.

RUM MEMORY

pansion *$41.50*

Dynamic Ram Chips. We use the dynamic as they are much denser than cupy less space. They are also much cheaper than the equivalent product m. The Ram is manufactured with high quality materials, and uses high Rams. It is supplied ready-built and only needs to be plugged into the rear of the components are fitted into holders. This massive add-on memory ou fully assembled and tested is the cheapest 16K memory available any- memory from 16384 to 32768. (Same as the Sinclair memory.)

16K (UNCASED) *$35.95*

pansion *$135.00*

rmation on the 16K also applies to the 64K Memory Expansion, but the he 64K giving nearly FOUR times the memory. This advanced model has nory. In addition, the block from 8K to 16K can be switched out to enable dd-ons. The graphics ROM is to be used in this area. Position in Memory: Block From 8192-16384 is switchable.

64K (UNCASED) *$132.00*

ry Expansion *$54.00*

ctrum to 48K of user Ram. The Spectrum memory expansion is simple to fit, inside the case, and then only requires plugging in. Full fitting instructions the only tool you will need is a screwdriver and just two minutes of your quires no electronic skills. Position in memory from 32768 to 65536. (The upgrade to 48K).

Please state type of machine, which Rom memory size, quantity and place when ordering

Please send me . . . . @ $ . . . .

Please send me . . . . @ $ . . . .

Please send me . . . . @ $ . . . .

Please add on $9.00 for P/P

In enclose cheque/P.O. payable to DK Tronics total . . . .

or debit my Access/Barclaycard / *Mastercard*

Signature . . . .

Name . . . .

Address . . . .

Send to DK Tronics Ltd, Unit 2, Shire Hill Ind Est, Saffron Walden, Essex CB11 3AQ, England.
Tel: (0799) 22036 (trade enquiries please telex 81653 attn. DK)

dk'tronics

New colour brochure now available, send $1.00 for quick return

For more information, circle 15 on Reader Service Card.

listing of that new ROM on the T/S2040 printer. The result was DISA-Z (listing 1).

BASIC is painfully slow for the job, but to get a printed listing speed hardly matters; the program can run while its user sleeps. Except for the difference in character codes, T/S1000 BASIC is essentially a subset of the 2000 version, so DISA-Z could be written and debugged on the 1000 and then keyed (not loaded) into the 2000, with a reasonable probability of running the first time. Writing a compatible program meant generally avoiding the CHR$ function so that the differences between the 2000's ASCII and the 1000's character code would not cause chaos in the program. PEEKs and POKEs of system variables also have to be avoided, because it appears that the 2000 will start its 6K display file just where the system variables of the 1000 begin.

The variable TS2 in line 190 equals 1 for the T/S2068 and 0 for the T/S1000 or 1500

Program Structure

The structure of DISA-Z is shown schematically in Figure 1. Since the data are stored in listed strings, you can run the program. The first and only entry you need to make is the decimal starting address of the code you want to disassemble. DISA-Z then builds a single string (H$) consisting of the address in hexadecimal (lines 350-410), the bytes of code for one instruction (lines 3630-3660), and the Z80 mnemonic for that instruction (lines 970-3490). The string can then be sent either to the screen, your printer or both, and DISA-Z loops back to build another H$ for the next instruction. You terminate the program with the BREAK key.

There are two main tasks in the building of H$. The first, after the address is translated to hex, is to determine the length in bytes of the instruction. One way to do this is to enter a string that contains a number from one to four for each of the possible 256 codes, and then to slice the string to get the instruction length. I tried both that method and the logical sort in lines 420-960. The look-up method, which is faster in machine code, turned out to be about 10 percent slower, so I decided to keep the logical sort, which is a bit longer, but much less subject to typing errors on entry.

```
0292 DDE1     POP IX
0294 FD4E28   LD C,(IY+28)
0297 FDCB3B7E BIT 7,(IY+3B)
029B 280C     JR Z 02A9
029D 79       LD A,C
029E ED44     NEG
02A0 3C       INC A
02A1 08       EX AF,AF"
02A2 D3FE     OUT FE,A
02A4 E1       POP HL
02A5 D1       POP DE
02A6 C1       POP BC
02A7 F1       POP AF
02A8 C9       RET
02A9 3EFC     LD A,FC
02AB 0601     LD B,01
02AD CDB502   CALL 02B5
02B0 2B       DEC HL
02B1 E3       EX(SP),HL
02B2 E3       EX(SP),HL
02B3 DDE9     JP(IX)
02B5 ED4F     LD R,A
02B7 3EDD     LD A,DD
02B9 FB       EI
02BA E9       JP(HL)
02BB 21FFFF   LD HL,FFFF
02BE 01FEFE   LD BC,FEFE
02C1 ED78     IN A,(C)
02C3 F601     OR 01
02C5 F6E0     OR E0
02C7 57       LD D,A
02C8 2F       CPL
02C9 FE01     CP 01
02CB 9F       SBC A,A
02CC B0       OR B
02CD A5       AND L
02CE 6F       LD L,A
02CF 7C       LD A,H
02D0 A2       AND D
02D1 67       LD H,A
02D2 CB00     RLC B
02D4 ED78     IN A,(C)
02D6 38ED     JR C 02C5
02D8 1F       RRA
02D9 CB14     RL H
02DB 17       RLA
02DC 17       RLA
02DD 17       RLA
02DE 9F       SBC A,A
02DF E618     AND 18
02E1 C61F     ADD A,1F
02E3 322840   LD (4028),A
02E6 C9       RET
```

The sort for instruction length is simplified by some regularities in the Z80 instruction set. For example, all codes between 63 and 192 are for instructions that are one byte long. It is the Z80 instruction prefixes, particularly the ED prefix, which complicates things, because they only have meaning with certain codes.

A disassembler has to be able to deal with meaningless sequences of code as well as to decode the meaningful ones, so that it doesn't crash in a patch of garbage or data. I handled the question of meaningless EDs by assigning an instruction length of one to them, whereas the meaningful codes are all either two or four bytes long. DISA-Z then lists "DATA" for the meaningless instructions. I also sorted out any multiple successive occurences of FD and DD prefixes, mainly because they could run H$ to any length and mess up the output format to screen or printer.

DISA-Z can be written and debugged on the 1000, then keyed into the 2068

When a single FD or DD prefix occurs in front of a code to which it does not apply, the prefix normally has no effect on the instruction, so the prefix is printed but the code is translated as it would be without the prefix. As far as I know, this is the way the Z80 operates, with the probable exception of the fabled "undocumented" prefixed instruction.

I've tested most of the undocumented instructions on my single-stepper and discovered that FD or DD will change an eight-bit register operation on either the H or L registers to the low (L) or high (H) eight bits of IX or IY. I have omitted these because they appear to me to be vastly impractical, given the penalty of double code

```
  10 REM ************************
            DISA Z
  20 REM ************************
         A Z80 DISASSEMBLER
         FOR TS COMPUTERS
  30 REM ************************
  40 REM ************************
         COPYRIGHT,1983
         RAY KINGSLEY
  50 REM ************************
         RUN AND ENTER START
         ADDRESS IN DECIMAL
  60 REM ************************
  70 REM ************************
  90 REM DATA STRINGS
 100 LET O$="ADDADCSUBSBCANDXORO
R CP "
 110 LET D$="BCDEHLSPAFIXIY"
 120 LET B$="BITRESSET"
 130 LET C$="RSLRC AL"
 140 LET R$="BCDEHLXA"
 150 LET E$="NZZ NCC POPEP M "
 160 LET F$="RLCARRCARLA RRA DAA
CPL SCF CCF "
 170 LET G$="LDCPINOT"
 180 REM FOR TS2000 SET TS2=1
 190 LET TS2=0
 200 LET NCOD=28+20*TS2
 220 GOTO 3500
 230 REM OCTAL DIGITS
 240 FOR K=0 TO 3
 250 IF X0>=K*64 AND X0<(K+1)*64
THEN LET X=X0-K*64
 260 NEXT K
 270 LET C=INT (X/8)
 280 LET T=X-C*8
 290 RETURN
 295 REM BYTE HEXER
 300 LET X=PEEK J
 310 LET Y=INT (X/16)
 320 LET Z=X-Y*16
 330 LET H$=H$+CHR$ (NCOD+Y+7*(Y
>9)*TS2)+CHR$ (NCOD+Z+7*(Z>9)*TS
2)
 340 RETURN
 350 REM ADDRESS HEXER
 360 LET X=INT (J/256)
 370 GOSUB 310
 380 LET X=J-256*X
 390 GOSUB 310
 400 LET H$=H$+" "
 410 RETURN
 420 REM GET INSTRUCTION LENGTH
 430 LET L=0
 440 IF X<>221 AND X<>253 THEN G
OTO 540
 450 IF L THEN RETURN
 460 GOSUB 480
 470 GOTO 440
 480 LET L=1
 490 LET J=J+1
 500 GOSUB 300
 510 LET J=J-1
 520 LET H$=H$( TO LEN H$-2)
 530 RETURN
 535 REM I-LENGTH FOR PREFIXES
 540 IF L THEN LET L=L+((X>51) A
ND (X<55) OR X=203)
 550 IF X=203 THEN GOTO 940
 560 IF X<>237 THEN GOTO 710
 570 GOSUB 480
 580 IF X<64 OR X>188 THEN RETUR
N
 590 IF X<160 AND X>123 THEN RET
URN
 600 IF X<124 THEN GOTO 630
 610 IF X-INT (X/8)*8<5 THEN GOT
O 950
 620 RETURN
 630 IF X=78 OR X=102 OR X=110 O
R X=112 OR X=113 OR X=118 OR X=1
19 THEN RETURN
 640 LET W=X-67
 650 IF NOT (W-INT (W/8)*8) THEN
 GOTO 920
 660 LET W=X-76
 670 IF W>=0 AND NOT (W-INT (W/8
)*8) THEN RETURN
 680 LET W=X-85
 690 IF W>=0 AND NOT (W-INT (W/8
)*8) THEN RETURN
 700 GOTO 950
 705 REM WITHOUT PREFIXES
 710 IF X<64 OR X>191 THEN GOTO
740
 720 IF L THEN LET L=L+(X-8*INT
(X/8)=6)
 730 GOTO 950
 740 IF X>191 THEN GOTO 830
 750 IF Z=1 THEN GOTO 920
 760 LET W=X-34
 770 IF W>=0 AND NOT (W-INT (W/8
)*8) THEN GOTO 920
 780 LET W=X-6
 790 IF W>=0 AND NOT (W-INT (W/8
)*8) THEN GOTO 940
 800 LET W=X-16
 810 IF W>=0 AND NOT (W-INT (W/8
)*8) THEN GOTO 940
 820 GOTO 950
 830 IF X=195 OR X=205 THEN GOTO
 920
 840 IF X=211 OR X=219 THEN GOTO
 940
 850 LET W=X-194
 860 IF NOT (W-INT (W/8)*8) THEN
 GOTO 920
 870 LET W=W-2
 880 IF NOT (W-INT (W/8)*8) THEN
 GOTO 920
 890 LET W=W-2
 900 IF NOT (W-INT (W/8)*8) THEN
 GOTO 940
 910 GOTO 950
 920 IF L=2 THEN GOTO 940
 930 LET L=L+1
 940 LET L=L+1
 950 LET L=L+1
 960 RETURN
 970 REM MNEMONICS
 980 LET H$=H$+" "
 990 IF LEN H$<14 THEN GOTO 980
1000 IF X0=118 THEN GOTO 1290
1010 GOSUB 230
1020 IF X0<64 OR X0>191 THEN GOT
O 1400
1030 IF X0>127 THEN GOTO 1310
1035 REM EIGHT-BIT REG LDS
1040 LET P$=R$(C+1)
1050 LET Q$=R$(T+1)
1060 IF P$="X" OR Q$="X" THEN GO
SUB 1170
1070 LET H$=H$+"LD "+P$+","+Q$
1080 RETURN
1085 REM ADD POINTER REGS TO H$
1090 IF NOT DD THEN GOTO 1120
1100 LET H$=H$+"IX"
1110 RETURN
1120 IF NOT FD THEN GOTO 1150
1130 LET H$=H$+"IY"
1140 RETURN
1150 LET H$=H$+"HL"
1160 RETURN
1165 REM CHECK FOR FD OR DD
1170 IF FD OR DD THEN GOTO 1210
1180 IF P$="X" THEN LET P$="(HL)
"
1190 IF Q$="X" THEN LET Q$="(HL)
"
1200 RETURN
1205 REM ADD INDEX REGS TO H$
1210 IF FD THEN LET W$="(IY+"
1220 IF DD THEN LET W$="(IX+"
1230 IF P$="X" THEN LET P$=W$+H$
(10 TO 11)+")"
1240 IF Q$="X" THEN LET Q$=W$+H$
(10 TO 11)+")"
1250 RETURN
1260 GOSUB 1170
1270 LET H$=H$+P$+","+R$(1+T)
1280 RETURN
1290 LET H$=H$+"HLT"
1300 RETURN
1310 REM ARITH/LOGIC OPS
1320 LET I=C*3+1
1330 LET Q$=R$(T+1)
1340 GOSUB 1170
1350 LET I$=" "
1360 IF I<7 OR I=10 THEN LET I$=
" A,"
1370 IF I>18 THEN LET I$=""
1380 LET H$=H$+O$(I TO I+2)+I$+Q
$
1390 RETURN
1395 REM 1ST AND 4TH GROUPS
1400 IF X0=203 THEN GOTO 3170
1410 IF X0=253 OR X0=221 THEN GO
TO 3420
1420 IF X0>191 THEN GOTO 1570
1425 REM 1ST GROUP
1430 IF T=1 THEN GOTO 2020
1440 IF T=3 THEN GOTO 2240
1450 IF T=4 OR T=5 THEN GOTO 230
0
1460 IF T=2 THEN GOTO 2370
1470 IF NOT T THEN GOTO 2080
1480 IF T<>6 THEN GOTO 1540
1490 LET Q$=R$(C+1)
1500 GOSUB 1170
1510 LET W=4*(FD OR DD)
1520 LET H$=H$+"LD "+Q$+","+H$(8
+W TO 9+W)
1530 RETURN
1540 LET W=4*C+1
1550 LET H$=H$+F$(W TO W+3)
1560 RETURN
1570 REM DIRECT ARITHMETIC
1580 IF T<>6 THEN GOTO 1660
1590 LET I=3*C+1
1600 LET I$=" "
1610 IF I<7 OR I=10 THEN LET I$=
" A,"
1620 IF I>18 THEN LET I$=""
1630 LET H$=H$+O$(I TO I+2)+I$
1640 LET H$=H$+H$(8 TO 9)
1650 RETURN
1660 REM 4TH GROUP
1670 IF X0=237 THEN GOTO 2700
1680 IF T=1 AND NOT (C-INT (C/2)
*2) THEN GOTO 1920
```

Listing 1. *The DISA Z program.*

```
1690 IF T=7 THEN GOTO 1990
1700 IF T=5 AND NOT (C-INT (C/2)*2) THEN GOTO 1940
1710 LET FL=0
1720 IF X0=205 OR T=4 THEN GOSUB 1830
1730 IF X0=195 OR T=2 THEN GOSUB 1800
1740 IF X0=201 OR NOT T THEN GOSUB 1870
1750 IF NOT T OR T=2 OR T=4 THEN GOSUB 1890
1760 IF T=3 AND NOT FL THEN GOTO 2510
1770 IF T=1 AND NOT FL THEN GOTO 2610
1780 IF FL THEN GOSUB 1850
1790 RETURN
1795 REM MAIN CONDITIONALS
1800 LET H$=H$+"JP "
1810 LET FL=1
1820 RETURN
1830 LET H$=H$+"CALL "
1840 GOTO 1810
1850 LET H$=H$+H$(10 TO 11)+H$(8 TO 9)
1860 RETURN
1870 LET H$=H$+"RET "
1880 GOTO 1810
1890 LET H$=H$+E$(2*C+1 TO 2*(C+1))
1900 IF NOT (C=1 OR C=3 OR C>5) THEN LET H$=H$+" "
1910 RETURN
1915 REM GROUP 4:C EVEN,T=1 OR 5
1920 LET H$=H$+"POP "
1930 GOTO 1950
1940 LET H$=H$+"PUSH "
1950 IF C=6 THEN LET C=8
1960 IF C=4 THEN LET C=C+6*(FD OR DD)
1970 LET H$=H$+D$(1+C TO 2+C)
1980 RETURN
1985 REM GROUP 4:T=7
1990 LET W=INT (C/2)
2000 LET H$=H$+"RST "+STR$ (W)+STR$ (8*(C-2*W))
2010 RETURN
2020 REM GROUP 1:T=1
2030 IF C-2*INT (C/2) THEN GOTO 2060
2040 LET H$=H$+"LD "+D$(C+1 TO C+2)+","
2050 GOTO 1850
2060 LET H$=H$+"ADD HL,"+D$(C TO C+1)
2070 RETURN
2080 REM GROUP 1:T=0 (JRS)
2090 IF C=0 THEN LET H$=H$+"NOP"
2100 IF C=1 THEN LET H$=H$+"EX AF,AF"""
2110 IF C=2 THEN LET H$=H$+"DJNZ "
2120 IF C>1 THEN GOTO 2140
2130 RETURN
2140 IF C>2 THEN LET H$=H$+"JR "
2150 LET W=C*2-7
2160 IF C>3 THEN LET H$=H$+E$(W TO W+1)
2170 IF NOT (C-2*INT (C/2)) THEN LET H$=H$+" "
2180 LET W=(PEEK (J-1) AND (PEEK (J-1)<128))+((PEEK (J-1)-256) AND (PEEK (J-1)>127))
2190 LET J0=J
2200 LET J=J+W
2210 GOSUB 350
2220 LET J=J0
2230 RETURN
2240 REM GROUP 1:T=3
2250 LET Q$="INC "
2260 IF C-2*INT (C/2) THEN LET Q$="DEC "
2270 LET W=2*INT (C/2)+1
2280 LET H$=H$+Q$+D$(W TO W+1)
2290 RETURN
2300 REM GROUP 1:T=4 OR 5
2310 LET Q$=R$(C+1)
2320 GOSUB 1170
2330 LET P$="INC "
2340 IF T-2*INT (T/2) THEN LET P$="DEC "
2350 LET H$=H$+P$+Q$
2360 RETURN
2370 REM GROUP 1:T=2
2380 LET H$=H$+"LD "
2390 IF NOT (C-2*INT (C/2)) THEN GOTO 2460
2400 IF C=5 THEN GOSUB 1090
2410 IF C=5 THEN GOTO 2430
2420 LET H$=H$+"A"
2430 IF C>3 THEN LET H$=H$+",("+H$(10 TO 11)+H$(8 TO 9)+")"
2440 IF C<4 THEN LET H$=H$+",("+D$(C TO C+1)+")"
2450 RETURN
2460 IF C<4 THEN LET H$=H$+"("+D$(C+1 TO C+2)+"),"
2470 IF C>3 THEN LET H$=H$+"("+H$(10 TO 11)+H$(8 TO 9)+"),"
2480 IF C=4 THEN GOTO 1090
2490 LET H$=H$+"A"
2500 RETURN
2510 REM GROUP 4:T=3
2530 IF C=2 THEN LET H$=H$+"OUT "+H$(8 TO 9)+",A"
2540 IF C=3 THEN LET H$=H$+"IN A,"+H$(8 TO 9)
2550 IF C=4 THEN LET H$=H$+"EX(SP),HL"
2560 IF C=5 THEN LET H$=H$+"EX DE,HL"
2580 IF C=6 THEN LET H$=H$+"DI"
2590 IF C=7 THEN LET H$=H$+"EI"
2600 RETURN
2610 REM GROUP 4:T=1
2620 IF C=3 THEN LET H$=H$+"EXX"
2630 IF C=5 THEN LET H$=H$+"JP("
2640 IF C=7 THEN LET H$=H$+"LD SP,"
2650 IF C<5 THEN RETURN
2660 GOSUB 1090
2670 IF C=7 THEN RETURN
2680 LET H$=H$+")"
2690 RETURN
2700 REM ED PREFIXES TO 3610
2710 IF L=1 THEN GOTO 3460
2720 LET X0=PEEK (J-L+1)
2730 GOSUB 230
2740 IF X0<160 OR X0>187 THEN GOTO 2850
2750 IF T>3 THEN RETURN
2760 IF T=3 AND C<6 THEN LET Q$="OUT"
2770 IF T=3 AND C<6 THEN GOTO 2800
2780 LET W=2*T+1
2790 LET Q$=G$(W TO W+1)
2800 IF (C-2*INT (C/2)) THEN LET Q$=Q$+"D"
2810 IF NOT (C-2*INT (C/2)) THEN LET Q$=Q$+"I"
2820 IF C>5 THEN LET Q$=Q$+"R"
2830 LET H$=H$+Q$
2840 RETURN
2850 IF X0<64 OR X0>123 THEN RETURN
2860 IF T>1 THEN GOTO 2910
2870 LET Q$=R$(C+1)
2880 IF T THEN LET H$=H$+"OUT(C),"+Q$
2890 IF NOT T THEN LET H$=H$+"IN "+Q$+",(C)"
2900 RETURN
2910 LET W=INT (C/2)
2920 IF T<>2 THEN GOTO 2960
2930 IF C-2*W THEN LET H$=H$+"ADC HL,"+D$(2*W+1 TO 2*W+2)
2940 IF NOT (C-2*W) THEN LET H$=H$+"SBC HL,"+D$(2*W+1 TO 2*W+2)
2950 RETURN
2960 IF T<>3 THEN GOTO 3020
2970 IF C-2*W THEN GOTO 3000
2980 LET H$=H$+"LD("+H$(12 TO 13)+H$(10 TO 11)+"),"+D$(2*W+1 TO 2*W+2)
2990 RETURN
3000 LET H$=H$+"LD "+D$(2*W+1 TO 2*W+2)+",("+H$(12 TO 13)+H$(10 TO 11)+")"
3010 RETURN
3020 IF T<>7 THEN GOTO 3120
3030 IF C>3 THEN GOTO 3090
3040 IF C=2*W THEN LET Q$="I"
3050 IF C<>2*W THEN LET Q$="R"
3060 IF C>1 THEN LET H$=H$+"LD A,"+Q$
3070 IF C<2 THEN LET H$=H$+"LD "+Q$+",A"
3080 RETURN
3090 IF C=4 THEN LET H$=H$+"RRD"
3100 IF C=5 THEN LET H$=H$+"RLD"
3110 RETURN
3120 IF T=6 AND C<4 AND C<>1 THEN LET H$=H$+"IM "+STR$ ((C>0)*(C-1))
3130 IF X0=68 THEN LET H$=H$+"NEG"
3140 IF X0=69 THEN LET H$=H$+"RETN"
3150 IF X0=77 THEN LET H$=H$+"RETI"
3160 RETURN
3170 REM BIT OPS
3180 LET CB=1
3190 LET X0=PEEK (J-1)
3200 LET W=INT (X0/64)
3210 LET X0=X0-W*64
3220 IF W THEN GOTO 3350
3230 LET B1=1+INT (X0/32)
3240 LET B2=INT (X0/16)
3250 LET X0=X0-B2*16
3260 LET B3=INT (X0/8)+3
3270 LET B2=B2+5
3280 LET X0=1+X0-(B3-3)*8
3290 LET Q$=R$(X0)
3300 GOSUB 1170
3310 LET H$=H$+C$(B1)+C$(B3)+C$(B2)
3320 IF B2<>6 THEN LET H$=H$+" "
```

```
3330 LET H$=H$+Q$
3340 RETURN
3350 LET W=3*W-2
3360 LET D=INT (X0/8)
3370 LET X0=X0-D*8+1
3380 LET Q$=R$(X0)
3390 GOSUB 1170
3400 LET H$=H$+B$(W TO W+2)+" "+
STR$ (D)+","+Q$
3410 RETURN
3420 REM FD,DD PREFIXES
3430 LET DD=X0=221
3440 LET FD=X0=253
3450 IF L>1 THEN GOTO 3480
3460 LET H$=H$+"DATA"
3470 RETURN
3480 LET X0=PEEK (J-L+1)
3490 GOTO 1000
3500 REM MAIN LOOP
3510 INPUT J
3520 LET H$=""
3530 LET P$=H$
3540 LET Q$=H$
3550 LET DD=0
3560 LET FD=DD
3570 LET CB=DD
3580 GOSUB 350
3590 GOSUB 300
3600 GOSUB 420
3610 LET H$=H$( TO LEN H$-2)
3620 LET X0=PEEK J
3630 FOR K=1 TO L
3640 GOSUB 300
3650 LET J=J+1
3660 NEXT K
3670 GOSUB 970
3680 LPRINT H$
3700 GOTO 3520
3710 SAVE "DISA z"
3720 LIST
```

length and especially because of the limitations on dealing with the operating system (ROM) and the IX and IY registers in the T/S1000. It is

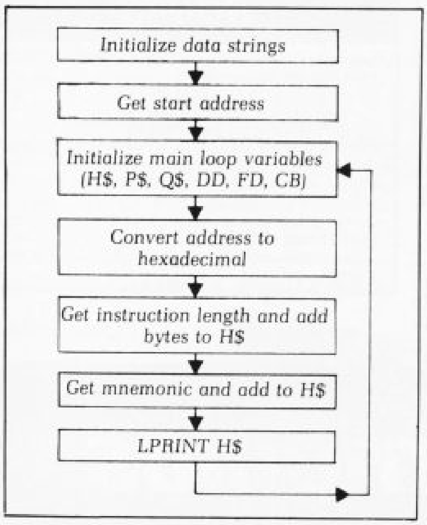


Figure 1. *Block diagram of Z80 disassembler.*

also unlikely that disassembly of the T/S2000 ROM would require anything so risqué. Nevertheless, it would not be difficult to add a few extra mnemonics for codes such as LD HY,A (load the contents of A in- to the high eight bits of IY) or LD B,LX (load the contents of the low

ing you to amputate your thumbs in order to count or calculate. The Z80 mnemonics are organized according to bit fields that correspond to the digits of the code in octal, but the digits are not always used in sequence.

