people's computers
PCC
Box E
Menlo Park, CA 94025

Second class postage
paid at Menlo Park,
California 94025 and
additional entry points

people's computers

$1.50

VOL 6 NO 2 SEPT – OCT 1977

COMPUTER NETWORKS

THE $595 PET

MORE tiny LANGUAGES

SUBMITTING ITEMS FOR PUBLICATION

LABEL everything please, your name, address and the *date*; tapes should also include the program name, language and system.

TYPE text if at all possible, double-spaced, on 8½ x 11 inch white paper.

DRAWINGS should be as clear and neat as possible in black ink on white paper.

LISTINGS are hard to reproduce clearly, so please note:

- Use a new ribbon on plain white paper when making a listing; we prefer roll paper or fan-fold paper.
- Send copies of one or more RUNs of your program, to verify that it runs and to provide a sense of how things work -- and to motivate more of us to read the code. RUNs should illustrate the main purpose and operation of your program as clearly as possible. Bells, whistles and special features should just be described in the documentation unless they're particularly relevant.
- Paper tapes of both the program and runs can provide us with a way to make our own listing if we need to. Then, if you give us permission, we can let CCC (Community Computer Center) sell your program cheaply via paper tape, to further the spread of inexpensive software. Finally, if we are so lucky as to have access to a system on which your program runs, we can try it out ourselves.
- Make sure your code is well documented -- use a separate sheet of paper. Refer to portions of code by line number or label or address please, not by page number. When writing documentation, keep in mind that readers will include beginners and people who may be relatively inexperienced with the language you're using. Helpful documentation/annotation can make your code useful to more people. Documentation should discuss just which cases are covered and which aren't.
- If you send us a program to publish, we reserve the right to annotate it (don't worry, we won't publish it if we don't like it).
- Last but not least, please try to limit the width of your listings: 50-60 characters is ideal. Narrow widths mean less reduction, better readability, and better use of space.

LETTERS are always welcome; we assume its OK to publish them unless you ask us not to. Upon request we will withhold your name from a published letter, but we will not publish correspondence sent to us anonymously. We reserve the right to edit letters for purposes of clarity and brevity.

SUBSCRIPTIONS

U.S. Subscriptions
□ $8/yr. (6 issues)
□ $15/2 yrs. (12 issues)
□ Retaining subscription @ $25
($17 tax deductible)
□ Sustaining subscription @ $100+
($92+ tax deductible)

Foreign Surface Mail
□ add $4/yr. for Canada
□ add $5/yr. elsewhere

Foreign AIRMAIL
□ add $8/yr. for Canada
□ add $11/yr. for Europe
□ add $14/yr. elsewhere

Back issues, $1 each; indicate Volume and Issue number, how many copies of each. An order card is at the center of the magazine.

Vol. 1, No. 3
Vol. 3, No. 1
Vol. 4, Nos. 3, 4, 5, 6
Vol. 5, Nos. 2, 3, 4, 5, 6
Vol. 6, No. 1

Foreign Distributors of *People's Computers*

Vincent Coen
LP Enterprises
313 Kingston Road
Ilford, IG1 1PJ
Essex, UK

Comicro AG
Baderstrasse 281
CH-8003 Zurich
SWITZERLAND

Pan Atlantic Computer Sys.
Frankfurter Strasse 78
D61 Darmstadt,
WEST GERMANY

Home Computer Club
1070-57 Yamaguchi
Tokorozawa, Saitama, JAPAN

Kougakusha Publ. Col, Ltd
Haneda Biru 403, 5-1
2-Chome, Yoyogi
Shibuya-Ku, Tokyo 151
JAPAN

Computer Age Company, Ltd
Kasumigaseki Building
3-2-5 Kasumigaseki
Chiyoda-Ku, Tokyo 100
JAPAN

ASCII Publishing
305 HI TORIO
5-6-7 Minami Aoyama
Minato-Ku, Tokyo 107
JAPAN

People's Computers is published bimonthly by People's Computer Company, 1263 El Camino Real, Box E, Menlo Park, CA 94025.
People's Computer Company is a tax-exempt, independent, non-profit corporation, and donations are tax-deductible.
Second class postage paid at Menlo Park, California, and additional entry points.

people's computers

VOL 6 NO 2
SEPT-OCT 1977

STAFF

EDITOR
Phyllis M. Cole
PRODUCTION
Meredith Ittner
Donna Lee Wood
ARTISTS
Craig Douglas
Maria Kent
Kathy Smith
TYPISTS
Maria Kent
Renny Wiggins
CIRCULATION
Bill Bruneau
DRAGON EMERITUS
Bob Albrecht

RETAINING SUBSCRIBERS

George Bowie III
The Computer Corner, Harriet Shair
Daniel Dick
Bill Godbout Electronics
Mark S. Elgin
Dexter Fletcher
John B. Fried
Dick Heiser
Brian W. John
Lt. Stan Jurgielwicz
John R. Lees, Jr.
John C. Lilly
James Muysenberg
Frank Otsuka
Bernice Pantell
Larry Press
John Rible
Joe Weisbecker

SUSTAINING SUBSCRIBERS

Algorithmics Inc., Bruce Cichowlas
Don L. Behm
BYTE Publications, Carl Helmers,
Virginia Peschke, Manfred Peschke
Paul, Lori and Tom Calhoun

As ever, thanks to the many many folk who supported our effort in putting this issue together.

CONTENTS

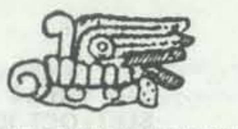
SPECIAL FEATURES

COMPUTER NETWORKS

ARTICLES

REGULAR STUFF

Fortran Man will reappear in our next issue after artist Ann Miya returns from a European vacation.



LETTERS

I find the world of microcomputers fascinating. As yet I haven't built anything or bought any hardware although I have been reading like mad and learning as much as I can. Would you please print this in your magazine in hopes that someone near to me (very near, I can't drive) with an operating computer will contact me? I would like to gain some experience on a computer and I can't do this without one. I am also interested in Amateur Radio so if there is anyone who has interfaced the two I would like to hear from them. And last but not least I am looking for back issues of BYTE, Popular Electronics, Radio Electronics, and Scientific American.

Victor O'Rear
7956 Lava Court
La Mesa, CA 92041

After reading *People's Computers* (Vol 5, No 6), I have just a few comments.

The new format of your newspaper was long due and now that it's here I feel it was well worth the wait – Great Job!