Figure 2 shows how a binary byte

Disassembling Z80 machine code is much simpler than you might think after looking at a listing of the mnemonics

eight bits of IX into B), whose codes would be FD67 and DD45.

Disassembling Z80 machine code is remarkably more simple than you might think from looking at a listing of all those mnemonics, which often appear as jumbled as the contents of a programmer's desktop. The least helpful listing is the alphabetical one that often accompanies assemblers. The listing in your T/S1000 manual is one of the better ones in that it makes the command structure versus numeric code about half clear. All those eight-bit register LDs from code 64 (40 hex) and the ADDs through CPs that follow show a logical pattern that lends itself easily to programming a dissassembler. It's the first and last blocks of 64 instructions that look like trouble.

It was when I first saw an octal listing of Z80 instructions that the full structure of the mnemonics set

A disassembler shouldn't crash if it runs into a patch of meaningless data

became apparent. With hexadecimal and binary (to say nothing of decimal) already on your plate, you may welcome another number system like a second mother-in-law, but if you'll bear with it briefly, you'll see how DISA-Z sorts to crank out the mnemonics that go with machine code, without even requir-

is split up for expression in hexadecimal and octal. The first four bits evaluate as one hex digit and the last four as the other. It takes three binary bits to make one octal digit, so the highest octal digit is left one bit short and takes values from 0 to 3 rather than the full range from 0 to 7. Octal values from 000 to 377 scan the byte values from 0 to 255.

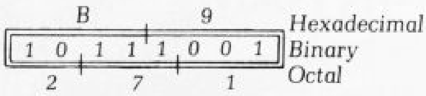


Figure 2. *The hit fields of a binary byte (185 decimal) expressed in hex and octal.*

The high digit, from zero to three, distinguishes the four groups of mnemonics with codes from 0 to 63, 64 to 127, 128 to 191, and 192 to 255. The second group is the eight-bit register loads, and the third includes the arithmetic and logical operations between eight-bit registers. These are easy to disassemble because each of the eight registers and eight operations corresponds to one of the eight octal digits.

To make sense of the first and last blocks of instructions, the numerical sequence doesn't help, but if you regroup the instructions according to the lowest digit, so that all instructions ending in zero are put together, all those ending in 1 together, and so on, then all of the 3X0 instructions are conditional forms of the REI instructions, all the

GLADSTONE ELECTRONICS

NEW Software

Tiny Logo

Let the world of computers become a comfortable one for your child through the use of this powerful and fun teaching language. In the age of the computer, **Tiny Logo** is the key to helping your child to learn and understand real computer programming. (Tiny Logo is a sub-set of the popular MIT Logo)
Z111 **$19.95**

Home Money Manager

This complete home budgeting system allows you to establish your own budget categories, keep track of monthly bills and income, and do a monthly and year end balance sheet. No bookkeeping knowledge is needed.
Z104 **$14.95**

Business & Management

Business Pac I: Cash Management

The most needed financial analysis routines for many business decisions, plus a detailed guide on how to use them. Five long programs are supplied on 2 cassettes. Invaluable aid for the business man. Break Even Analysis. Cash Flow Analysis. Payback Analysis, Internal Rate of Return. Depreciation Analysis. Z49 **$19.95**

Business Pac II: Project Scheduler & Proposal Sheet.

Time is the most valuable commodity for the businessman. **Project Scheduler** helps you keep track of work in progress and billings in progress. **Proposal Sheet** is a tool to help you quickly record work items and hours, billing rates, overheads, material and labor costs, to finalize your proposals.
Z50 **$14.95**

Personal Financial Planning Pac

Contains 2 tapes with a detailed reference flash instruction manual to help you gain and maintain control of your personal finances. The programs include: Checkbook balancing; Personal Net Worth; Budget Plan (for home or business); Mortgage Plan.
Z53 **$19.95**

Personal Stock Market Planning

Contains 4 programs recorded on 2 tapes, accompanied by a detailed manual. To help you plan, record and control your stock market activities. These programs have been adapted for the **ZX81** and **TS1000** from routines used in large scale computers by a leading software house. Programs include: Dow Jones Moving Average; Personal Stock/Bond Transaction Recorder; Portfolio Review; Point and Figure.
Z54 **$19.95**

FREE CATALOG *TS1000/1500/ZX81*
Send for yours today

This exciting new 34 page color catalog lists arcade, fantasy and family games, business and educational programs; books for beginners as well as experienced users; hardware add-ons and other peripherals for use with ZX81/TS1000 Home Computer!

Family Entertainment

KRAKIT™ Bored with crosswords? Want a real mental challenge? Try solving **KRAKIT**, the ultimate adventure and treasure hunt for the personal microcomputer. **KRAKIT** consists of 12 clues that will lead you around the world in search of 12 cities, 12 countries and 12 numbers. And if you are clever enough to be the first to solve these clues, then you can win at least **$20,000** prize money. Each cassette also includes a full sized **KRAKIT** poster at no charge!
Z48 **$9.95**

Computerized Diet Program

Gain control over your eating patterns. This is an individually tailered, systematic weight loss program. Uses a personal status assessment. Based on your assessment, the **TS1000/ZX81** becomes your monitor and feedback system, allowing you to discretely track yourself against your optimum model. Complete with a 22 page Guide Book, personal data storage tape, and vinyl storage case. Z47 **$19.95**

ZX Scramble You are flying into the heart of alien territory. As you dodge in and out of the rapidly changing terrain you fire laser rockets and bombs at the enemy weapons installations. They respond with missiles and flying bombs. Destroy them before they destroy you!
Z90 **$9.95**

Gobbleman Arcade action fun for the family. A superb realization of this exciting classic game, showing the true capability of the **TS1000** and **ZX81** computer. Written in machine code, this one is fast, fun and challenging.
Z70 **$9.95**

Alien Space Ship A long, involving word adventure game for the **TS1000** and **ZX81**. While on space reconnaissance, your ship has been dragged onto an enemy space cruiser. They are intent on taking humanoid life forms as slaves. You must reach their computer room and release your ship. Fast and varied responses arise and there are many different ways to win the game.
Z81 **$9.95**

Galaxy Invaders Another arcade classic, masterfully translated to the **TS1000** and **ZX81**. Protect your seven lives against fleets of hostile invaders. Fast, exciting and unpredictable.
Z91 **$9.95**

All software is compatible with the TS1000/TS1500/ZX81 Computers!

Chess Choose from 4 games, for beginner to Master!

Beginner (1K) Chess A single level of play introduces Chess. Z87 **$9.95**

2K Chess (for unexpanded **TS1000**) Better graphics and vastly improved range of moves.
Z55 **$12.95**

Chess I (Enhanced) 16K Full graphics display with six levels of play. Choose black or white. Castling and en passant. Save games in progress.
Z85 **$14.95**

Chess II Master The ultimate 16K chess for **TS1000**. 7 levels of play. Highest rating by Timex Sinclair User magazine.
Z86 **$19.95**

Trader A trilogy of 16K programmes that combine to give an epic 48K graphic adventure. As a galactic trader, you deal with some very bizarre customers indeed. Will you live to tell the tale?
P10 **$14.95**

Home Mathematics Tutor

Grades 1 to 6

$29.95

Each package contains 32 full 16K programs, plus a 32 page workbook! The most ambitious software project undertaken for the TS1000 demonstrating the ability of the computer to aid in mathematics instruction. These are self-teaching programs with sufficient fun and feedback to keep young minds involved. Give your child the TS1000 learning advantage. Complete with handsome vinyl storage case. M1 (Gr. 1) through M6 (Gr. 6).
Ea. $29.95

Shipping and handling
Software and Books: add $1.50 for first item, .50c for each additional item, to a maximum of $6.00. **Hardware:** add $3.00 for first item, $1.00 each additional item, to a maximum of $6.00. Maximum shipping charges on combined software, books and hardware orders is $9.00.

If an item you order is temporarily out of stock you will be advised and notified of your options.

Clip and mail

1585 Kenmore Ave., Buffalo, New York 14217

Toll Free (800) 833-8400 In N.Y. Call (716) 874-5510

ORDER FORM

CHARGE TO:
☐ Check
☐ Money Order
☐ VISA
☐ MASTERCARD

Expiry date ____
account number ____

Code#	Qty	Price	Total	Code#	Qty	Price	Total

Sub-Total ____
New York Res. add sales tax ____
SHIPPING ____
TOTAL ____

NAME ____
ADDRESS ____
CITY ____ STATE ____ ZIP ____

GLADSTONE ELECTRONICS, INC. 1585 Kenmore Ave., Buffalo N.Y. 14217

In Canada: Gladstone Electronics, 1736 Avenue Rd., Toronto, Ont. M5M 3Y7 (416) 787-1448 800-268-3640
TSU6

For more information, circle 16 on Reader Service Card.

3X2s are conditional jumps, and all the 3X4s are conditional calls. The eight conditions themselves (in E$) are a function of the second octal digit, the X.

Other groups of instruction types vary with whether the X is even or odd, often with a regular group alternating with an irregular one, but just about all the order in the system can be extracted by looking at the codes in octal form. DISA-Z first sorts on the high octal digit of the code it is reading and then derives the second and the low digit as the variables C and T (lines 230-290), on which the details of the mnemonic sort are based.

When a group is irregular, like the 0X7 group, then the value of X is simply used to slice the mnemonic out of a data string like F$ = "RLCARRCARLA RRA DAA CPL SCF CCF" (line 160), taking four characters at a time.

The CB (203 decimal) prefix redefines almost all of the code values in a way that is easy to sort for the mnemonics BII, RES and SET. (Consult again the appendix on the character set in your T/S instruction book.) The block of the first 64 codes contains eight more that are not normally defined in the Zilog canon; these are CB 30 through CB 37. In fact all of these instructions "work" in the sense that they rotate the proper register values left and move a 1 into the vacated right bit, effectively replacing the value X in the register with the value 2X + 1. Since I decided to build this group of mnemonics from the string RSLRC-AL (one of the first pair of letters, one of the second pair, and one of the last four), it was easier to christen this group of mnemonics SLL (shift left logical) and to go ahead and disassemble it rather than to pretend that the instruction didn't exist and wouldn't affect the code. You'll be able to fill in these extra mnemonics in the gap in your T/S instruction book, starting with SLL B.

The FD and DD prefixes affect only those instructions that involve (HL) or HL, ignoring the undocumented instructions mentioned earlier (and which affect any mnemonics involving H or L). DISA-Z sets a variable flag called DD or FD for these prefixes and then postpones any action until the (HL) pointer or HL pair are processed. If the undocumented instructions were disassembled too, the H and L registers would have to be processed separately to check for the DD and FD flags.

The register-pair names are contained in the D$ (line 110) and the eight-bit-register names are contained in the R$ (line 140). Notice that the register names begin with the B and C registers, and that the A and AF-pair are the last in their sets. This order corresponds to the way the registers are referred to by the second or third octal digit of the instruction byte. Zero refers to the B register or the BC pair, and a 7 to the A register or the AF (or SP) pair.

With hex, binary and decimal already on your plate, you may not wish to learn another number system

Using DISA-Z

The variable TS2 in line 190 must be set to one for the T/S2068 and to zero for the 1000 or 1500. TS2 affects the single use of the CHR$ function in line 330 to adjust for ASCII or "Clive" code.

If you want DISA-Z to print to your screen, then change line 3680 to PRINT H$ and add a line 3690 to control the scroll. On the T/S2068, the line should be POKE 23692,255 (but check your instruction book to make sure that this is still the SCROLL COUNT variable). On the T/S1000, the line should be IF PEEK 16442 = 2 THEN SCROLL, and a line 210 SLOW is necessary to see the screen.

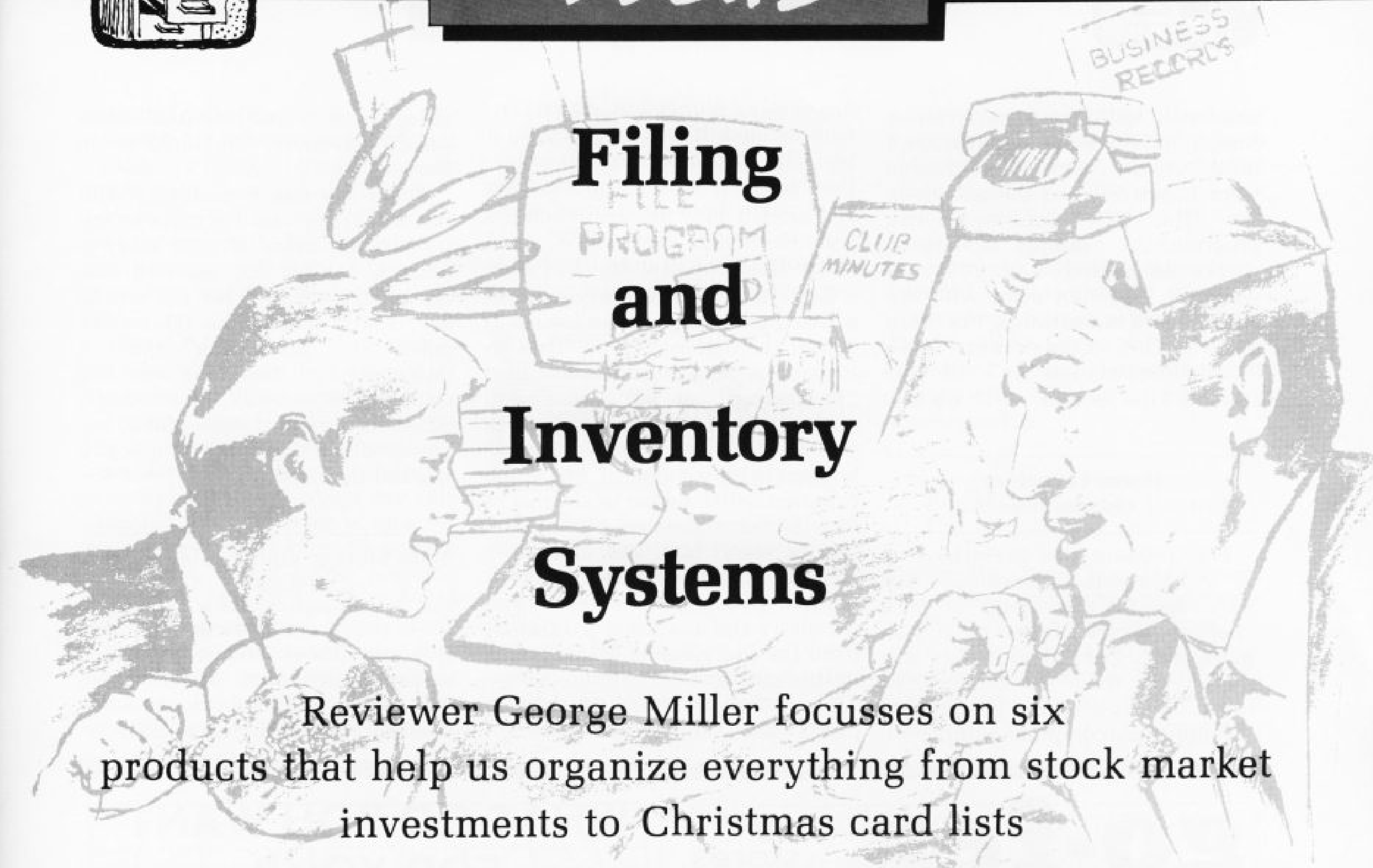
Your BASIC ROM on the 2000 will begin at address 0 just as it does on the T/S1000, but it will run for 16K (to 3FFF hex) rather than 8K. Both ROMs include a number of data files, which are not Z80 code, but which DISA-Z will merrily disassemble as if they were. Until the map of the ROM is known, it is you the reader who will have to recognize these patches of Z80 nonsense and to work them out as address files, ASCII files or system data. On the T/S1000, the instructions RST 08 and RST 28 are also followed by one (for 08) or more (to the first 34 hex) bytes of data; DISA-Z also misses these and can be occasionally tricked by such bytes into misreading an instruction or two before it locks back onto the correct sequence. I would have corrected for this feature if I could have been sure that the 2000 would make the same use of the RSTs as the 1000. When you find out you can modify the code at line 2000 to allow for the data bytes.

Not having yet seen a live T/S2068, I am not certain how to get at the remaining 8K of ROM, known as SYSTEM ROM, for disassembly. SYSTEM ROM will be bank-switched into the 0-8K memory block, but it is unlikely that you will be able to run a BASIC program like DISA-Z after switching it in. Those of you who can handle machine code might try a short LDIR routine to copy SYSTEM ROM to a higher block of RAM that can be active along with the full BASIC interpreter. You can then change lines 360 and 2200 to subtract the offset from the address so that your disassembly is printed as if it began at address 0.

You might also want to search your instruction book for the RAM address of the friendly function dispatcher and to disassemble that. We all look forward to reading about your discoveries on these pages. — ***Ray Kingsley***

FILE PROGRAMS

Focus



Filing and Inventory Systems

Reviewer George Miller focusses on six products that help us organize everything from stock market investments to Christmas card lists

IN RECENT MONTHS, quite a few products of varying use have appeared on the market designed to help control inventory or set up filing sytems for home and business.

Although many people might sneer at the idea of a company using a T/S1000,ZX81 as the corporate computer;, many of us have need of some sort of filing or inventory system at home. Consider the uses: Christmas card list, birthdays, recipes, record and tape log, insurance purposes, stock market or investments, telephone numbers, and so on.

These need constant updating and editing and that is what the Focus is about this month.

The Organizer

Timex, 16K

Clearly this product has several advantages over others designed to do the same job. Once it is loaded, you have several tasks to complete. In order to compare tapes I decided to set each with the task of handling a small mailing list.

With this Timex program, the first job I faced was to set up a format for the file itself. In other words, imagine a blank filing card in front of you. Now decide what goes on it and where. These titles are a permanent feature to be included in all files. I used name, street, city, state, zip, phone number, date of birth and notes for headings. Then you decide where each entry will begin.

One program's documentation is a 60-page book

From here on the T/S,ZX guides you very smoothly through the process of setting up files. There are 13 commands available to work through the files. ENTER allows additions to the system. ALTER changes the file displayed on the screen. INFORM tells you how many files are in memory, how many pieces of data per file and what percentage of the file is filled. FORWARD and BACK simply move you through the file. RESET prepares the files for use. LIST takes you through them. ORDER allows you to decide you you go through; either by the way they were entered, by name, date of birth, state or whatever else strikes your fancy. SELECT lets you search for all files with some feature in common. This feature can be retained for further use or disposed of immediately. PRINT and COPY are self-explanatory, as are DELETE and QUIT.

Each of the commands is accomplished by typing in the first letter of the word: for example "A" will let you ALTER the file on display.

The load time was quick — two and a half minutes — but there were drawbacks. On loading I was presented with the record/format screen, not the main menu. When I hit PRINT, the memory lost the file format, but retained the entries in the file. The accompanying program, an example of how it works is called GAZETEER. It wouldn't load, not even with the winky board in operation. For more about the Organizer, see "Computerized Dee-Jay" in our Hints & Tips section. ★★★½

Home Inventory
Orbyte, 16K

This product also performed up to expectations. When loaded, it took me to the menu (a nice change!), where I found I could inquire, add, change or delete any item, total them by number, or finish. In this program I decided to inventory my computer equipment. Fearlessly I plunged in. ADD ITEM, I commanded. It was extremely simple. The screen was set up to take the following bits of information: last update, item description, quantity in stock, unit price cost, category code and an add or change.

Each item you inventory requires a five-digit number. The lowest is 00001; the highest, 99999. This brings up a 30-second delay called "Processing" at all times and nothing will work without a number. If in your home you had 87 widgets with a value of $1.19, the program willshow the total value of your widget collection.

It is very fast and could be adapted to a business inventory without difficulty. However, it is an inventory and as such it is different from the file systems under Focus in this article.

The TOTAL BY ITEM NO. feature prints (on screen) the number, category code, and cost of all items on the inventory in blocks of 10 items.

There are minor problems with this one. After each prompt or entry you are asked if your entry is correct. I hit Y for yes and was repeatedly told I'd hit the wrong key. I also found the 30 second delay for "Processing" quite a nuisance. The load time was five minutes, 39 seconds. This increased as information was added, but the save/load was quite simple and worked the first time. ★★★½

Data Storage and Display System
ZX-Panding, 16K

This program loads in just under five minutes: so far, so good. The screen then shows you themenu and invites you to choose between opening a new file; listing file titles;

entering data into an existing file; correcting a point in a file; tabular listing of a file; graphing a file; saving to tape; close out a file; end.

Well, the instructions mentioned that the program "has the capability of storing up to sixty data points in a file under any user-defined titel."

"Great," I thought. "I'm back to my telephone book plan and I'll try to graph the phone numbers."

Wrong again, Charlie! Pressing a key for option number 1 (opening a new file) is the usual place to being. It would hae been except my silly machine kept telling me of an error code message (you know those little messages taht add a new meaning to the word cursor.) This one was 2/320. Aha — a quick check to my ever-handy owner's manual revealed that this was an undefined variable. Fortunately, the listing was not beyond my grasp of BASIC, so I figured a way around it and started again.

I actually got to see the next screen before — you guessed it, error code message 2. To make a long story short, my machine really didn't like the way Al Bandy of ZX-Panding Ltd. had written the program. Now in all fairness I'll confess right here that I was using a ZX81 with 16K RAM and not a T/S1000 or T/S1500. Maybe they differ.

Some programs are ideal for small businesses

So I checked that out. Wrong again, Charlie! I really began to feel like a tuna. However, before I went on to the next one, I listed a few other annoying things. The program came in a sandwich bag with the instructions around the cassette under an elastic. The cassette had no case, but a paper strap between the guard tape sprockets. I'm not sure how to store it. I'm not even sure if I want to store it.

However, winter is coming and maybe it'll be fun to se if I can get this thing working over the next few months. Stay tuned for future developments. ★½

Business Inventory Control
Mindware, 16K

Inside the gray box (cardboard) comes the cassette in a plastic case — the next program to focus on, with a feature called Quickload. Because of quickload, I barely had time to light a cigarette before — it crashed. Try again. This time I actually got the cigarette lit before — it crashed. Well I tried LOAD, LOAD "STOCK I", LOAD "STOCK II on both sides. Then I tried again with my Winky Board. Success! But this success lasted only seconds — just long enough to read the title screen.

Well, I was getting tired, so I trudged off to seek sleep and solutions to those nagging problems. The next days were no different. Anyway, I have no choice but to tell you what the manual says these programs can do. I'm sure you'll keep in mind that they didn't do it for me, but who knows what your luck is like.

With STOCK Iyou have three options: find, add or delete a line; print stock list; save data on tape. EAch entry requires a name, which is alphanumeric in nature. Numbers are sought before alphanumeric entries. The title of the file is then complete and you fill in unit price, supplies code, type code, stock level and re-order level for each.

STOCK II is a variation of the first program (so I read) which "accommodates over 2000 inventory items in 16K, if only numeric line codes and inventory levels are entered.★½

Inver Inventory Accounting
D. Lipinski, 16K

Would you believe it? After four tries it finally loaded. The opening screen asks for the length in digits of the number you'll assign to each item to be inventoried. You follow with the lenth of the word description. Basically from here on this product is on inventory control rather than a pure filing system. Unlike the Home Inventory, this is business oriented and its object is to insure adequate stock of items is maintained. It does this extremely well. For people operating home businesses (like Amway distributors), this is an excellent tool.

I had some difficulty altering each item. In fact, the only solution seemed to be to re-enter the entire file for that item. The instructions are clear and concise. The program allows you to save, make changes, add a new item, delete an item, search items, print complete inventory, check on reorder items, and clear for new accoutning period. As mentioned, I had problems with changes and searches. ★½

ZX Pro/File
Thomas B. Woods, 16K

Well, I finally got around to the last one for this Focus. I left it to the end because the documentation with it is literally a book! Its 59 pages include BASIC and machine code listings with explanatory notes; suggestions to adapt to a faster loading progam (Thomas Woods recommends the International Publishing & Software Fastload), the Timex or Sinclair printers and larger mnemonics. Also, Woods will supply a quarterly newsletter at $9.95 yearly to update and develop the program.

Clearly this is the best of the programs I focussed on. Consider the obvious advantages — it loaded and ran! Then after entering my telephone-address files it was able to search all files for any *one or two word* combination of numbers and letters and list them. It can alphabetize your files no matter how you enter them. This is not an automatic feature, but works efficiently and quickly.

The progam comes with adaptations through a Memotech interface to a full size paper printer. The program listing changes it to adapt to my Sinclair printer were concise and worked.

Combined with Fastload (from International Publishing & Software) it does actually load in 73 seconds. The change in the listings were provided, and I checked and they do work.

All in all, this is the one I'd recommend. As Woods himself says: "This program is dedicated to all those Timex and ZX81 owners who refuse to believe the Apple, TRS-80, OSI, IBM Commodore and Atari users and dealers who constantly remind us that the Timex 1000 "is just a try!" He proves his case dramatically. ★★★★

— George Miller

PROGRAM PRINTOUT

Contest Winners

HERE they are, the ten winners to the Clever Contest! The rules asked for a program that did something different, unusual or clever and the programs you sent in did that! The response was great, thank you.

Each winner's program is given here with the explanation sent in with it. We hope you enjoy them as much as our editors did (at least I think they enjoyed them, judging from all the laughter I heard coming from their offices).

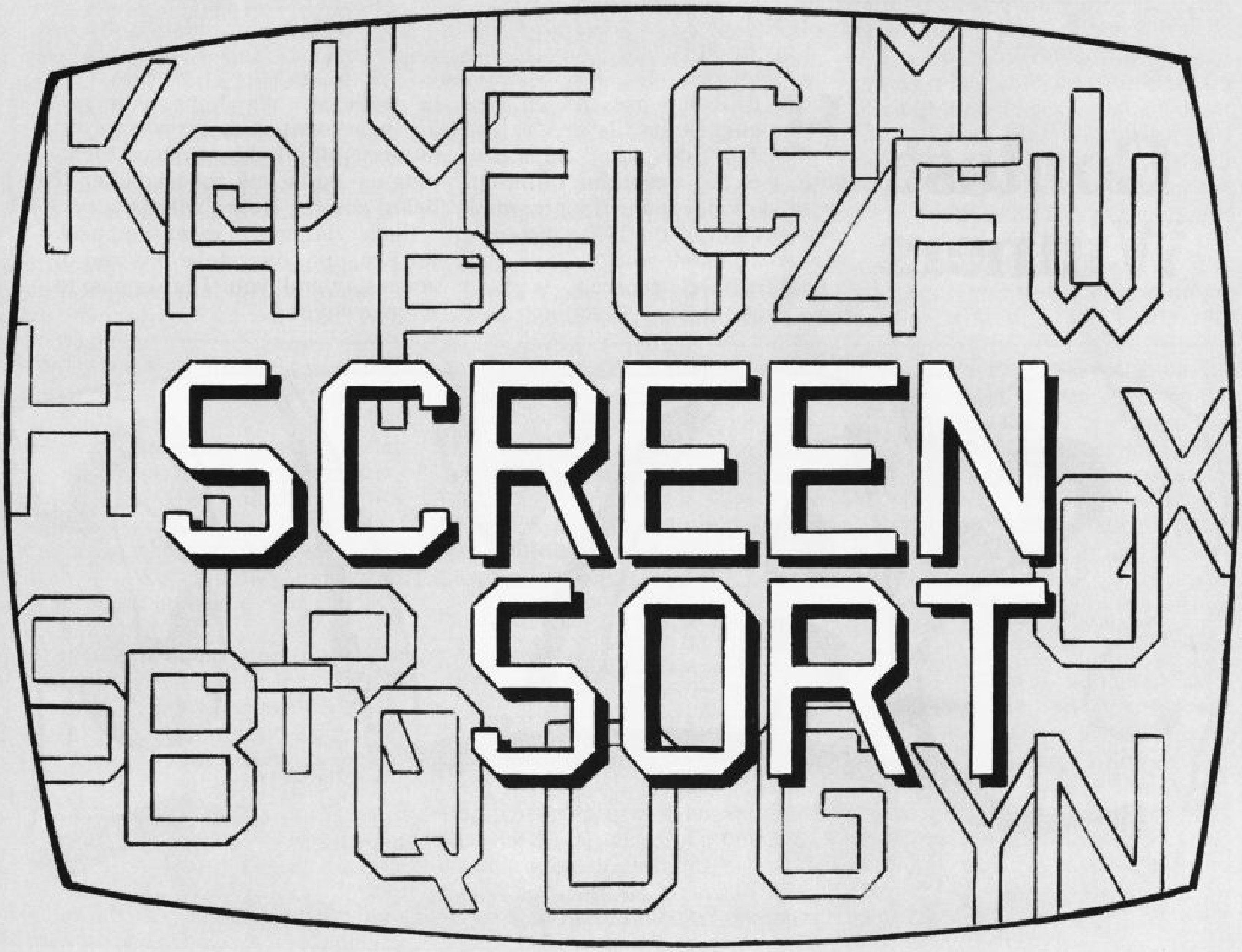
Each winner will receive a check for $20, our congratulations and of course a thank you. Thanks also to all who entered.

SAVE

THE only purpose of this program is to save itself. Submitted by A Peter Hollis of Wilson, North Carolina. (T/S1000,1500,ZX81 2K).

```
  5 REM SAVE "SAVE"
 10 REM 2K
 15 PRINT AT 8,13;"HUMAN";TAB 1
3;"▀▀▀▀▀";AT 11,2;"SET UP YOUR R
ECORDER, HUMAN.";TAB 4;"THEN BRI
NG TAPE, HUMAN.";TAB 6;"BRING LO
TS OF TAPE "
 20 PAUSE 1000
 25 CLS
 30 SAVE "SAVE"
 35 PRINT AT 9,7;"MORE TAPE, HU
MAN,";TAB 7;"BRING MORE TAPE--";
TAB 8;"WE ARE GOING TO";TAB 7;"N
EED ALL THE TAPE";TAB 9;"IN THE
WORLD."
 40 PAUSE 1000
 45 CLS
 50 SAVE "SAVE"
 55 PRINT AT 10,10;"MORE, HUMAN
";TAB 10;"STILL MORE."
 60 GOTO 40
```

THIS program is somewhat unusual but also useful. It fills the screen with alphabet characters in random order, then does a sink sort on them to get them in order. It is interesting to watch and gives a very graphic demonstration of how a sink (or bubble) sort works. Of course, it should be run in the slow mode to watch it. It is very slow, however, and takes about 11 hours to run completely. Submitted by David Lee of Longwood, Florida. (T/S1000,1500,ZX81 16K)

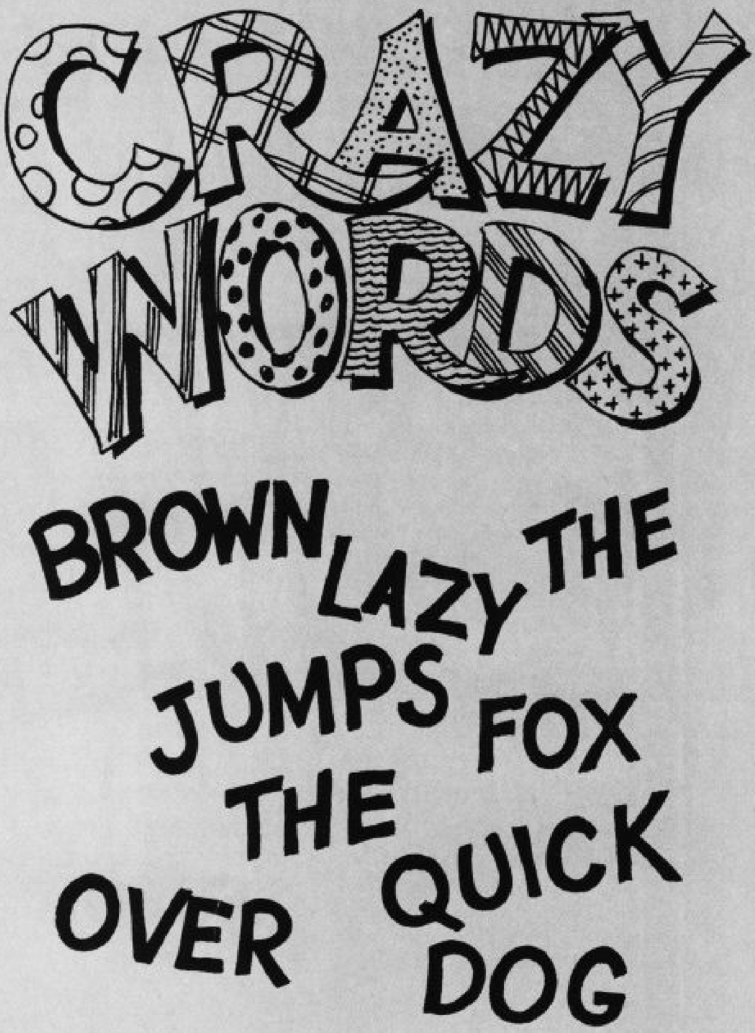



```
   1 REM ************************
   2 REM **                    **
   3 REM **    SCREEN SORT     **
   4 REM ** BY DAVID LEE       **
   5 REM **                    **
   7 REM ************************
  10 FOR A=1 TO 704
  20 PRINT CHR$ INT (RND*26+38);
  30 NEXT A
  40 LET S=1+(PEEK 16396+256*PEE
K 16397)
  50 LET B=S+723
  60 LET C=0
  70 FOR A=S TO B
  80 LET X=PEEK A
  90 LET Y=PEEK (A+1)
 100 IF X<=Y THEN GOTO 180
 110 POKE A,Y
 120 POKE A+1,X
 130 LET C=1
 140 NEXT A
 150 IF C=0 THEN STOP
 160 LET B=B-1
 170 GOTO 60
 180 IF Y<>118 THEN GOTO 140
 190 LET Y=PEEK (A+2)
 200 IF X>Y THEN GOTO 230
 210 LET A=A+1
 220 GOTO 140
 230 POKE A,Y
 240 POKE A+2,X
 250 LET A=A+1
 260 GOTO 130
```

TOTALLY AWESOME turns the name of this totally awesome magazine into the most unusual, most different, cleverest, and strangest graphic you have ever seen.

To see this awesome graphic, just enter the program using your 16K RAM pack. (The program will run in 2K but the results are better in 16K.) Type RUN and ENTER, and watch. If you see something real cool just press the P key. Then it will wait for you to press another key. If you want to stop the program, hold down the S key and press BREAK. One more thing: if you turn up the volume of your television, you will hear music. Submitted by Mark Neirick of Des Plaines, Illinois. (T/S1000,1500,ZX81 2K).

```
 1Ø REM Y2 GOSUB ?TAN
 2Ø SLOW
 3Ø POKE 16517,71
 5Ø PRINT AT 9,2;"...##...###.#.#...#...#..."
 6Ø PRINT AT 10,3;".###.#####.#.#.###.###.#."
 7Ø PRINT AT 11,3;".#.#...###.#.#...#..##..."
 8Ø PRINT AT 12,3;".#####.###.#.###.#.###.."
 9Ø PRINT AT 13,3;".###...###...#...#...#.#."
1ØØ LET L=USR 16514
11Ø LET K=RND*29
12Ø IF INKEY$="P" THEN PAUSE 4E4
13Ø IF INKEY$="S" THEN LET K=3Ø
14Ø POKE 16515,K
15Ø GOTO 1ØØ
Note: . = inverse space
      #= space
```



THIS program is very similar to Madlibs in that you supply nouns, adjectives and verbs to go with the rest of the sentence supplied by the computer.

Each of the five verbs, adjectives and nouns as well as the sentences to go with them are selected completely at random, that is, you will not get the same combination twice while running through the program fives times. After the fifth time through, arrays C, D, G and X are reset to 0. This allows the program to run through five more (usually) different combinations.

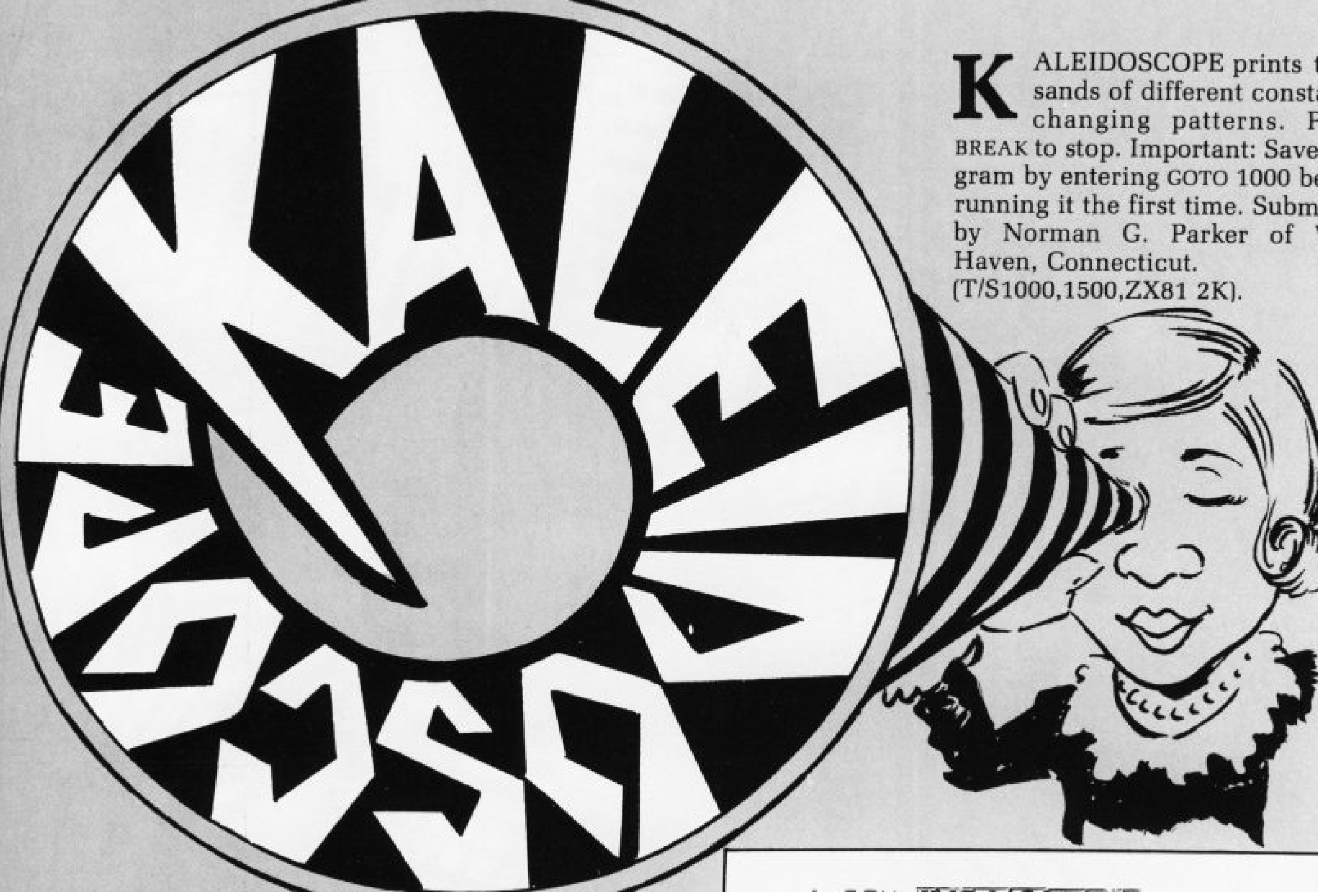
If you wish to save the words you have entered, change line 730 to GOTO 300 and enter GOTO 730 in the direct mode.

If you break the program never use RUN as you will destroy the words you have in the arrays. Instead use GOTO 300 in the direct mode.

Once this program has been saved on tape, it will come up running when loaded into the computer. Submitted by Wendell R. Fischer of Dalton, Ohio.
(T/S1000,1500,ZX81 16K)

```
  10 RAND
  20 DIM X(5)
  30 DIM E$(5,12)
  40 DIM G(5)
  50 DIM B$(5,12)
  60 DIM G$(5,18)
  70 DIM C$(5,12)
  80 DIM C(5)
  90 DIM D(5)
 100 PRINT "ENTER 5 ADJECTIVES,O
NE AT A TIME12 LETTERS OR LESS"
 110 FOR A=1 TO 5
 120 INPUT B$(A)
 130 NEXT A
 140 CLS
 150 LET G$(1)=" WANTS TO "
 160 LET G$(2)=" SHOULD NOT "
 170 LET G$(3)=" SHOULD ALWAYS "
 180 LET G$(4)=" IS GOING TO "
 190 LET G$(5)=" WILL IF POSSIBL
E "
 200 PRINT "ENTER 5 NOUNS,ONE AT
 A TIME     12 LETTERS OR LESS"
 210 FOR B=1 TO 5
 220 INPUT C$(B)
 230 NEXT B
 240 CLS
 250 PRINT "ENTER 5 VERBS,ONE AT
 A TIME     12 LETTERS OR LESS"
 260 FOR B=1 TO 5
 270 INPUT E$(B)
 280 NEXT B
 290 CLS
 300 FOR E=1 TO 5
 305 LET G(E)=0
 310 LET X(E)=0
 320 LET C(E)=0
 330 LET D(E)=0
 340 NEXT E
 345 FOR N=1 TO 5
 350 LET K=INT (RND*5)+1
 360 FOR J=1 TO 5
 370 IF X(J)=K THEN GOTO 350
 380 NEXT J
 390 LET X(N)=K
 400 LET Y=INT (RND*5)+1
 410 FOR M=1 TO 5
 420 IF C(M)=Y THEN GOTO 400
 430 NEXT M
 440 LET C(N)=Y
 450 LET W=INT (RND*5)+1
 460 FOR L=1 TO 5
 470 IF D(L)=W THEN GOTO 450
 480 NEXT L
 490 LET D(N)=W
 500 LET H=INT (RND*5)+1
 510 FOR I=1 TO 5
 520 IF G(I)=H THEN GOTO 500
 530 NEXT I
 540 LET G(N)=H
```

KALEIDOSCOPE prints thousands of different constantly changing patterns. Press BREAK to stop. Important: Save program by entering GOTO 1000 before running it the first time. Submitted by Norman G. Parker of West Haven, Connecticut. (T/S1000,1500,ZX81 2K).




```
 550 LET D$=" A "
 560 IF B$(Y,1)="A" OR B$(Y,1)="
E" OR B$(Y,1)="H" OR B$(Y,1)="I"
 OR B$(Y,1)="O" OR B$(Y,1)="U" T
HEN LET D$=" AN "
 570 PRINT AT 9,10;"[graphics]"
 580 PRINT AT 10,10;"[graphics]"
 590 PRINT AT 11,10;"[graphics]"
 600 PRINT AT 13,9;"ENTER YOUR N
AME"
 610 SLOW
 620 INPUT A$
 630 LET A$="           "+A$+G$(H
)+E$(K)+D$+B$(Y)+"  "+C$(W)+"
           "
 640 LET B=LEN (A$)-10
 650 FOR F=1 TO 2
 660 FOR X=1 TO B
 670 PRINT AT 10,11;A$(X TO X+9)
 680 NEXT X
 690 NEXT F
 695 FAST
 700 NEXT N
 710 GOTO 300
 720 SAVE "CRAZY WORD[graphics]"
 730 GOTO 10
```

```
   1 REM KALEIDOSCOPE
          123456789012345678901234
56789
  10 FAST
  20 FOR F=16545 TO 16573
  30 SCROLL
  40 INPUT A
  50 PRINT A
  60 POKE F,A
  70 NEXT F
```

RUN above program and enter the following codes:

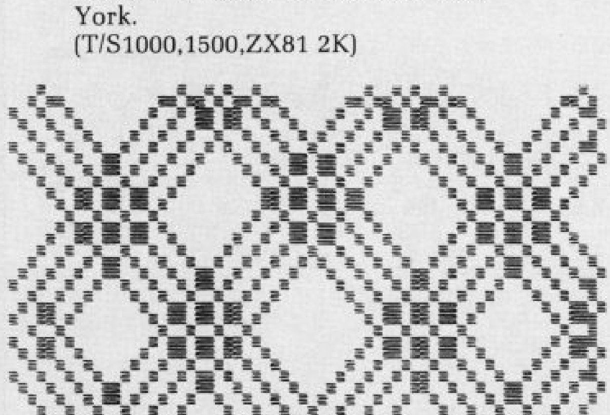
14	64	190	244
24	6	40	24
42	13	7	237
12	35	26	35
64	62	119	13
17	59	19	32
130	135	16	245
			201

```
  20 POKE 16418,0
  30 FOR F=0 TO 23
  40 PRINT AT F,31;" "
  50 NEXT F
  60 LET A=13
  70 LET X=USR 16545
  80 SLOW
  90 FOR D=1 TO 200
 100 NEXT D
 110 FOR F=A TO 31
 120 IF 32/F=INT (32/F) THEN NEX
T F
 130 POKE 16554,F
 140 FOR E=1 TO 100
 150 POKE 16514+INT (RND*F),INT
(RND*11)+(128 AND RND>.5)
 160 LET X=USR 16545
 170 NEXT E
 180 NEXT F
 190 LET A=3
 200 GOTO 110
1000 SAVE "K[graphics]"
1010 RUN
```

Important: Save program by entering GOTO 1000 before running it the first time.

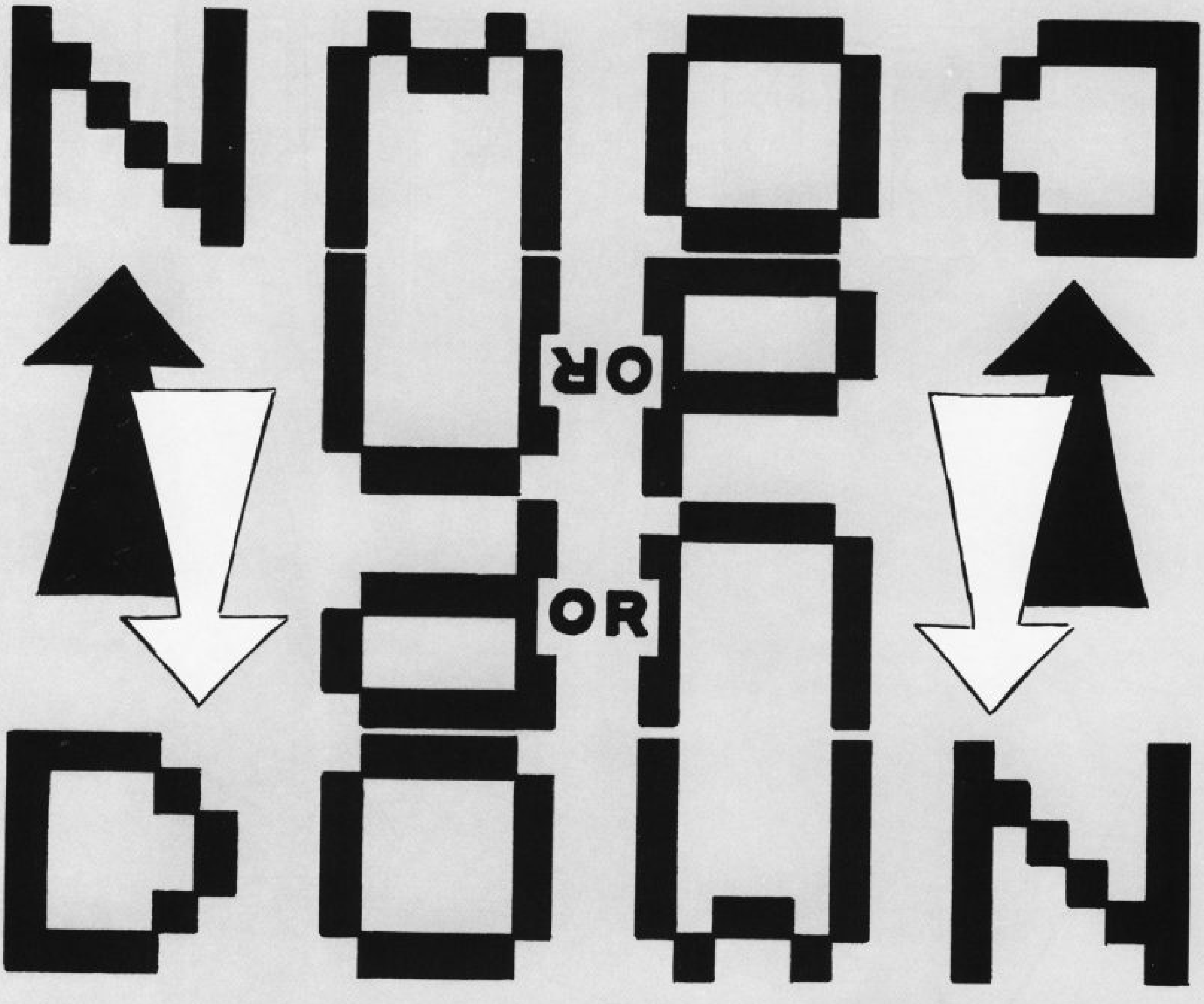

```
 1 LET B=0
 2 LET A=0
 5 LET C=0
 6 LET E=280
10 LET A=A+1
15 IF A<=63 THEN GOTO 20
17 LET A=A-63
20 LET B=B+1
25 IF B<=43 THEN GOTO 30
28 LET B=B-43
30 PLOT A,B
40 PLOT (62-A),(42-B)
42 PLOT A,(42-B)
44 PLOT (62-A),B
47 LET C=C+1
49 IF C<=E THEN GOTO 10
50 LET A=A-1
52 IF A>=0 THEN GOTO 54
53 LET A=A+63
54 LET B=B-1
55 IF B>=0 THEN GOTO 57
56 LET B=B+43
68 UNPLOT A,B
69 UNPLOT (62-A),(42-B)
70 UNPLOT A,(42-B)
80 UNPLOT (62-A),B
90 GOTO 50
```

PATTERN is a program that draws lines bouncing off the sides of the television screen. It draws for a limited time, then starts erasing them. It has a hard time erasing some marks on the sides of the screen. Submitted by Donald Hanson of Adams, New York.
(T/S1000,1500,ZX81 2K)




```
   1 REM AAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAA
  20 FOR X=16514 TO 16548
  30 INPUT A
  40 PRINT A;" ";
  50 POKE X,A
  60 IF A=201 THEN STOP
  70 NEXT X



DATA - 42 12 64 17 124 1 25 126
254 0 32 2 62 28 60 254 38 32 5
54 28 43 24 239 119 205 187 2
62 255 189 202 130 64 201



NOW TYPE:   LET X=USR 16514
AND THE PROGRAM WILL RUN.
```

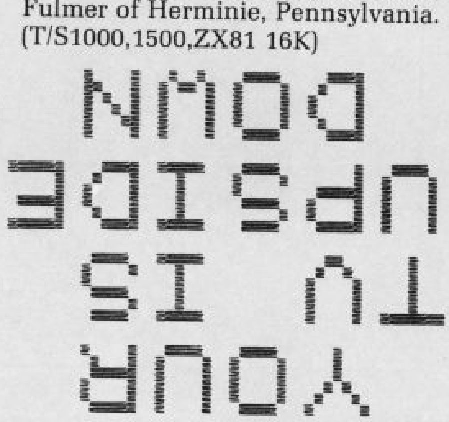
COUNTING is a program that demonstrates how fast machine language can count. This program counts by 1s until a key on the keyboard is pressed, then the program stops counting. Submitted by Lyn Neir of Overland Park, Kansas. (T/S1000,1500,ZX81 1K).




```
   5 DIM P$(4,8)
  10 LET A$="  YOUR   TV IS   UP
SIDE   DOWN  "
  20 FOR I=0 TO 3
  30 LET P$(I+1)=A$(I*8+1 TO (I+
1)*8)
  40 NEXT I
  50 FOR I=1 TO 4
  60 FOR J=1 TO 8
  70 FOR K=0 TO 7
  75 IF CODE P$(I,J)=0 THEN GOTO
 130
  80 LET C=PEEK (7680+CODE P$(I,
J)*8+K)
  85 IF C=0 THEN GOTO 125
  90 FOR L=8 TO 1 STEP -1
 100 IF C-2*INT (C/2)=1 THEN PLO
T 61-(J-1)*8-L+1,41-(5-I)*8+K
 110 LET C=INT (C/2)
 120 NEXT L
 125 NEXT K
 130 NEXT J
 140 NEXT I
 150 PRINT AT 0,26;"08/0"
 160 POKE 16384,74
```

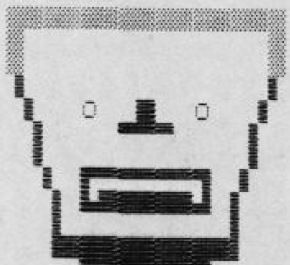
THIS program displays the message in A$ upside-down on the screen in enlarged characters. A$ *must* be at least 32 characters long. After the message is printed upside-down, the error message 0/80 appears upside-down on the screen. To make the program print the message right-side-up, change line 100 to:

100 IFC-2*INT (C/2)=1 THEN PLOT (J-1)*8+L-1, (5-I)*8-K and delete lines 150 and 160. Submitted by David Fulmer of Herminie, Pennsylvania. (T/S1000,1500,ZX81 16K)



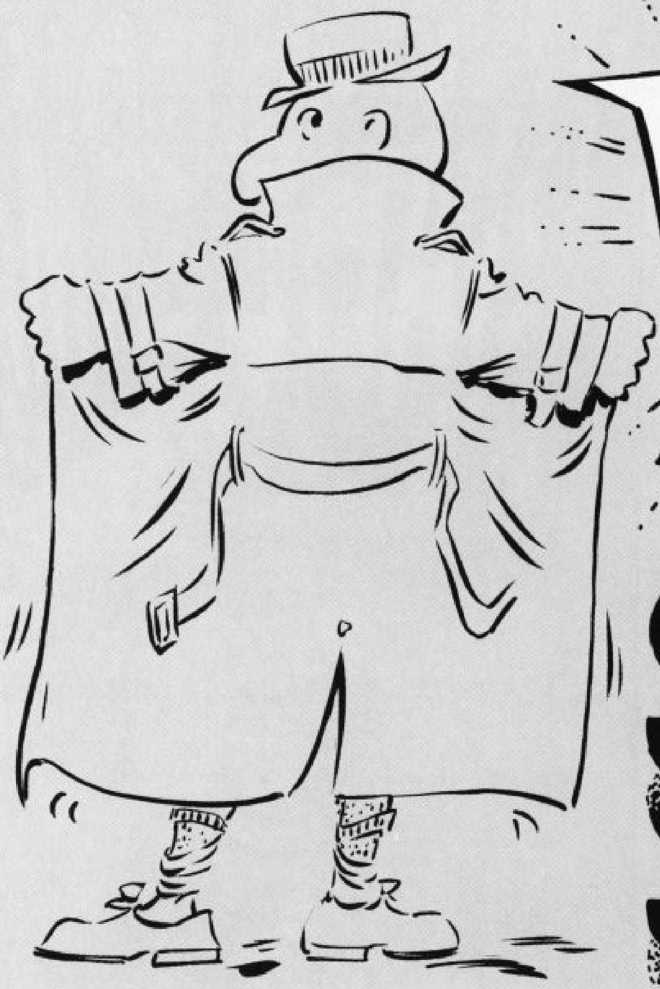
BLABBERMAN

```
10 CLS
20 SLOW
30 PRINT AT 5,10;"▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒"
40 PRINT AT 6,10;"▒";AT 6,24;"▒"
50 PRINT AT 7,10;"▒";AT 7,24;"▒"
60 PRINT AT 8,10;"▌";AT 8,24;"▌"
70 PRINT AT 9,11;"▌";AT 9,14;"O";AT 9,17;"█";AT 9,20;"O";AT 9,23;"▌"
80 PRINT AT 10,11;"▌";AT 10,16;"███";AT 10,23;"▌"
90 PRINT AT 11,11;"▌";AT 11,23;"▌"
100 PRINT AT 12,12;"▌";AT 12,14;"███████";AT 12,22;"▌"
110 PRINT AT 13,12;"▌";AT 13,22;"▌"
120 PRINT AT 14,12;"▌";AT 14,22;"▌"
130 PRINT AT 15,12;"███████████"
140 PRINT AT 16,14;"███████"
150 LET X=0
160 PRINT AT 9,14;"-"
170 PAUSE 10
180 PRINT AT 9,14;"O"
190 PAUSE 10
200 IF X=4 THEN GOTO 220
205 LET X=X+1
210 GOTO 160
220 PRINT AT 12,14;"▛▀▀▀▀▀▜"
230 PRINT AT 13,14;"▙▄▄▄▄▄▟"
240 PRINT AT 20,10;"I "
250 PAUSE 20
260 PRINT AT 12,14;"███████"
270 PRINT AT 13,14;"       "
275 PAUSE 20
280 PRINT AT 12,14;"▛▀▀▀▀▀▜";AT 13,14;"▙▄▄▄▄▄▟"
290 PRINT AT 20,12;"LOVE "
300 PAUSE 20
310 PRINT AT 12,14;"███████"
320 PAUSE 20
330 PRINT AT 12,14;"▛▀▀▀▀▀▜";AT 13,14;"▙▄▄▄▄▄▟"
340 PRINT AT 20,17;"MY T/S 1000"
345 PAUSE 20
350 PRINT AT 13,14;"▙█████▟"
355 PRINT AT 14,15;"█████"
360 PRINT AT 21,10;"NYAAAAAAAAH"
370 STOP
```



I LOVE MY T/S 1000

THIS program shows a man who winks his eye and opens and closes his mouth saying "I LOVE MY T/S1000"; after he gets done talking he sticks out his tongue and says, "NYAAAAAAHH" and then the program ends. The man loves his T/S1000 because he has found many new things to do with it in *Timex Sinclair User*. Note: in lines 70 and 180, the Os are the letter O for eyes. Submitted by Kevin Gregg of Columbus, Indiana.



FLASH AND SCROLL

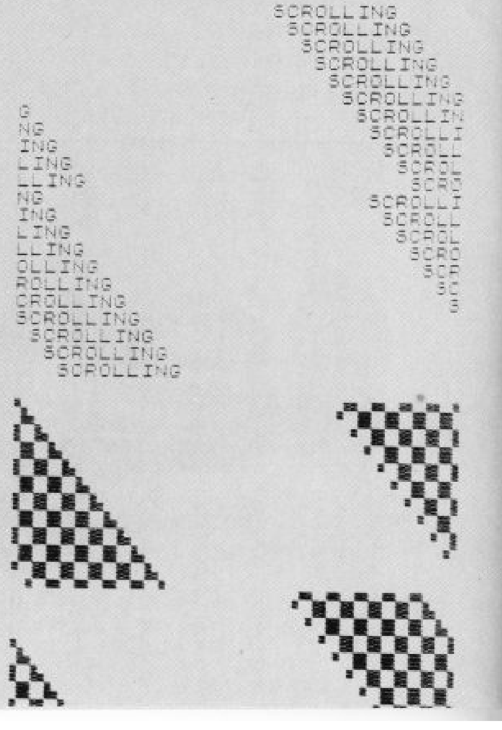
```
   1 REM 12345678901234567890123
45678901234567890123456789012345
6789012345
  10 LET A$="AF2A0C4023545D23F51
AF5011F00EDB0F12B77F13C232323131
3FE0B20EA00"
  20 LET A$=A$+"AFF52A0C4001D502
09545D7EF52B011F00EDB8F112F13CFE
0BC8F51B1B2B18EB"
  50 LET J=16514
  60 FOR I=1 TO LEN A$ STEP 2
  70 POKE J,16*CODE A$(I)+CODE A
$(I+1)-476
  80 LET J=J+1
  90 NEXT I
 100 STOP
 200 FOR I=0 TO 21
 210 PRINT AT I,I;"SCROLLING"
 220 NEXT I
 230 RAND USR 16514
 240 GOTO 230
```

THIS program flashes politely. As it is written, it will run successfully in 2K on the Timex Sinclair but not on the unexpanded ZX81; you will need to delete lines 10, 20, 50, 60, 70, 80, 90 and 100, then hit CONT to see the results. The string in line 210 may be any 32 characters you choose.

Particularly effective is alternating ▞ and ▚ in succession, i.e. ▞▚▞ ▚▞▚▞▚ etcetera.

The two machine code routines to do the scrolling are at locations 16514 and 16545. Submitted by Gary Szekeres of Erlanger, Kentucky.

(T/S1000,1500,ZX81 2K).



Feature

HEXEDIT: A Simple Hexadecimal Machine Code Editor

HEXEDIT is a short simple BASIC program that will allow you to write, enter and debug hexadecimal machine code routines with a minimum of frustration and wasted time. HEXEDIT makes the proofreading and correction of machine code programs especially easy since the one to four bytes of each Z80 instruction are stored in a separate REM statement; this produces an easily understood display like those found in larger disassemblers. Corrections can then simply be made with the user-friendly cursor-driven facilities of the Sinclair 8K ROM (LIST, EDIT, DELETE, cursor controls and so on). Even users who know little about machine language programming can use HEXEDIT to great advantage when typing in other peoples' programs. It requires only 2K of RAM for most applications.

HEXEDIT: The BASIC Version

Type in the program shown in Listing 1, then proofread and correct it carefully. Try out HEXEDIT by typing this short machine code test routine in REM statements into the main program:

10 REM D7

20 REM 18

30 REM FD

(You could also enter this at 10 REM D718FD.)

Save a copy in case it crashes, and then run it. The screen should go white for a few seconds and then give you the number of bytes of code that have been processed (in this case three). List the program and you should now see your translated code in line 1:

1 REM NOT / CLEAR. . . . . (etcetera)

Your three bytes of machine code have been translated and POKEd down into the 1 REM line; another copy has been placed above RAMTOP which will have been automatically re-set to 18176. If you make a mistake entering code in any of the REM lines after line 1, you can use the 8K ROM's cursor and editing functions to bring it down and make any corrections needed. Now execute these three bytes of code in line 1 by:

RAND USR 16514

in the immediate mode and the screen will fill with graphics "L"s. If this doesn't work you've made an error somewhere; find and correct it. Once everything is working properly, delete lines 10 through 30 and edit out the three-byte subroutine stored in the top line so it looks just like Listing 1. Save this on a fresh tape as HEXEDIT.

```
   1 REM .................................
..........
9700 FAST
9710 POKE 16389,71
9720 CLEAR
9730 LET L=16514
9740 LET H=18176
9750 LET P=L
9760 LET P=P+1
9770 IF PEEK P=118 AND PEEK (P+5
)=234 THEN GOTO 9800
9790 GOTO 9760
9800 LET P=P+5
9810 IF PEEK P<>234 THEN GOTO 99
00
9820 LET P=P+1
9830 POKE L,16*PEEK P+PEEK(P+1)
-476
9840 POKE H,PEEK L
9850 LET L=L+1
9860 LET H=H+1
9870 LET P=P+2
9880 IF PEEK P=118 THEN GOTO 980
0
9890 GOTO 9830
9900 PRINT "BYTES= ";L-16514
```

Listing 1. *HEXEDIT*

Inverse

Once you have tested and debugged HEXEDIT, you may use it to enter longer machine language subroutines. INVERSE is a useful 26-byte program which can be used in any BASIC program to quickly change every character on the screen to its inverse and vice versa. Load HEXEDIT from your previously saved tape and add the additional REM lines (10 to 160) shown in Listing 2. Do not type in the assembly-language mnemonics in the right-hand column, however;

```
 10 REM 2A0C40   LD HL,(D-FILE)
 20 REM 23       INC HL
 30 REM 0605     LD B,05
 40 REM C5       PUSH BC
 50 REM 0691     LD B,91
 60 REM 7E       LD A,(HL)
 70 REM C600     ADD A,00
 80 REM FE76     CP 76
 90 REM 2803     JR Z,03
100 REM C680     ADD A,80
110 REM 77       LD (HL),A
120 REM 23       INC HL
130 REM 10F3     DJNZ,F3
140 REM C1       POP BC
150 REM 10ED     DJNZ,ED
160 REM C9       RET
```

Listing 2. *INVERSE*

they are included for interest only. Make some typographical errors on purpose and then correct them (and any others) to see how easy this is. When you are fairly sure the code is error-free, save everything to tape just in case you crash. Make sure there are at least 26 characters after 1 REM so there will be enough space to store the program (a few extra does not hurt.)

Now run and you will get a report of BYTES=26. Write this down for later use and then press ENTER. You will see that your translated code for INVERSE has been placed in the top REM line. (Never try to edit this line after code has been transferred into it.)

Since you have all this stored on another tape, it is safe to try to execute the code for INVERSE: Just use:

RAND USR 16514

and 0/0 will appear if everything is okay. (INVERSE does nothing at this point since the display file is empty.)

You *could* isolate the code in the 1 REM line by deleting every single line below it, but this would be a rather unpleasant chore, especially with longer programs. Since HEX-

EDIT has automatically moved RAMTOP down to 18176 and has placed another copy of INVERSE above this, you can simply use NEW and ENTER to erase everything else. All the BASIC lines will be removed, but the code above RAMTOP will be preserved. If you don't believe this try:

PRINT PEEK 18176

and 42d, the first byte of INVERSE, will appear on the screen. Now type in DOWNLOAD, exactly as shown in Listing 3. Proofread it and make sure there are at least 26

```
  1 REM ...............................
  2 PRINT "HOW MANY BYTES?  ";
  3 INPUT N
  4 PRINT N
  5 LET H=PEEK 16388+256*PEEK 16389
  6 FOR I=0 TO N-1
  7 POKE 16514+I,PEEK (H+I)
  8 NEXT I
```

Listing 3. *DOWNLOAD.*

characters in line 1 after the REM. Also, make sure you are in the FAST mode. Now save DOWNLOAD to tape. (I find it is convenient to keep it on side two of my C-5 HEXEDIT tape.)

Run the program, input the number of bytes when prompted (26), and INVERSE will be transferred down into the 1 REM line of DOWNLOAD. Delete lines 2 through 8 and save your finished copy of INVERSE in 1 REM to tape.

Test the INVERSE routine by adding the BASIC lines shown in Listing 4. Run SCREENFILL and when the screen is full tap any key

```
  1 REM E£RND7▛VAL ▛LEN ▛?7(
NEXT HT (" GOSUB TAN
 10 SLOW
 20 LET A$=CHR$ INT (RND*64)
 30 FOR I=1 TO 6
 40 LET A$=A$+A$
 50 NEXT I
 60 PRINT A$;A$;A$( TO 192)
 70 IF INKEY$="" THEN GOTO 70
 80 LET X=USR 16514
 90 GOTO 70
```

Listing 4. SCREENFILL.

except BREAK and watch the screen "invert". If you have made a mistake simply reload the tape you saved of HEXEDIT with INVERSE included, correct the errors, and repeat downloading.

Always be sure there is enough space in the first line dummy REM statement to hold any code you are going to put there. Otherwise some very nasty things can happen. (If you don't believe this, try it!) It is a good policy to always save before running or a USR call when using HEXEDIT. It only takes a few seconds to save and it will avoid much (re)typing time.

HEXEDIT makes debugging machine code easy

Never try to bring down the 1 REM line itself for editing once it has been filled with code. If your program contains 7E, the five bytes that follow will simply vanish from your program and you will crash. (7E is the code for LD A, (HL); it occurs in most programs!)

With HEXEDIT and DOWNLOAD in your software library you can write and debug your own hexadecimal machine language programs or use it to enter other people's listings. Debugging and editing are much easier since the component bytes of each Z80 instruction are listed out to exactly the right length. Since both programs are so short, you can save or re-load in a matter or seconds as "insurance" against crashes. Once you have become familiar with these two routines, I think you will use them frequently and find them to be great time savers.

HEXEDIT: 16K RAM Version

You can increase the capacity of HEXEDIT if you plan to write and edit routines longer than 256 bytes and you have more than 2K of RAM. If you have 16K, for example, you can simply make the following changes in HEXEDIT:

9710 POKE 16389, 100

9740 LET H = 25600

This will reset RAMTOP to 25600 instead of 18176. DOWNLOAD doesn't need any changes since line 5 PEEKs RAMTOP for the starting address of the higher copy. Remember that whenever you use DOWNLOAD you will need to be sure that there are enough characters in the first line to hold all the bytes of your code. The price you will pay for this extended capacity is that loading and saving will take a bit longer; the 2K version is preferable for most applications of HEXEDIT.

HEXEDIT2: The Machine Code Version

You can use the BASIC 2K version of HEXEDIT to help produce a machine code version of itself! I have written this in 83 bytes, but I am sure other programmers will be able to improve upon this. HEXEDIT2, the machine code version of HEXEDIT, runs quite a bit faster and takes up a lot less RAM space. As presented, it operates nicely in 2K or more.

Load your previously tested and debugged tape of HEXEDIT. Now type in the REM statements containing the hexadecimal code for HEXEDIT2 (Listing 5) beginning with 10 REM FD360547. Do not type in the assembly language mnemonics in the right hand column. Proofread

```
 10 REM FD360547   LD(IY+05),47
 20 REM CD9A14     CALL 149A
 30 REM 110047     LD DE,4700
 40 REM 018240     LD BC,4082
 50 REM C5         PUSH BC
 60 REM 60         LD H,B
 70 REM 69         LD L,C
 80 REM 23         INC HL
 90 REM 7E         LD A,(HL)
100 REM FE76       CP 76
110 REM 20FA       JR NZ,FA
120 REM E5         PUSH HL
130 REM 23         INC HL
131 REM 23         INC HL
132 REM 23         INC HL
133 REM 23         INC HL
134 REM 23         INC HL
140 REM 7E         LD A,(HL)
150 REM FEEA       CP EA
160 REM E1         POP HL
170 REM 20EE       JR NZ,EE
180 REM 23         INC HL
181 REM 23         INC HL
182 REM 23         INC HL
183 REM 23         INC HL
184 REM 23         INC HL
190 REM 7E         LD A,(HL)
200 REM FEEA       CP EA
210 REM 201E       JR NZ,1E
220 REM 23         INC HL
230 REM 7E         LD A,(HL)
240 REM D61C       SUB 1C
250 REM 07         RLCA
251 REM 07         RLCA
252 REM 07         RLCA
253 REM 07         RLCA
260 REM C5         PUSH BC
270 REM F5         PUSH AF
280 REM 23         INC HL
290 REM 7E         LD A,(HL)
300 REM D61C       SUB 1C
310 REM 4F         LD C,A
320 REM F1         POP AF
330 REM A9         XOR C
340 REM C1         POP BC
350 REM 02         LD (BC),A
360 REM 12         LD (DE),A
370 REM 03         INC BC
380 REM 13         INC DE
390 REM 23         INC HL
400 REM 7E         LD A,(HL)
410 REM FE76       CP 76
420 REM 280A       JR Z,0A
430 REM 18E3       JR E3
440 REM 60         LD H,B
450 REM 69         LD L,C
460 REM C1         POP BC
470 REM A7         AND A
480 REM ED42       SBC HL,BC
490 REM 44         LD B,H
500 REM 4D         LD C,L
510 REM C9         RET
```

Listing 5. *Code for HEXEDIT2.*

and edit the listing very carefully and change line 1 so that it contains at least 83 characters after REM (more is okay). Now save everything on tape. Run and you should get a "bytes" report of 83; write this down. Now list and your code will be seen in the 1 REM line. Use NEW to erase all the BASIC lines, then type in DOWNLOAD2 as shown in Listing 6. We want to end up with out first line REM statement

```
9901 REM .......................
...............................
...........................
9910 FOR I=0 TO 82
9920 POKE 16514+I,PEEK (16176+I)
9930 NEXT I
```

Listing 6. *DOWNLOAD2.*

having a *high* line number, 9901 in this case, so our code for HEXEDIT2 can reside at the very end of the program, just as it did in the BASIC version. Make sure that there are *exactly* 83 characters after the REM in line 1. Now run. The 83 translated bytes of code will appear in line 9901. *Do not edit* this line under any circumstances. Delete lines 9910 to 9930 and add lines 1 and 9900 to produce Listing 7. Make sure you get line 9900 exactly right, as this chooses the starting address (D-FILE)-84) that the code in line 9901 is to be run from . Save these three lines on tape as HEXEDIT2. It should look just like Listing 7 when you are done.

```
   1 REM .......................
...........
9900 PRINT USR (PEEK 16396+256*P
EEK 16397-84)
9901 REM CLEAR ▆?LN ▆=) ?▆RANDU
▆ ???????????????▆VAL PRINT ?AT ▆
>▆<7 STOP ??AT ▆ GOSUB PI???TAN
```

Listing 7. *HEXEDIT2.*

You can test this program by entering the short routine described above.

10 REM D718FD

Running this should put NOT / CLEAR into the top REM line; RAND USR 16514 should fill the screen with "L"s, as before. I would suggest putting a copy of DOWNLOAD (Listing 3) on the second side of your HEXEDIT2 tape for convenient downloading. Remember that you will need to adjust the number of characters in the 1 REM line each time you use DOWNLOAD after NEW-ing away the BASIC lines. You should write this on your tape (or put it in a REM statement). As an exercise, try using HEXEDIT2 to produce a 1 REM line containing the code for INVERSE.

User's Summary of HEXEDIT and DOWNLOAD

1. Load HEXEDIT.
2. Enter hexadecimal machine code in REM statement between lines 10 and 9690.
3. Proofread these REM statements carefully and edit out any errors you have made in the code.
4. Add enough characters to line 1 so there is plenty of space for your code.
5. Save in case of a crash.
6. Run.
7. Record the number of bytes of code reported.
8. Test your code with RAND USR 16514 (or whatever the appropriate starting address is).
9. If it works, then erase all BASIC lines with NEW, ENTER.
10. Load DOWNLOAD from your tape.
11. Make sure the computer is in the FAST mode.
12. Adjust the number of dummy characters in line 1 so that it equals the number of bytes reported above.
13. Run DOWNLOAD.
14. Input the number of bytes when prompted.
15. Delete lines 2 to 8.
16. Save and label this 1 REM line on tape for later use in your own programs.
17. Execute the code with RAND USR 16514 or LET XXX = USR 16514 (or the appropriate starting address).

Modifying HEXEDIT2

Just as with HEXEDIT you can extend the capacity of HEXEDIT2 if you need more than 256 bytes and if you have 16K of RAM. Plug in your 16K RAM pack (be sure to disconnect the power supply first!) and load HEXEDIT2. Now carefully enter the following four lines in the immediate mode.

```
POKE 16389, 100
CLEAR
POKE PEEK 16396 + 256 * PEEK 16397 - 81, 100
POKE PEEK 16396 + 256 * PEEK 16397 - 75, 100
```

This version will automatically change RAMTOP to 25600. Save this to tape. DOWNLOAD (Listing 3) will work fine, even with the change in RAMTOP.

You can make the machine code in line 9901 of HEXEDIT2 unlistable by using

```
POKE PEEK 16396 + 256 * PEEK 16397 - 89, - 1
```

in the immediate mode. Now only lines 1 and 9900 will appear on the screen and you will have additional RAM space available for REM lines containing hexadecimal code. Even though you cannot see line 9901, it is still there and will be executed by line 9900. (This technique will work in either the 2K or 16K version.)

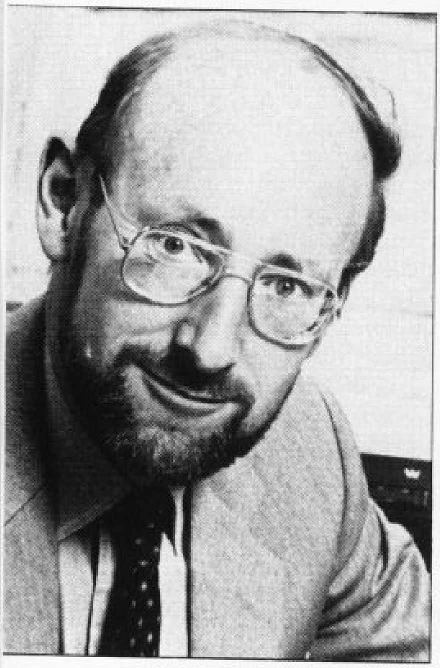
Either version of HEXEDIT with DOWNLOAD will save you many, many hours of debugging time and will increase the pleasure that efficient machine language programming brings to the Timex Sinclair user. I hope you will find these routines as helpful as I have and welcome hearing about any refinements you might make. ∅/∅

— *Dr. Dan Tandberg*

Electric car, mini-TV, keep Sinclair's researchers busy

LONDON — Meet Sir Clive Sinclair, the man who wants to put more money in his pocket by placing a television set in yours.

Not content with being president of the world's largest personal computer company, he will soon unveil five-centimeter, flat-screen televisions that can



Sir Clive: technological wizard.

be carried in your pocket.

After the flat-screen mini-television will come next year's project: an electric car. After that, who knows?

Undoubtedly Sir Clive has technological wizardry buzzing around in his head. Since he left school at 17, he has been fascinated by gadgetry and convinced that the world was ready for his inventions.

His first company produced stereo systems, then one of the world's first pocket calculators, then digital watches, then the first prototypes of flat-screen televisions. But the biggest seller has been the Sinclair home computers which helped to make him a multi-millionaire.

Now he is diverting some of his company's resources into an electric car, the first models of which should be available next year. The giant multinational automobile manufacturers have been toying with electric cars for years. Why does Sir Clive believe he will succeed where they have failed?

"The reasons we might hope to do something where large companies might not is that in a funny sort of way, large companies in any industry have a strong incentive not to change things," he said during a recent lunch with correspondents.

"They would like, because they have a large capital investment, to go on making the same sort of projects they make today . . . We don't have any such existing business so our intention is quite the opposite: to find some new niche, some new slot.

"The sort of slot we're looking for is a short-range vehicle for shopping or commuting which will be well-suited to the limits of the battery."

The lead-acid battery is the great stumbling block to designers of an electric car. When Sir Clive first began thinking about electric cars a decade ago, he was convinced the lead-acid battery would soon be obsolete.

Instead, the promised alternatives never developed, and Sir Clive's electric car will use the lead-acid battery. That will keep the car's range to 50 kilometers without recharging, enough to putter about in cities.

Without saying so — Sir Clive is reluctant to let his putative competitors in on any secrets — he is probably hoping the "niche" will be a second car selling for about $3,000.

Sir Clive's electric car will almost certainly contain mini-computers to regulate power. One report suggests the car will include a digital steering system using a microprocessor to calculate the angle of turn. The steering wheel would be replaced with a stick.

The miniaturization of computer technology is central to Sir Clive's success. After leaving school, he became a journalist specializing in technology for four years before starting his own company in 1962. That company sold radio and amplifier kits by mail order, later branching out into computer electronics.

In a country with a notorious resistance to change, he represents a small breed who caught the electronic revolution on the first wave. He is now an apostle for the second industrial revolution, for new technology, for entrepreneurial inventiveness.

"I have long believed that a necessary part of recovery in this, or indeed in any other Western country is an increase in new company starts. It (recovery) will not come from old companies re-employing labor when recession ends because they simply won't," he said.

"We have fallen into the habit of educating people to become employees, to see their futures with the great institutions — the banks, the great companies — and not to see their futures with small organizations started by friends or indeed by themselves. We need to return to educating people to be self-reliant . . .

"It's my opinion that semi-conductors are so much at the heart of the present industrial revolution, as shaping metals was in

Sinclair's electric cars may contain mini-computers to regulate power

the first industrial revolution, that to be without an indigenous semi-conductor facility within the nation is akin to being unable to forge metals in the first industrial revolution."

Britain is awash with U.S. and Japanese subsidiaries making chips and other bits of computer technology. Sir Clive prefers not to deal with them.

"My company is very proficient in semi-conductors and we have worked with British and American companies," he said.

"We find when we work with American companies, we work well when we deal with the head office in America. We don't get anywhere when we deal

News

with British subsidiaries. The control isn't here. They don't have freedom to act. The fact that they have subsidiaries here is no help at all."

Sir Clive is advertising for the "best and brightest" scientists to work at a new laboratory near Cambridge.

Called MetaLab, it will provide everything a scientist could need, including what Sir Clive describes as "shockingly high salaries." It will be financed by a share offering of 10 per cent of Sinclair Research which raised about $24-million.

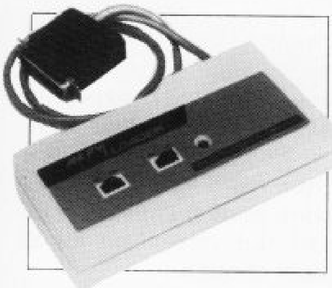
From the new laboratory — and from his three others — Sir Clive hopes new, practical ideas will pour forth. Color for the flat-screen television will be a priority. The existing Japanese ones — and the model Sir Clive will begin marketing later this year — are black-and-white. He lost the race to be first with the pocket television, although his model will be substantially cheaper than the Japanese products. He intends to be first in color.

— *Jeffrey Simpson*
Globe & Mail

Download from Radio

First we had tape cassette programs, then stringy discs, floppy discs and mini-discs, telephones, phonograph records and now the radio. Microperipheral Corporation has developed a receive only modem that operates at 4,800 baud rate designed to receive computer data from commercial radio systems.

According to the company, the device is simple to install. One cable plugs into the audio output jack of the radio. Another, terminated in a DB-25 connector, plugs into the serial port of the computer. The power



comes from a regular nine volt radio battery.

One pilot station, KMPS in Seattle, has been using the system since March. The station uses low listener times to broadcast computer data. The end user can either record the data on a tape for later use or take the information directly into the computer using the new modem. Why this

Data is broadcast on AM

need? Mike Oarland, president of Microperiphal says, "The advantage of the system is that anyone with a radio, our high-speed downloader and a computer may receive the transmissions. The system eliminates the need for telephone lines and main frame host computers."

Radio is only the start. Obviously cable television is next. A commercial cable operator would not have to wait for down times. The operator could use an open FM channel location and continuously send data out into subscribers' homes.

For more information write to Microperipheral Corporation, 2565 152nd Avenue N.E., Redmond, Virginia 98052 or circle 31 on the reader service card.

T/S2068 Display Primer

Author Fred Blechman explains how to get your computer to put things on the screen where you want them to go

YOUR Timex Sinclair 2068 is now hooked up and running, and you've practiced using some of the commands. You've probably even written a few simple programs with INPUT, PRINT and several other statements. You may have even done some calculations and had the results PRINT on the screen. But you've also discovered that computers seem to have a mind of their own about where they put things on the screen. The purpose of this article is to help you get control of your computer and make it put things on the screen where you want them! You'll also find out how to do some simple graphics.

Display Mode 1

The display looks like a big blank area when there's nothing on it but the little blinking cursor in the lower left corner. Actually, there are four display "modes" available with the T/S2068 computer. In this article we'll only be directly involved with Display Mode 1, since using the other modes is fairly sophisticated.

Note: This article is an excerpt from the author's new book, "Timex 2000 Beginner/Intermediate Guide" (Howard W. Sams and Co. Book #22225).

Display Mode 1 provides 24 lines of 32 character locations on each line, with the bottom two lines reserved for INPUT, EDITing and reports. However, you can also PLOT individual points — called "pixels" — in any of 256 locations across, and 176 locations up-and-down! For those of you weaned on the ZX81 or Timex Sinclair 1000 or 1500, this is *four times* as many locations in each direction, or 16 times more pixel locations (45056 vs. 2816). All these locations are shown in Figure 1.

Character & Pixel Locations

The first thing to understand is that in Display Mode 1 character locations each use eight pixels across by eight pixels high. That's why there are 256 pixel locations across the screen (32 characters × 8 pixels per character) and 176 pixel locations vertically (22 usable lines × 8 pixels per line). Another thing to understand is that *all* locations start with the first place designated as 0 rather than 1. That's why character locations on Figure 1 go from 0 to 31 horizontally and 0 to 21 vertically, and pixel locations run from 0 to 255 across and 0 to 175 vertically. But that isn't all! Character locations start in the upper left corner, counting across and down. Pixel locations, on the other hand, reflecting the normal Cartesian coordinate standard, start at the lower *left* corner and count across and *up*!

Why do you need to know all of this? You don't — unless you want to control your screen displays. With a few simple commands and Figure 1, you'll become the master. Read on.

To Scroll or Not to Scroll?

Type in this simple program to print on all 22 screen lines:

```
10 FOR X=1 TO 22
20 PRINT "Test"
30 NEXT x
```

Of course, the FOR, TO, PRINT and NEXT are keywords, not separate characters. Now, when you RUN the program, the word "Test" (without the quotation marks) appears at the left side of the screen on all 22 lines. Simple enough.

Now, just to see what happens when you try to put more than 22 lines on the screen at a crack — or whenever the next printed line would fall below the last user screen line — change the 22 in line 10 to 30. Now RUN and you get

"Test" on 22 lines, then a blank line and "scroll?". Hmmmm. What to do to get the rest? Just press any key (except N, STOP, BREAK or SPACE) and the display continues and then stops with an OK report. Actually, the "scroll?" has been carefully planned to keep you from losing screen data unintentionally by having it scroll up off the screen. If you don't want the "scroll?" at the end of a full screen, then simply add this line:

```
15 POKE 23692,255
```

Now when you RUN, the program goes to the end before it stops. The POKE simply tells the computer to PRINT 255 more lines before "scroll?". Since you've placed this in the program so that it is operative before the PRINT statement, the program is not even limited to 255 lines, since it is reset to 255 *every time* you go through line 15. This may seem unimportant to you now, but can be critical when printing long tabulations — and "scroll?" will also stop the printer if you're using one!

Print Formatting Using the Comma and Semi-colon

Two of the easiest-to-use and most powerful formatting commands are the comma and the semi-colon. Just change line 20 by adding a comma after "Test". Now RUN and you'll get two columns of the word "Test" running down the left and center of the screen, with 15 in

> PLOT INVERSE and DRAW INVERSE are like erasers

each column. What has happened is that the comma at the end of line 20 moves the printing to the next half-screen width. If you use two commas, that moves the next printing a full screen line. As a matter of fact, just to illustrate how commas can be used to space printing several lines apart (with very little memory use), change the 30 in line 10 to 8, delete line 15, and use *six* commas at the end of line 20 instead of one comma. Now the program is simply this:

```
10 FOR x = 1 TO 8
20 PRINT "Test",,,,,,
30 NEXT x
```

When this is RUN you get "Test" on display line numbers 0, 3, 6, 9, 12, 15, 18 and 21 (refer to Figure 1 for display line numbers). The six commas at the end of the PRINT statement on line 20 of the program have merely created two blank lines each time. After the last line, the "scroll?" appears, since the commas are trying to print two more blank lines.

To see the effect of a semi-colon, just remove the commas from the end of program line 20 and use one semi-colon instead. Now RUN —

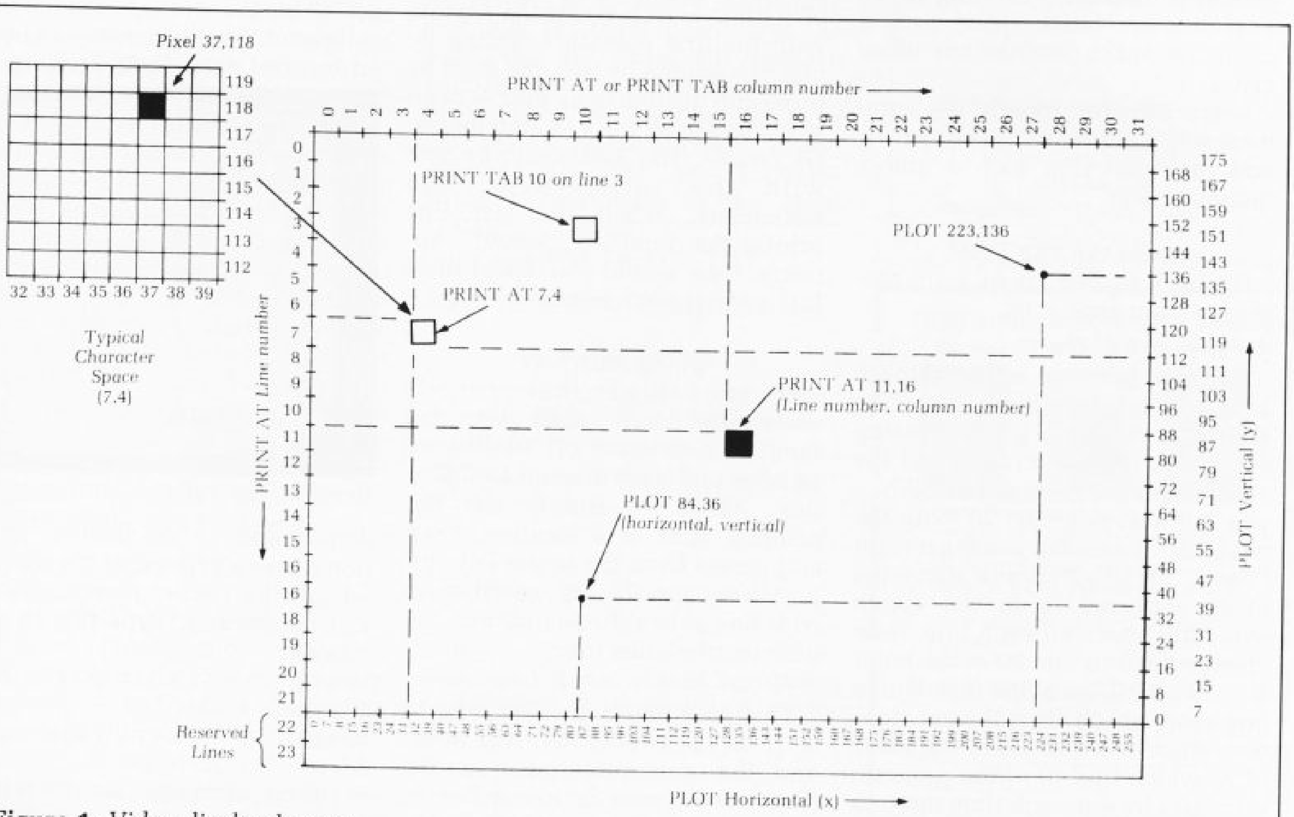


Figure 1. *Video display layout.*

Very simply stated, PLOT putteth and PLOT INVERSE taketh away

and all the "Test" words are on one line, with no spaces at all between words! The semi-colon at the end of a PRINT statement simply freezes the printing cursor at that location, unless overridden by PRINT AT or PRINT TAB, which we'll get to soon.

But what about blank spaces? Can they be used to format? Absolutely! Change line 20 of the program to add four blank spaces after the word "Test" but *before* the closing quotation mark, like this:

```
20 PRINT "Test    ";
```

When you RUN the program each "Test" is separated by four blank spaces. *Any* text, including blank spaces, between quotation marks in a PRINT statement, will be printed, even if it forces the printing to the next line. A blank space uses a character space just like any other character!

Learn how to properly use commas, semi-colons and blank spaces, and you're on your way to "punctuation power".

Take the TAB Test!

Go back to line 20 in your program and change it to:

```
20 PRINT TAB 8;"Test"
```

The TAB function is above the P key. You need to first access the E-cursor, which you get by pressing one of the CAPS SHIFT keys and the SYMBL SHIFT key together. When you RUN this new line 20 (with the old lines 10 and 30), you'll get eight "Tests" all lined up one above the other, indented in to the eighth character space on each line. Now change the 8 in line 20 to the letter X. When RUN, each line is indented one more space than the previous one, since TAB now takes the value of X, which line 10 of the program increases by one each time through the FOR-NEXT loop (lines 10 and 30).

Multiple TAB statements are common. For example, use this for line 20 in the program:

```
20 PRINT "Take"; TAB 8;"the"; TAB 16;"TAB"; TAB 24;"test. . ."
```

RUN and see how nicely each word is spaced. Can you see the value of this in tabulations? Of course, the *real* value is when you can use variables. Use NEW (the A key) and ENTER to erase the current program in memory, and then type in this program:

```
10 LET a=1: LET b=2: LET c=3: LET d=4
20 PRINT a; TAB 8;b; TAB 16;c TAB 24;d
30 LET a=a+1: LET b=b+2: LET c=c+3: LET d=d+4
40 GOTO 20
```

RUN and watch the four columns of figures march down the screen, with the first column counting by one, the second by two, the third by three and the fourth by four — neatly formatted by the TAB statements. Of course, you can combine text with the variables in TAB statements. When the last line prints, the familiar "scroll?" appears. How would you defeat that? Just add: 35 POKE 23692,255

Using PRINT AT

If you think that TAB is something, wait until you get familiar with PRINT AT! You'll need to refer to Figure 1 again to follow this. PRINT AT simply has the printing start at a location *down* and *across* from the upper left corner. You specify the coordinates with *line* (down) first, and then *column* (across), like this:

```
PRINT AT 20,15
```

This will cause the printing to start on the twenty-first line down and the sixteenth space across. (Remember, zero is a number in computing, and is used where you normally would start with one. Also, "lines" are frequently referred to as "rows".) To illustrate how PRINT AT can be used, delete your existing program with NEW and ENTER this program:

```
10 LET y=0:FOR x=0 to 21
20 PRINT AT y,x; "Test"
30 LET y=y+1: NEXT x
```

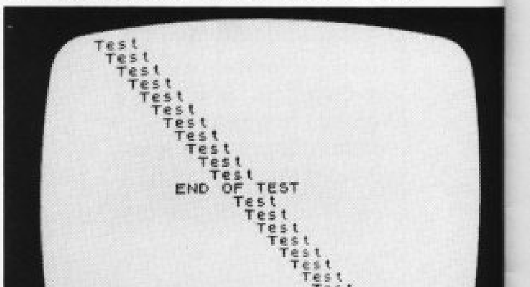
Be sure to use the SYMBL SHIFT and I key to get AT. RUN this and you'll get a string of "Test" words running diagonally down the screen. But to illustrate how PRINT AT can jump around the screen, and even erase previous printing, add line 40:

```
40 PRINT AT 11,7; "END OF TEST!"
```

See how it overprints the twelfth line (screen line 11)? Using blank spaces between the quotes, you can erase anywhere on the screen.

Displaying the Entire Character Set

You might want to see what each number means to the computer when it's used in the "character set". This is accessed with the CHR$ statement. This program can be used to print each CHR$ code on the



Result of the PRINT AT *test program.*

screen and on the printer. If you don't have a printer, or it's not turned on, the LPRINT commands will just be ignored. Type this in after NEW:

```
10 FOR x=32 TO 255 STEP 2: POKE 23692,255
20 PRINT x;" "; CHR$ x,x+1;" "; CHR$ (x+1)
30 LPRINT x;" "; CHR$x,x+1;" "; CHR$ (x+1)
40 PRINT : LPRINT : NEXT x
```

Feature

A couple of things are significant here. For one thing, most numbers under 32 represent "control codes". Not only will they *not* print out, some will stop the program — so start with 32, which is a blank space. Another significant thing is that the numbers and letters are standard ASCII (pronounced ASK-ee, and stands for American Standard Code for Information Exchange), unlike the T/S1000, 1500,ZX81. This is of prime importance when interfacing the computer to peripherals, such as printers, plotters, modems and so forth. When you RUN the program, the comma in lines 20 and 30 "zones" the printer so you get two columns, each one-half screen width. Notice the graphics from 128 (blank space, like 32) to 143. What appear to be a repeat of upper-case letters A to U (numbers 144 to 164) are actually locations where you can design and program any character shape to fit into an 8 x 8 matrix — but that's another whole subject! Notice that the keywords run from 165 to 255; did you realize there were so many?

High Resolution Graphics

There are three ways to produce graphics with the T/S2068. The "low-resolution graphics" are those available directly from the keyboard — the eight squares on number keys 1 to 8, and their inverse video counterparts (when the SYMBL SHIFT key is held down, or INV.VIDEO is used.) The so-called "high-resolution graphics" are the pixels referred to earlier. The "ultra-high" resolution color graphics are only available in Display Mode 4, not covered in this article.

Getting Pixel-ated

Figure 1 (again!) shows the coordinates used for pixels (picture elements — individual dots). The important thing here is that pixel coordinates are specified just like conventional graphs, with the x-axis horizontal, running left to right, and the y-axis being the left vertical side, running upwards. This is entirely different from the PRINT AT coordinates. This takes some getting used to, but makes sense when you think of it this way: PRINT AT is intended for text, which extends from top to bottom, left to right; pixels, on the other hand, are graphical in nature, and follow x-y graph convention.

Furthermore, since there are 64 pixels (8 x 8) in each character space the numbering system is entirely different. To illustrate the use of pixels, delete the existing program with NEW and type in this program:

```
10 LET y=0: FOR x=1 TO 175
20 PLOT x,y
30 LET y=y+1: NEXT x
40 LET y=0: FOR x=1 TO 175
50 PLOT INVERSE 1;x,y
60 LET y=y+1: NEXT x
70 GOTO 10
```

When you RUN this program, a thin line is drawn, one pixel at a time, from the lower left corner of the display to the top of the display (but only 176 pixel locations to the right, not the entire 256 spaces), and then the line is erased, pixel by pixel. (The T/S2068 does not have the UNPLOT statement used in the T/S1000,1500,ZX81, but "PLOT INVERSE 1;" amounts to the same thing, though a little trickier to use).

Two powerful formatting commands are the , and ;

Just for a little fun with sound (although it will slow down the program) add this:

```
45 BEEP .1×/8
```

Okay, Pardner — DRAW!

Just to give you a taste of the power of just one other graphical command, press the A key (NEW) and ENTER. Now type and ENTER this short program:

```
10 PLOT 0,0: DRAW 175,175
30 PLOT INVERSE 1;0,0: DRAW INVERSE 1;175,175
50 GOTO 10
```

Very simply stated, PLOT putteth and PLOT INVERSE taketh away. DRAW almost instantly creates a line from the PLOT starting point to an ending point x pixels to the right and y pixels up. Zap! PLOT INVERSE and DRAW INVERSE act like erasers. Press R for RUN and ENTER and you get a flashing line in exactly the same position as the previous program! Actually, the line is being

Characters may be shaped 18 billion-billion different ways

"drawn" and erased so quickly it appears to be flashing. To slow it down, so you can see how fast the DRAW command is, put in the following two lines:

```
20 PAUSE 30
40 PAUSE 30
```

The PAUSE is just that. The computer counts to itself the number of television frames going by (60 per second) and does nothing new in between — so PAUSE 30 gives you about a half-second delay. (Don't depend on PAUSE for precise timing, however, since computing time is involved, as well as television frames . . .)

Pixel Power

There is really hardly any limit to the detailed graphics you can design with the fine resolution of the T/S2068 in Display Mode 1. We haven't even touched on *color* (there are eight of those with two BRIGHTness levels for each), and have only mentioned special character shapes, of which there can be a choice (are you ready for this???) of 18 *billion-billion* possible shapes in *each* character space! Consider also that in Display Mode 2 you can have 64 characters or 512 pixels on a line! Now, let's see, 18 billion-billion possible shapes in each of 64 characters on a line, times 22 lines . . . hmmmmm . . . that's a lot of possible *different* screens! Ø/Ø

Programming Arcade Games continued

from page 13

The leftmost eight bits comprise the high order register.

Register Pair Example

Suppose the BC register pair contained the following information:

BC = | 01000000 | 10000010 |

The contents of BC can be converted to decimal and hexadecimal.

Decimal	*	*Hexadecimal*
BC	*	BC
64 | 130	*	+40 | +82
BC = 64 × 256 + 130 = 16514	*	BC = +4082

The decimal and hexadecimal numbers are equivalent.

$+4082 = 4 \times 16^3 + 0 \times 16^2 + 8 \times 16^1 + 2 \times 16^0$
$= 16514$

Getting Ready for Machine Language

In a previous example you saw the machine language instruction: LD A,4. This is actually a mnemonic for the instruction. Machine language is entered into the computer as a series of numbers. Two numbers are entered for this instruction.

Code	*Mnemonic*
62	LD A,4
4	

Writing a machine language program purely as numbers would be almost impossible to understand. For this reason, mnemonics are always written beside the code.

You have now covered the background material necessary for machine language programming. In the next chapter you will learn the entire machine language instruction set. You will soon be writing your arcade games in fast moving machine language. Ø/Ø

Reader Survey

Help us give you the magazine you want

We want to give you the magazine you want! By filling out this questionnaire you do two things. First, you let us know who you are so we can make T.S.U. more to your liking. The more we know about you, the better T.S.U. will be. Second, you get a chance to win a T/S 2040 printer. By completing this form and mailing it back to us (postage paid, of course) your name will be entered into a draw for a T/S 2040 printer — five will be given away.

3 STEPS TO WINNING A T/S 2040

1. Fill in the questionnaire.
2. Mail it back to us.
3. Watch the February, 1984 issue for the announcement of winners.

Fold down here

1. What machine(s) do you own?
( ) ZX 81 ( ) T/S 1000
( ) T/S 1500 ( ) T/S 2068
( ) ZX Spectrum
Other (Please Specify) ______

2. What "add-ons" do you own?
( ) 16K Ram ( ) 64K Ram
( ) Keyboard ( ) ZX Printer
( ) T/S 2040 Printer ( ) Tape Interface
( ) Joystick
Other(s) (Please Specify — Use Back if Needed) ______

3. Where do you use your computer?
( ) Home ( ) Business
( ) School ( ) Other ______

4. What do you use your computer for? (Check more than one box if needed)
( ) Business ( ) Home Management
( ) Personal Materials Management (e.g., records, addresses, recipes, etc.)
( ) Games ( ) Education
( ) Computer Programming
Other (Please Specify) ______

5. Which issues of Timex Sinclair User have you read?
#1 ( ) Collector's Issue
#2 ( ) "Kids and Computers"
#3 ( ) "TSU Goes to Camp"
#4 ( ) "Two New Arrivals"
#5 ( ) "Color for the T/S 1000
#6 ( ) "Get Organized with Filing Software"

6. How many people, including yourself, will read this issue of T.S.U.?
Print Number: ______

7. How did you acquire this copy of T.S.U.?
( ) Subscription ( ) Newsstand
Other (Please Specify) ______

8. How old are you?
( ) Under 16 ( ) 16 to 19
( ) 20 to 24 ( ) 25 to 34
( ) 35 to 50 ( ) Over 50

9. What is your household's income?
( ) under $15,000 ( ) $15,000 to $24,999
( ) $25,000 to $34,999 ( ) $35,000 to $49,999
( ) over $50,000

10. What is your sex?
( ) male ( ) female

11. What is your marital status?
( ) single ( ) married

12. What is your education?
( ) Attended/Graduated from High School
( ) Some College ( ) College Graduate
( ) Some Graduate School ( ) Graduate Degree

13. What is your area of work?
( ) Professional ( ) Managerial
( ) Sales/Marketing ( ) Student
( ) Other

14. How much have you spent on software over the past 12 months?
( ) under $100 ( ) $101 to $200
( ) $201 to $300 ( ) over $300

15. How much have you spent on hardware over the past 12 months?
( ) under $100 ( ) $101 to $200
( ) $201 to $300 ( ) over $300

16. How much have you spent on books over the past 12 months?
( ) under $100 ( ) $101 to $200
( ) $201 to $300 ( ) over $300

17. What other computer magazines do you read?
______

18. Do you subscribe to a computer magazine?
( ) yes ( ) no

19. If your answer to number 18 is yes, please list the magazine(s).
______

20. How do you rate Timex Sinclair User?
( ) Excellent ( ) Very Good
( ) Good ( ) Average
( ) Poor ( ) Very Poor

Fold up here

Please enter my name in the draw for a T/S 2040 printer.

Name ______
Address ______
City ______ State ______ Zip ______

Do you subscribe to Timex Sinclair User magazine? ( ) Yes ( ) No

Please feel free to return the questionnaire without completing the details of your name and address.

WIN
ONE OF FIVE
TIMEX SINCLAIR 2040 PRINTERS —
Details on Reverse.

timex sinclair user

NO POSTAGE NECESSARY IF MAILED IN THE UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 282 BUFFALO NY

POSTAGE WILL BE PAID BY

READER SURVEY
Timex Sinclair User
49 La Salle Avenue
Buffalo, New York 14214 - 1414

No one with a Timex Sinclair Computer should be without Timex Sinclair User

Computerized Dee-Jay

Jack Hodgson explains how to set up a music library using your computer and "The Organizer" from Timex

MY ROOMMATE Mark collects "Top 40" singles. He owns over 3,500 45 rpm records. Some were hits as long ago as the 1930s, but most are from the last 20 years. In addition to maintaining this collection for his own enjoyment, he uses it to earn money as a dance deejay. He, his sound system, and his 3,500-plus records hire out to dances, parties, and other assorted functions.

Over the years Mark has created a file of his collection using 4-in. by 6-in. index cards. In addition to the title of the record and the artists' names, these cards contain information about when the record was released, when it entered the top 100, when it dropped from the top 100, the highest position it reached, and cross references of performers who were with other groups. Mark keeps this file in alphabetical order by artists' name. The file occupies two and a half 15-in. file drawers.

A few months ago Mark created a data base of his collection using The Organizer program from Timex. Although it would be valuable to be able to update and print out the file more easily, his main goal was to be able to search through the date and top position information. Often, when he was playing records at a party a guest would come to him with a request like this: "We graduated from school in 1972. Can you play some songs that were hits that year?"

In the past Mark dealt with these requests from his own knowledge of music. And although this knowledge is amazingly comprehensive, it is not totally complete.

In creating the file Mark realized right away that it was going to be difficult, if not impossible, to code 3,500 records into 10,000 bytes of storage space (DIM F$(9999)). He dealt with this problem in the time-honored fashion of deciding to think about it later. In the meantime he resigned himself to storing the information in more than one cassette tape loading.

The next question was how to format the file. He set up the screen rather straightforwardly. The first field contains the artist's name, the second is a cross reference, when needed, to other groups. Subsequent fields alternate between one with a song title and one containing the year and position information. On his index cards numbers are simply separated by slashes (/). He knew from memory which column

ILLUSTRATION • Rochelle Rubinstein-Kaplan contained which piece of information. To make data base searches possible he added letter postscripts to the numbers. Without the letter a search for "65" could be interpreted as "entered chart at 65" when it really meant "released in 1965". In the data base "65N" means the former and "65Y" the latter.

As it stands the program does the job expected of it. It can search for and find entries by song title, top position, year released, and so on. It can reorder the file by any field and it can be easily updated. The main problem now is that, in order to do a complete search the program must be loaded several times. Each loading contains part of the data base. Mark is currently looking for solutions to this problem.

Here are a few of the solutions under consideration. Reduce the artist's name entry to a one or two byte number which can then be looked up on an external index. Encode common words such as "love", "you", "him", "her" into a single byte which is decoded and printed correctly on the display. Mark is also looking at various more sophisticated data compression techniques. The last two solutions will, of course, require changes to the Organizer program. But then it's not written in stone!

The program can find entries by song title, top position on the charts, year released and so on

Perhaps the ultimate solution to these problems is to add hardware. A 64K RAM, a stringy floppy, or a full-fledged disk drive would do the job easily. But I'd like to think that the first way to solve any problem is through efficient software.

So there's Mark's Organizer application. It's not perfect but then it beats the hell out of those index cards. I'd be real interested to hear about other applications of The Organizer. Write to me care of this publication.

Regarding PEEK

MOST first-time programmers (and for that matter a great many programmers unfamiliar with PEEK and POKE) regard these two commands with dark foreboding. Often lack of interest in these useful functions comes about from the "if you don't want to get burnt, don't stick your fingers in unfamiliar territory" syndrome. Regarded as very mysterious, the PEEK command, as its name suggests, allows the programmer to look at the contents of a specified memory location. Its opposite, POKE, allows the programmer to place a value in a specified memory location, which will vary depending on available RAM; the greater the RAM the higher the number of memory (POKE) locations. Both commands have fixed formats composed of two variables — the memory address and the contents of that address (or location). The address variable (designated V in the following examples) must be an integer value in the range 0 to 65,535. The contents variable, called C, is also an integer variable and has the numeric range 0 to 255. The commands are used as follows:

```
LET C = PEEK V
```

which effectively loads the contents of memory address V into the variable location C and

```
POKE V, C
```

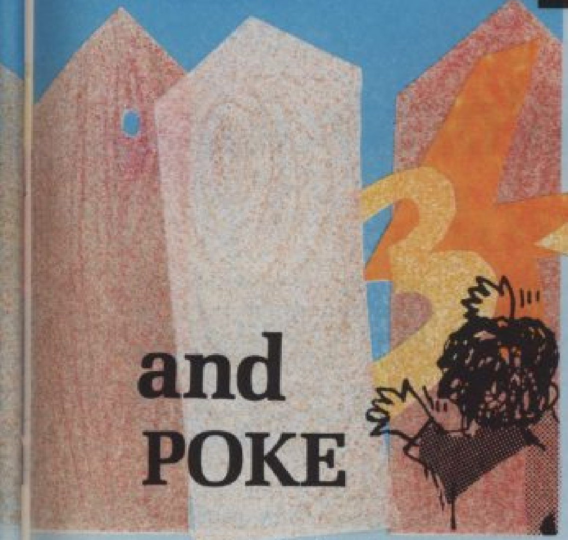
which places the value in variable C into memory location V.

On the T/S1000,ZX81 the monitor — the controlling machine code program stored in ROM, which controls the machine's keyboard, display and permits programs to be written in BASIC — is located in the bottom 8K of memory, i.e. in the bottom 8192 locations. The RAM memory which is used to store user programs and data, starts at location 16384 and, if the machine has been expanded, can go up to 32767. The contents of both types of memory can be looked at with the PEEK command, but only the contents of RAM memory can be changed with the POKE statement.

If you have more than 3.5K of memory available, then the PEEK and POKE commands can be used to great effect in generating video displays. The video display on the T/S1000 being stored as a file of 24 character strings, each 32 characters long and terminated with an ENTER character (marker). This 793 byte long display file is stored in RAM memory at a starting address contained in locations 16396 and 16397 and an end address stored in 16400 and 16401. The following program will print out the first 21 bytes in ROM and their addresses:

```
10  PRINT "Address: "; TAB 8; " Byte"
20  FOR A = 0 TO 20
30  PRINT A; TAB 8; PEEK A
40  NEXT A
```

Complementing the previous program, the next will print out all of the ASCII character codes of the current contents of each location on the screen. Such a short utility program is particularly useful for



and POKE

organizing display data, especially when using machine code which is essentially the manipulation of address registers.

```
10 LET P = PEEK 16396 + 256 *
   PEEK 16397
20 LET Q = (PEEK 16400 + 256 *
   PEEK 16401) - 1
30 LET T = Q - P
40 DIM A(T)
50 FOR K = P TO Q
60 LET A(K-P) = PEEK(K)
70 NEXT K
80 FOR K = 1 TO T
90 PRINT A(K)
100 NEXT K
```

In kind with the above program, the following program permits the user to POKE characters onto the display. Just as the above program will give the ASCII codes of each character stored in the display file, we can use this next program to place characters on the display:

```
10 LET P 5 PEEK 16396 + 256 * PEEK
   16397
20 LET Q = PEEK 16400 + 256 * PEEK
   16401
30 FOR K = 1 TO 255
40 IF (K/33) - INT(K/33) = 0 THEN
   GOTO 60
50 POKE K + P,15
60 NEXT K
```

Change the value to be POKEd into memory in line 50 and the resulting screen display will change. Line 40 has been included to 'protect' the ENTER characters at the end of each line in the display file.

T/S2000

Memory organization on the T/S2000 is not as straightforward as the T/S1000; in fact, the whole memory appears, at first sight, to be a confusion of addresses. The apparent randomness of memory can be infuriating and extremely frustrating when constructing machine code programs. Where a memory-mapped display allows the character in a given screen position to be determined by a simple PEEK into memory, PEEKing into the T/S2000's memory reveals no useful information as all is completely chaotic. The memory itself is divided into different 'sections', handling different functions and for storing varying types of information. These 'sections' are large enough only for the information they actually contain. If, by inserting more information in a given area, adding a further program line or variable, no more space is available, the computer 'makes' extra space by shifting up everything above that given area. Similarly, if information is deleted, everything is shifted downwards — the directions up and down referring to increasing or decreasing memory addresses.

The display file is that section which handles the television picture, the actual screen output. Each character position on the screen is composed of an 8 x 8 matrix of dots and each dot can be either 0 (paper) or 1 (ink), alternatively on and off. By using binary notation a pattern of dots can be stored as eight bytes, one for each row of dots. Herein lies the complexity of T/S2000 memory because these eight bytes are not stored together. The corresponding rows in the 32 characters of a single line are stored together as a scan of 32 bytes — the electron guns require this when scanning from left to right across the screen. Since the complete picture has 24 lines of eight scans each, you would expect the 172 scan total to be stored in order, one after the other. You would be misled to think so! First come the top scans of lines 0 to 7, then the next scans of lines 0 to 7 and so on to the bottom scans of lines 0 to 7.

PEEK allows you to inspect the contents of a memory location

POKE allows you to insert a value into a memory location

The same rulings apply to lines 8 to 15 and then for lines 16 to 23.

A little experimentation reveals that the eight bytes forming the character at the top left-hand corner of the screen are held in memory addresses 16384, 16640, 16896, 17152, 17408, 17664, 17920 and 18176, with 16384 starting the display memory area and the others being obtained by adding successive 256s. For example:

Address = 16384 + 256*(n-1) + 32*x + y = 16128 + 256*n + 32*x + y

The address contains the nth byte of the character in line x, column y. But, as mentioned, because of the odd layout, this equation fails after eight screen lines have been looked into with an entirely new block starting 2048 addresses after the first. So:

Address = 18176 + 256*n + 32*(n-8) + y = 17920 + 256*n + 32*x + y

Again, after a total of 16 lines a further new block commences:

Address = 20224 + 256*n + 32*(n-16) + y = 19712 + 256*n + 32*x + y

Obviously therefore, when writing machine code programs or constructing user graphics with PEEK/POKE commands, care has to be taken to determine exactly where characters will be stored so that characters that are to manipulated are, in fact, those and not others elsewhere on the screen.

— Aldo G. Rabaiotti

∅/∅

Joysticks to the World

What is it besides bulky size and fancy price tags that other computers have that our own T/S doesn't? Joysticks — those marvelous little devices that spice up any game. Well our cup runneth over because joysticks are available for the T/S. No longer need we cramp our fingers using movement keys that are too close together.

To begin with, what exactly is a joystick? A joystick is a moveable control device shaped like a paddle, used to control movement in computer games. How do they work? Joysticks are connected to your computer in conjunction with a digital analog converter (which may be found in the computer or in the joystick itself), thus actually allowing your digital computer to be used as an analog device.

I tested two joysticks: the Atto Soft and the Atari. Essentially both look, feel, and handle the same. The Atari is relatively easy to install with the use of the Zebra Joystick Adaptor. This adaptor will work with most Atari joysticks and it is simple to use. To install, plug the joystick into the connector on the Zebra adaptor. The adaptor then plugs into the rear edge connector on your computer, and there is a rear edge connector on the adaptor that accommodates other peripherals.

The Atto Soft joystick is initially more complicated to use as it requires some assembly to install. Assembling the Atto Soft involves removing approximately one inch of the outer insulation from the joystick cable, then removing one-quarter-inch of the insulation from each of the six wires in the cable. You have to tin the ends of each wire to prepare them for soldering on to the printed circuit board. After removing the computer case, solder each of the color-coded wires to diagrammed points on the board. Route the main cable out through the case, re-assemble the case, and you're ready to go.

Both the Atari and Atto Soft joysticks offer increased control and accuracy when playing video games. Once installed they are easy to manage and are self-centering. They each have one control button as well as the paddle. I found these two joysticks to be equally satisfying in performance and maneuverability. Were it not for the more complicated assembly procedure required to install the Atto Soft, these two joysticks could be deemed equal in all respects.

Not many joystick games are yet available for the T/S computer, but they are on the way. To convert your regular software games to joystick games, you must change subroutines within the program. Warning: Directions for program modifications are included in both joystick packages. Or if you prefer, you can always write your own.

— ***M.K. Wilson***

What'll They Think of Next?

It had to come! It's one of those things that make you say, "Why didn't I think of that?" They're called Hand Saver and Octoputs and are add-ons for one of your add-ons, the joystick.

The Hand Saver is a soft, rubberized sleeve that fits over the "stick" part of the joystick. After slightly moistening the inside of the sleeve, you push it onto the lever of your joystick. It's a snug fit and once on it is difficult to remove. But it does what the name implies, it saves your hand from slipping, sweating and shifting.

The Octoputs are four, clear, soft plastic suction cups. You remove the feet from the bottom of the joystick and insert these in their place. Moisten the bottoms of the Octoputs and press the joystick down onto any flat surface. You now have one-handed control. The joystick will not move. These things really hold. When you are ready to put the joystick away, simply peel the suction cups up using the tabs provided on them. It's a simple concept that does the job.

They can be bought as separate items or in single package. The samples sent to us had no pricing information but they look like under $10 items. For more information about Octoputs or Hand Saver, write to Electrocomp Distributing Inc., Centerline, Michigan 48015 or circle 27 on the reader service card.

How to Build an Inexpensive Joystick Interface

It takes just $10 and a few hours, writes James C. Righter

WHILE the low cost of the T/S1000,ZX81 is a strong selling point, not having a joystick input is one of the drawbacks of a no-frills machine. Fortunately, a joystick input can be added for a low cost (less than $10) and a few hours' time. Some decent games are on the market for the T/S,ZX, or you can write some, that make good use of a joystick control. In particular, two games sold commercially, Mazogs and Flight Simulation, seem to be naturals for a joystick control. After playing these two using the keyboard controls, I decided that a joystick control would be a great improvement.

First we will look at the interface circuit, then some example programs.

Circuit Details

Only three integrated circuits and a handful of parts are needed to construct the interface circuit. All materials are readily available at Radio Shack stores with the exception of an Atari* joystick or one of its clones.

The Atari* joystick was chosen because of its simplicity of having switches instead of variable resistors. The joystick has five normally open switches that are closed whenever the stick is moved or the "fire" button is pressed. Moving the stick to the top or forward position closes one switch, while moving the stick back to the bottom position allows the top switch to open and closes the bottom switch. The same goes for left and right movement of the stick and when the button is pushed. All switches are open when the stick is in the neutral position and the button is not pressed.

A block diagram of the interface circuit is in Figure 1, and consists of a one-of-eight decoder (74LS138), an or gate (74LS32) and an octal non-inverting buffer (74LS244). The first integrated circuit (IC1) functions as an address decoder for lines A13 through A15 of the T/S,ZX. IC1 is enabled when the memory request line ($\overline{MREQ}$) from the computer goes low. When this happens one of the eight output

*Atari is a registered trademark of Atari Inc.

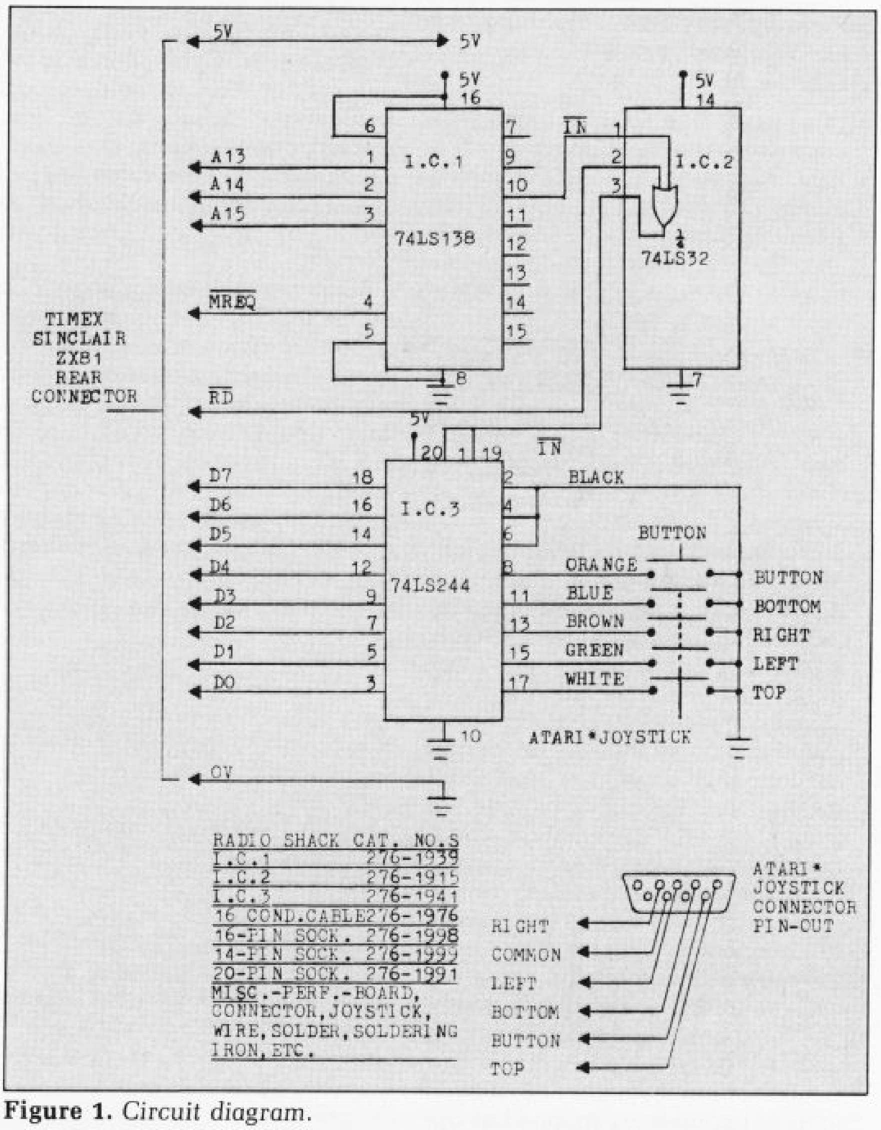


Figure 1. *Circuit diagram.*

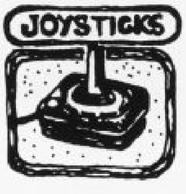
lines of the decoder chip goes low (15 through 9 and 7). Which of the decoders output lines goes low depends on which addresses within an 8K segment are chosen. Each segment or 8K block of addresses and the status of the decoders output lines are shown in Figure 2. In

Segment	Address			Line # (n)
1	0	to	8191	15
2	8192	to	16383	14
3	16384	to	24575	13
4	24576	to	32767	12
5	32768	to	40959	11
6	40960	to	49151	10
7	49152	to	57343	9
8	57344	to	65535	7

Figure 2. *Any address within these 8K segments makes line (n) low. Decoder I.C. 74LS138.*

this circuit line 10 was used so the addresses were well above the 16K RAM addresses. Line 10 from IC1 along with the read-enable line ($\overline{RD}$) from the computer are combined in an or gate of IC2 to form the enable line ($\overline{IN}$) that turns on the three-state

Mazogs and Flight Simulation are naturals for joystick control



buffer (IC3). Normally the input buffer is in a high impedance state until the correct address, a memory request signal and a read-enable signal are output from the computer. Only when all of these signals are present on the computer's buss will the data lines from the computer "read" the input from the buffer. Information read in from the buffer can then be stored in memory for later use or discarded as the program dictates.

There are probably as many ways to decode the address lines from the computer as there are hardware designs to do so, but this is a simple, inexpensive scheme that works well for this purpose. This particular scheme is generally referred to as being memory mapped as opposed to input/output mapped. Machine language must be used with the T/S1000,ZX81 when input/output decoding is used because there are no specialized instructions contained in the T/S,ZX BASIC language for that purpose. In contrast, memory mapped decoding will work with PEEKs and POKEs (in BASIC) as well as machine language. The circuit that has been discussed here will also work with the 16K RAM unattached.

Any convenient way of constructing the circuit is fine, as there is nothing at all critical about the layout. One of the universal printed circuit boards from Radio Shack works well, or you may find perfboard more to your liking. Whichever way you choose to wire the circuit, be it point-to-point, wire-wrap, or a custom made printed circuit board, it is best to use sockets for the integrated circuits. This eliminates the danger of ruining an integrated circuit while soldering.

Sixteen conductor ribbon cable can be used to attach the completed circuit to the computer. Each conductor or wire should be soldered very carefully to the appropriate buss line of the computer. The T/S,ZX circuit board should be removed from the case for this operation, so the wires can be attached at least a one-half inch from the edge of the board. This will allow the RAM connector room to

Listing 1.

```
10 LET A=PEEK 46000                Address of input port
20 IF A=31 THEN GOTO 10            If in neutral, try again
30 SCROLL
40 IF A=30 THEN PRINT "UP"         30=up pos. of stick
50 IF A=29 THEN PRINT "LEFT"       29=left pos. of stick
60 IF A=27 THEN PRINT "RIGHT"      27=right pos. of stick
70 IF A=23 THEN PRINT "DOWN"       23=down pos. of stick
80 IF A=15 THEN PRINT "BANG"       15=button pressed
90 GOTO 10
```

slide onto the edge of the computer's edge connector. A slot can then be cut in the side of the case (a heated knife works well) for the cable or wires to exit. Keeping the ribbon cable as short as possible will alleviate the interference problem of the television display.

Listing 2.

```
2  REM XXXXXXXXXXXXXXXXXXXX-
   XXXXXXXXXXXXXXXXXXXXXXXX-
   XXXXXX
3  PRINT "ENTER STARTING AD-
   DRESS"
4  INPUT A
5  PRINT "ENTER CODE IN
   DECIMAL"
6  INPUT B
7  POKE A,B
8  SCROLL
9  PRINT A, PEEK A
10 LET A=A+1
11 GOTO 6
```

Keep in mind that any soldering or cutting on the T/S,ZX will void the warranty. If the appropriate connectors can be located or made for a feed-through buss, the case need not be opened or any soldering done on the computer circuit board.

Attaching the joystick to the interface circuit can be accomplished with a detachable plug, a header, or hard-wiring directly to the circuit. The diagram in Figure 1 illustrates how the Atari* joystick should be wired to the interface circuit. The color-coded wires are exposed by cutting off the female connector at the end of the joystick cable and stripping back the outer cover. Each colored wire can then be used as a guide for connecting the joystick to the interface circuit. After the circuit has been assembled and all soldering has been completed, check the circuit for solder bridges and shorts before applying power.

Example Programs

If you are satisfied that your interface circuit is in good working order, turn the computer power on and check if the cursor is on the

Listing 3. *Code to Mazogs program.*

Address	Dec. Code	Mnemonic
16606	205, 221, 81	CALL 20957
16609	201	RET

Write code to #2 REM statement with 50 Xs.

Address	Dec. Code		Mnemonic
20957	58, 0, 176		LD A, (46000)
	254, 31		CP 31
	40, 28		JR Z, EXIT
	254, 30		CP 30
	32, 3		JR NZ, LEFT
	62, 8		LD A, 8
	201		RET
	254, 29	LEFT	CP 29
	32, 3		JR NZ, RIGHT
	62, 96		LD A, 96
	201		RET
	254, 27	RIGHT	CP 27
	32, 3		JR NZ, DOWN
	62, 80		LD A, 80
	201		RET
	254, 23	DOWN	CP 23
	32, 3		JR NZ, EXIT
	62, 6		LD A, 6
	201		RET
	167	EXIT	AND A
	33, 255, 255		LD HL, 255,255
	237, 75, 37, 64		LD BC, (16421)
	237, 66		SBC HL, BC
	124		LD A, H
	133		ADD A, L
21004	201		RET

Listing 4. *Code to Flight Simulation program.*

Address	Dec. Code	Mnemonic
17316	205, 247, 75	CALL 19447

Write code to #2 REM statement with 50 Xs.

Address	Dec. Code		Mnemonic
19447	58, 0, 176		LD A, (46000)
	254, 31		CP 31
	40, 32		JR Z, EXIT
	254, 30		CP 30
	32, 4		JR NZ, DOWN
	33, 239, 239		LD HL, 239, 239
	201		RET
	254, 23	DOWN	CP 23
	32, 4		JR NZ, LEFT
	33, 239, 223		LD HL, 223, 239
	201		RET
	254, 29	LEFT	CP 29
	32, 4		JR NZ, RIGHT
	33, 247, 223		LD HL, 223, 247
	201		RET
	254, 27	RIGHT	CP 27
	32, 4		JR NZ, EXIT
	33, 239, 247		LD HL, 247,239
	201		RET
	205, 187, 2	EXIT	CALL KSCAN
19489	201		RET

left-bottom part of the display screen. If after a moment or two the cursor does not appear on the screen, immediately disconnect the power and check the circuit for wiring errors, shorts, and so on. When the interface circuit is working correctly, enter and run the test program in Listing 1. For the programs in this article to work, the joystick must be wired as shown in Figure 1. If a problem still exists with running the test program, again power down and check the circuit for wiring errors and shorts.

Of course you may choose to wire the joystick differently, but the programs used here must be changed to reflect those differences.

Programs in Listings 3, 4 and 5 are written in machine code and

Listing 5. *Linedraw with joystick (150 bytes).*

ADDRESS	*DEC. CODE*	*MNEMONIC*	*COMMENT*
16514	42, 123, 64	START: LD HL, (16507)	*Coords plot pos.*
(Addresses used for locating code in REM *state.* Can be located anywhere.)	58, 176, 179	LD A, (46000)	*Load A port status.*
	254, 15	CP 15	*Check button status.*
	40,112	JR Z PLOT/UNPLOT	*If button pressed.*
	254, 27	RIGHT: CP 27	*Check stick status*
	32, 9	JR NZ, LEFT	*Check left.*
	44	INC L	*Add one to horizontal plot position.*
	125	LD A,L	
	254, 64	CP 64	*Check if off screen.*
	32, 40	JR NZ,PLOT	*If not got to plot.*
	45	DEC L	*Subtract 1 from plot position*
	24, 37	JR,PLOT	
	254, 29	LEFT: CP 29	*Check stick status.*
	32, 9	JR NZ,UP	*Go to up.*
	45	DEC L	*Subtract 1 from plot position.*
	125	LD A,L	
	254, 255	CP 255	*Check if off screen,*
	32, 27	JR NZ,PLOT	*If not go to plot.*
	44	INC L	*Add 1 to plot pos.*
	24, 24	JR,PLOT	
	254, 30	UP: CP 30	*Check stick status.*
	32,9	JR NZ,DOWN	*Check down.*
	36	INC H	*Add 1 to vertical plot position.*
	124	LD A,H	
	254, 44	CP 44	*Check if off screen.*
	32, 14	JR NZ,PLOT	*If not go to plot position.*
	37	DEC H	*Subtract 1 from plot position.*
	24, 11	JR,PLOT	

Listing 5. *continued from page 57*

ADDRESS	DEC. CODE	MNEMONIC	COMMENT
	254, 23	DOWN: CP 23	Check stick status
	32, 7	JR NZ, PLOT	Go to plot.
	37	DEC H	Subtract 1 from plot
	124	LD A,H	position.
	254, 255	CP 255	Check if off screen,
	32, 1	JR NZ,PLOT	If not go to plot.
	36	INC H	Add 1 to plot pos.
	68	PLOT: LD B,H	Load coords from
	77	LD C,L	HL into BC.
	34, 123 ,64	LD (16507),HL	Store plot coords.
	237, 67, 54, 64	LD(16438),BC	Last point plotted.
	62, 75	LD A,75	Plot/unplot status.
	50, 48, 64	LD(16432),A	T-ADDR
	205, 178, 11	CALL ROM	Plot routine in ROM
	17, 0, 9	LOOP: LD DE,900	Determines delay.
	27	DELAY: DEC DE	Delay loop.
	122	LD A,D	
	179	OR E	
	32, 251	JR NZ,DELAY	
	58, 33, 64	LD A,(16417)	Plot/unplot status.
	254, 0	CP O	If p/u not changed,
	32, 12	JR NZ,EXIT	GOTO to exit routine.
	237, 75, 123, 64	LD BC,(16507)	Retrieve plot coords.
	62,0	LD, A,0	Change plot to unplot.
	50, 48, 64	LD (16432),A	T-ADDR
16615	205, 178, 11	CALL ROM	Plot routine in ROM .
16618	167	EXIT: AND A	Reset carry.
	33, 255, 255	LD HL,255,255	
	237,75,37,64	LD BC,(16421)	Keyboard scan.
	237, 66	SBC HL,BC	Check what key
	124	LD A,H	pressed.
	133	ADD L	
	254, 3	CP 8	Check if "W" is pressed,
	200	RET Z	If so exit program.
	24, 134	FAR: JR,START	Go to beginning.
	17, 0, 80	PLOT/UNPLOT: LD DE 5000	Determines delay time
	27	DELAY: DEC DE	for button status.
	122	LD A, D	Delay loop
	179	OR E	
	32,251	JR NZ,DELAY	
	58, 33, 64	LD A,(16417)	Plot/unplot status.
	254,0	CP O	If unplot,
	40,7	JR Z,SETPLOT	Goto setplot.
	62,0	SET UNPLOT: LD A,O	Change plot to unplot.
	50,33,64	LD(6417), A	Store p/u status.
	24,232	JR, FAR	Go to start
	62,75	SETPLOT: LD A,75	Change unplot to plot.
	50, 33, 64	LD (16417), A	Store p/u status
16662	24,225	JR, FAR	Go to start
16664	End of Program.		

make use of the joystick interface. A short program in Listing 2 may be used to enter the code from the programs. To use the code-entering program, enter a starting address and then begin entering the code: for example type 58 and hit enter, type 0 and hit enter, and so on. When you wish to terminate the program, enter any letter and the program will stop with an error code. Remember to enter the line 2 REM statement with 50 Xs before entering the machine code.

Programs in Listings 3 and 4 are for use with the games mentioned at the beginning of this article. Mazogs is distributed by Bug-Byte, Flight Simulator by Timex-Sinclair. It can only be assumed that each cassette sold is identical to the ones I used. If you have either or both of these games, load one into the computer. After loading is complete, hit the break key to stop the program. Enter LIST 2 and note what is in lines 2 through 11 of the program. If any lines contain something other than REM statements, write it down so it can be re-entered later. Now enter the line 2 REM statement with 50 Xs and the program in Listing 2. After entering the code from Listing 3 or 4 to the addresses indicated, delete the code-entering program; do not delete the line 2 REM statement, though. Re-enter any lines that you may have deleted earlier from the original game program. The procedure outlined above is used for both games. The keyboard of the computer should function normally for both.

Listing 5 contains a program for drawing lines horizontally and vertically using the joystick to control the direction of the plotted or unplotted line. This program makes use of the plot routine in the ROM and can be run in the slow mode. The program is not tied to any address, so it can be entered and used at any location in RAM that you desire. Possibly the best place would be a REM statement (with 150 Xs) entered as the first statement in program memory. Code entered into a REM statement has the advantage of being saved easily for later use.

After the code has been entered for the line-draw routine, run the program by entering a program line in BASIC: 10 LET L=USR (your address). To stop the program hit the "W" key on the keyboard. To plot a line press the "fire" button on the joystick once; to unplot press the button again. The first delay in the program may be changed to a larger or smaller value. For instance, LD DE,900 could be changed to LD DE,200 for faster plotting or a larger value entered for slower plotting.

Hopefully, this article has aroused your imagination. If you have any ideas for the joystick interface, I would like to hear about them: write to me care of this magazine. (Please include a self-addressed, stamped envelope if a reply is expected.) The Timex Sinclair computer can be used for solving serious problems, but why not have a little fun, too? We deserve it!

Ø/Ø

Packrabbit With A Joystick

You might like to use your joystick with PACKRABBIT, the game that *Timex Sinclair User* sends to new subscribers. This method allows the joystick to be hooked up to the arrow keys (5, 6, 7 & 8). Just put the following statements into the program as direct commands (no line numbers).

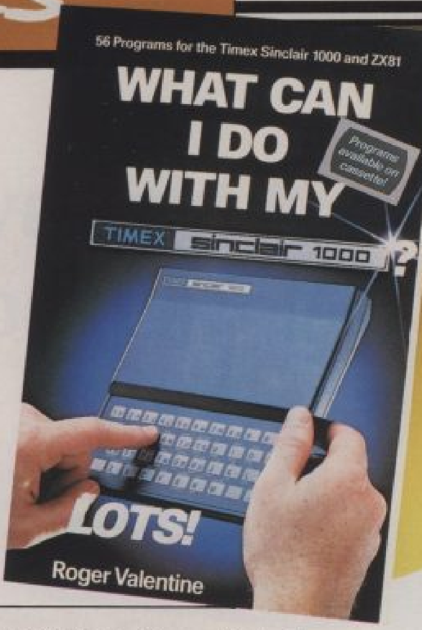
```
POKE 16622,239
POKE 16623,247
POKE 16612,247
POKE 16613,223
POKE 16632,239
POKE 16633,239
POKE 16642,239
POKE 16643,223
```

Now the joystick movement corresponds directly with the movement of the rabbit on the screen.

— Thomas Costa

Books

Something for Everyone



What Can I Do with my TS1000? LOTS!*, by Roger Valentine (John Wiley & Sons, $9.95)*

I MUST say right off the bat that I find prices of books related to Timex Sinclair equipment extravagant (usually multiples of $10), and the contents of these wallet-flatteners disappointing.

Now that I've got that off my chest, let me tell you about this one. Actually this one is *two*, originally published in England as *What Can I Do With the 1K?* and *What Can I Do With the 16K?* The Wiley Press version has a list price of only $9.95!

The next pleasant surprise comes with the contents, specifically layout and approach. The book contains 56 programs. They are clearly presented and if you've ever run into trouble with the "so simple a child could do it" scam, this will literally knock you over. Now even I (generally considered subnormal in computing and mental exercises) can tackle machine code! That's right! The instructions and notes which accompany *every* program listing have introduced me to the joys of machine code instructions. I'd heard about it before, and even tried to read about it, but all I'd succeeded in doing was putting the HEX on my meager brain cells.

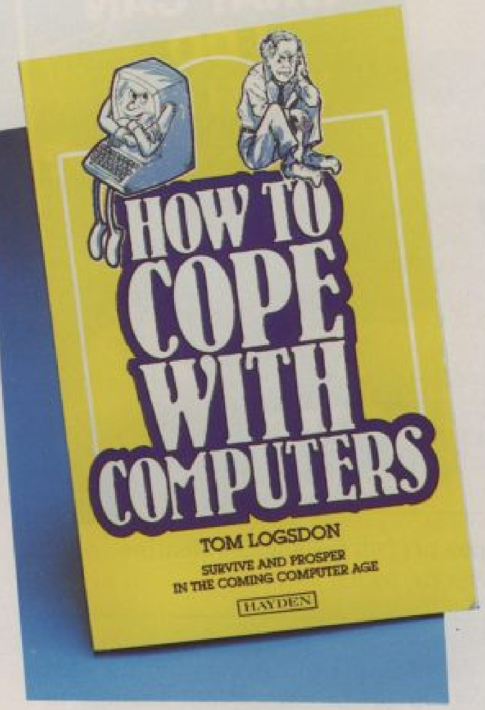
This book should be standard equipment to every Timex Sinclair operator. It refers specifically to the manual and explains special uses and functions not usually found in it. These little programming aids crop up in subsequent listings to ensure that you learn as you play.

The games you do play will interest all members of the family. The first half of the book (the 1K section) contains 4 listings. Their elegance will astound you. Then comes another bonus — they are not all games. They are divided into sections called Fate and Fortune, Printing with More Frills, Casino Gambling, Data Files, Business Programs, Utilities, and Games.

But it's Part 2 of the book that really impresses me. Stop reading here if you don't have 16K. The first section alone (Graphics Programs) makes the price seem reasonable. Then add five truly outstanding games (Pontoon, Battleships, Fortress, Wordsearch, Kami-Kazi Drive) that actually work and hold your interest. The notes and discussion explain these listings so that you can't help but be a better programmer after entering them. The final section of the book includes functional programs. I'm not sure they are, but the author's delightful style compelled me to read them.

This book is a must for all Timex Sinclair owners who would like to get the most fun and knowledge possible for the minimum investment.

— George Miller



Learning to Love the Beast

How to Cope with Computers *by Tom Logsdon (Hayden Book Company, $7.95)*

TOM LOGSDON has written an excellent book for all people who come in contact with computers — in other words, for all of us.

The book is divided into seven sections. The first, The Menace of Computers, provides some startling statistics regarding the influence of computers in our daily lives, and their nerve-wracking possibilities in the future. The next, Fighting Back Against Computers, has great appeal for those of us who have ever felt the urge to "bend, staple, spindle or mutilate." Again, we are urged to fight back intelligently and Logsdon clearly has a great sense of humor.

The third section, Learning Not To Worry About Computers, can be summed up in one of the fine illustrations titled "The Five Most Popular Coping Mechanisms." They are: 1) selective forgetting; 2) staking a claim to a different set of feelings; 3) attributing your own hateful thoughts to others; 4) coming up with a pseudo-rational justification; 5) picking a new target for your negative emotions.

These provide the core of Logsdon's advice to the computer novice and a satisfactory method of eliminating fear of what will inevitably be the machine on the right side of all humankind.

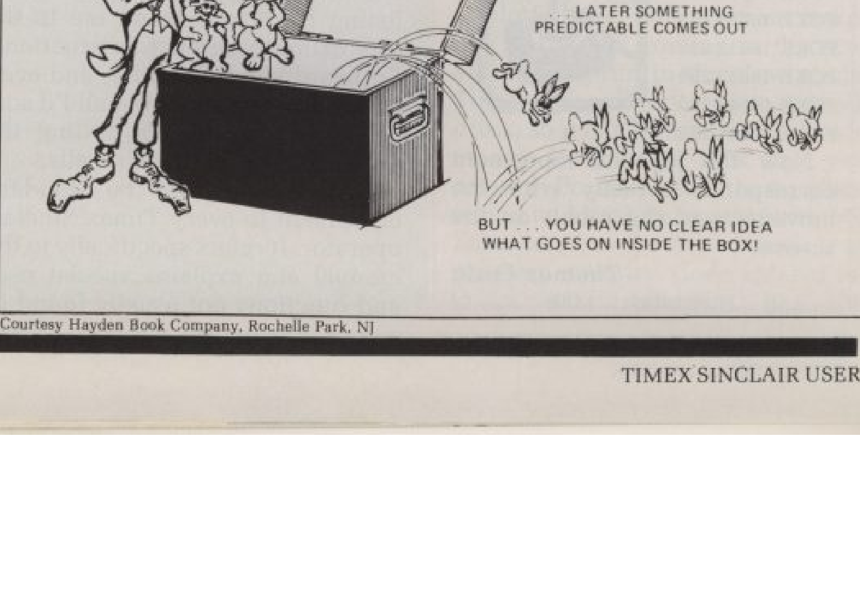
Learning The Innermost Secrets of Computers discusses the meanings of the jargonistic terms for hardware and makes reference to the appendix of currently vogue buzz-words. This is followed by a truly introductory lesson in BASIC.

The section on Landing A Job In The Data Processing Industry is weaker than the others, in my opinion. As a "how to" guide it is satisfactory, but the necessary skills and aptitudes are ignored in favor of educational possibilities. Clearly, not everyone is suited for employment in the computer industry.

The final section, Buying And Operating Your Own Computer, provides a guide to the features and applications of several specific brands of computers. Notably missing from this list are Sinclair products.

The book is clearly written and for those who have had very little exposure to computers will make an excellent choice. If your Aunt Maud can't understand your addiction to computers, this might make a great birthday gift. A word of warning, though, if Aunt Maud is picky, she'll complain to you about the *one* misprint in the book. That's right, there is only one and it changes "A not equal to B" into "A = B". Then again, if Aunt Maud is that picky, just recommend the book to her. It will make a useful beginning volume to anyone's computer library. — ***George Miller***

0/0



Courtesy Hayden Book Company, Rochelle Park, NJ

Limited mail-order availability for new ZX Microdrive

SINCLAIR RESEARCH has introduced a new form of selling to the U.K. market. Following its success with mail order, it has now introduced the concept of limited mail order. Instead of offering goods for sale to anyone, it is making sure it can control demand by sending order forms to a select group of possible customers.

The novel idea has been introduced into the selling of its latest technological breakthrough, the ZX Microdrive and associated interfaces. It is the result of the company's experiences in launching previous major products, the ZX80, ZX81 and Spectrum. All three suffered from delivery delays mainly because the demand was far larger than anticipated. The first few months following the launch of the Spectrum were dominated by news of missed delivery dates with some people having to wait up to four months for their machines which were ordered on the promise of 28 days delivery. The Spectrum was launched in April, but the situation did not return to normal until Christmas.

It was expected that when the Microdrives finally appeared that Sinclair Research would attempt to ensure that problems did not occur again by having sufficient stocks to take care of any explosion in demand. This of course is not an ideal solution when a revolutionary product is being launched. Past experience of other new products can provide little more than a rough guide as to what demand can be expected.

If this proves to be an underestimate, then delivery delays occur again. If it is an overestimate, Sinclair is left holding stocks for which it has no immediate customers.

> To avoid delays in delivery, Sinclair is controlling demand by sending order forms to select groups of customers

The company, however, has two advantages over any other company in the same position. Because of its mail-order sales, it is able to identify many potential customers. It also has a captive audience as no one else has brought out a mass-storage device for the Spectrum at a similar price. It has been able, therefore, to send order forms to the first 1,000 people who bought the Spectrum last year in the knowledge that it will not lose any potential customers to a rival producer by asking them to wait a little longer.

By limiting the demand in this way, Sinclair Research is hoping to avoid the confusion and annoyance at ordering something on the promise of 28 days delivery and then not knowing when it is going to arrive. It is also intended to compensate for the delays of last year. Those who had to wait the longest were among the first to order the new machine.

In addition, the company hopes to be able to gauge the response of the Sinclair market to the new device. It is doubtful if it will be successful in giving an early indication of that unless production can be increased quickly. With large numbers of people having waited a long time already it is thought that most people with an order form will buy one whether they want one or not in the hope of being able to resell it at a premium. An alternative would be to sell the order form to the highest bidder.

By the time the full list of mail-order customers has been exhausted, however, a better idea of the proportion of Spectrum owners wanting the Microdrive will have been obtained.

How long it will be before Timex takes advantage of its agreement with Sinclair Research to make use of all its development work is still not known. — ***Nigel Clark***

in London

Ø/Ø

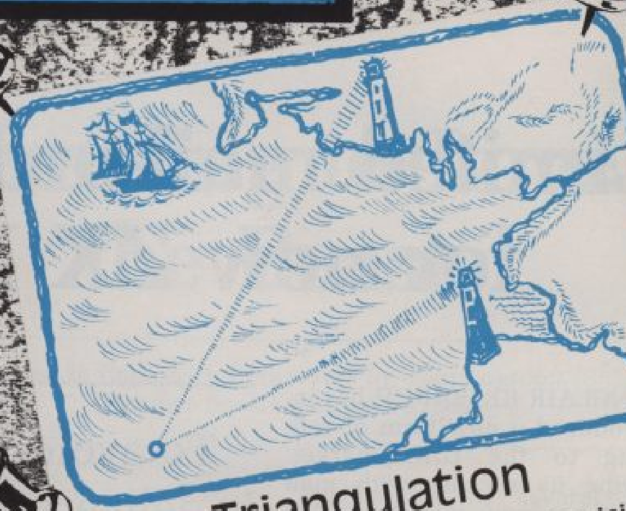
Bulletins

A Celebration

The Sinclair-Timex User Group of the Boston Computer Society is celebrating its second anniversary by holding a Timex-Sinclair Celebration. It will be on Saturday, October 22 from 10 a.m. to 6 p.m. at the Boston Park Plaza Hotel.

Third party software companies will be demonstrating and selling their products (rumor has it that bargains will abound). In addition, workshops and hands-on activities will let users discover ways in which the T/S1000,1500 can be used in the home, office, school and for entertainment. Gregory Coffin, Ph.D., Director of Urban Schools Collaborations, Northeastern University will explain how T/S computers are being used in Boston schools.

It appears that this celebration will truly have something for everyone. Details can be obtained from the Boston Computer Society, 3 Center Plaza, Boston, MA 02108. Phone: (671) 367-8080. Happy birthday Boston group!



Triangulation

G. Jude Miller of Caremcro, Louisiana, writes: "I'm having a hard time finding a program for my T/S1000 that will do triangulation. We do triangulation using maps on our Search and Rescue missions. If you know of anyone who has a program of this sort, please forward the address or program, if possible."

Anyone with such a program or wishing more information should write to Miller at P.O. Box 105, Caremcro LA 70520.

Upcoming Events

Oct. 14-15: Computers and Reading/Learning Difficulties, Dallas, TX

Oct. 14-15: 5th Annual Forth Convention, Palo Alto, CA

Oct. 15: Microcomputer Show & Flea Market, East Rutherford, NJ

Oct. 19-21: 4th Annual Canadian Symposium on Instructional Technology, Winnipeg, Manitoba, Canada

Oct. 19-21: National Software Show, San Francisco, CA

Oct. 19-21: SIBEC Info Expo, Montreal, Quebec, Canada

Oct. 20-23: Computer Showcase Expo, Pittsburgh, PA

Oct. 24-27: Comdex/Europe, Amsterdam, The Netherlands

Oct. 27-30: Computer Show & Office Equipment Expo, Washington, D.C.

Oct. 31 - Nov. 3: IEEE International Conference on Computer Design, New York City, N.Y.

Nov. 3-6: Computer Showcase Expo, Denver, CO

Nov. 10-13: Computer Showcase Expo, Los Angeles, CA

Nov. 10-13: Computer Showcase Expo, Washington, D.C.

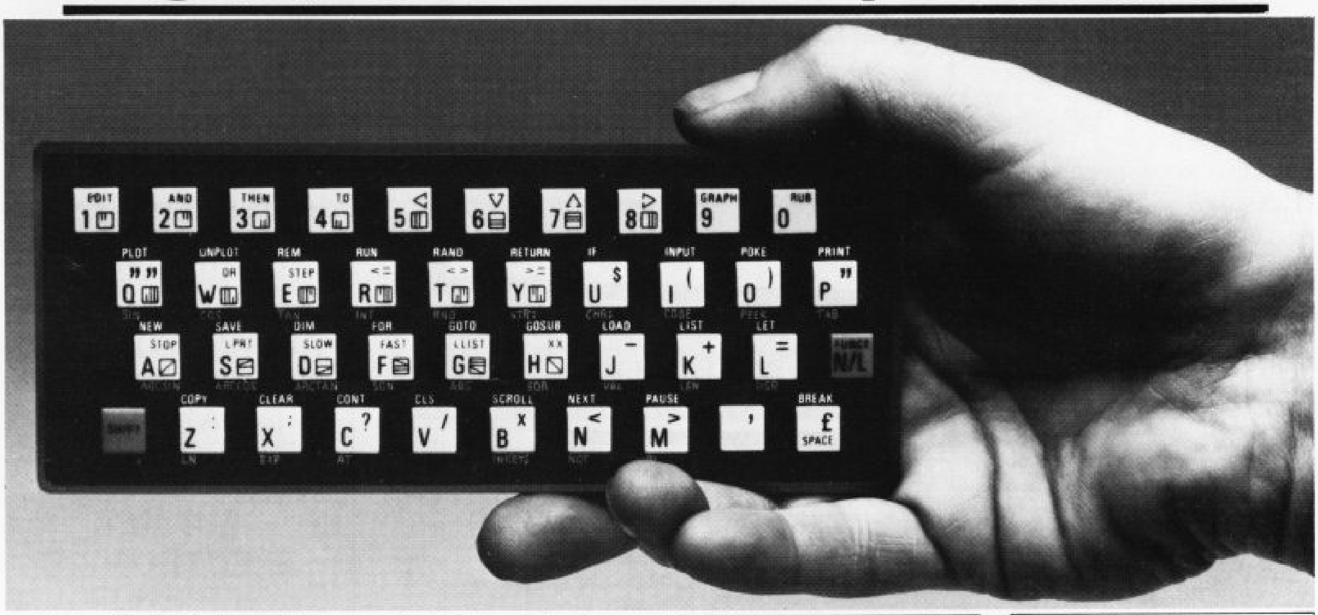
Telecommunications Device for the Deaf

A TDD (Telecommunications Device for the Deaf) is a device that allows a deaf person to make telephone calls directly, without the need of another person to interpret. The conversation is typed rather than spoken, both parties in the conversation must use TDDs, and their devices must be compatible.

J. Kenneth Lewis of Punta Gorda, Florida, who works with the deaf community in southwest Florida, is eager to find out whether the T/S1000 can be adapted to function as an inexpensive TDD. "It appears that to use the T/S1000 as a TDD, it must be provided with a telephone-compatible modem and an ability to transmit and receive using the Baudot code (compatible with teletype machines)," he writes.

Lewis has examined the User Manual and other T/S1000 reference books and found nothing regarding TDDs. He believes the lives of many of the 29 million deaf Americans would be enhanced by an inexpensive TDD that would also help them learn computing. If you can help, circle 37 on the reader service card and we will forward your name.

NOW. A TS1000/ZX81 PUSH-BUTTON KEYBOARD FOR UNDER $20.00.

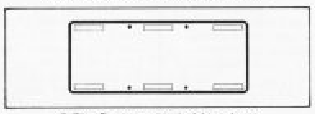


At last there's a really cheap but efficient way of ironing out the TS1000/ZX81's only real bug: its keyboard. The Filesixty Buttonset offers

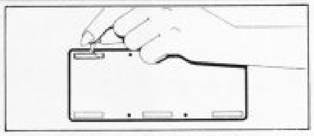
- ■ A full–travel calculator–type moving keyboard for only $19.50.
- ■ Installed in seconds. The peel off adhesive backing means you just register into position and press.
- ■ No messy labels, dismantling or soldering.
- ■ 3 groups of colour keys to pick out shift, numerals and newline.
- ■ Precision moulded in ABS to match your TS1000/ZX81, with contrasting legends for maximum legibility.

Orders to Filesixty Ltd.,
37 Fruehauf St., Buffalo, N.Y. 14226

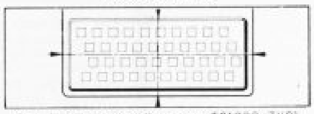
1. Make sure the original keyboard is clean and check that all the keys function.



2. The Buttonset is held in place by self adhesive pads.



3. So all you do is remove the protective backing.



4. And place it centrally on your TS1000/ZX81

Cheques/money orders made payable to Filesixty Ltd.

Please send me__________(qty.) Buttonset(s) at $19.50 each (inclusive of postage & packaging).

Total $__________ BLOCK CAPITALS

Name__________

Address__________

____

State__________

Zip__________ FILESIXTY USA2TSU

For more information, circle 33 on reader service card.

Bulletins

Use this free info service

To obtain more information on any advertiser or product in *Timex Sinclair User*, circle the appropriate number on the reader service card opposite.

QUICKSILVA
IS HERE
AMERICA
QUICKSILVA
Awesome software which extends your computer's potential to the maximum.
Enjoyed by millions in Europe, now Quicksilva comes to America.
QUICKSILVA
Britain's leading manufacturer and distributor of video games and other software. Programs available for the Timex-Sinclair 1000 and 2000™ VIC-20™ and Commodore 64™, BBC™ and Dragon™ Atari 400 and 800™.
Distributor
inquiries invited for USA and Canada: Please contact us for further information
Dealer
inquiries invited for USA and Canada: Please contact us for further information
QS™
THE GAME LORDS OF BRITAIN
QUICKSILVA INC.
426 West Nakoma
San Antonio, TX 78216
Tel: (512) 492-8054
Please send me a free color catalog
I currently own or plan to buy a ....computer
I enclose a stamped, self-addressed envelope
Name...........
No/Street...........
City...........
State...........
Zip...........
S1
QS
For more information, circle 2 on reader service card.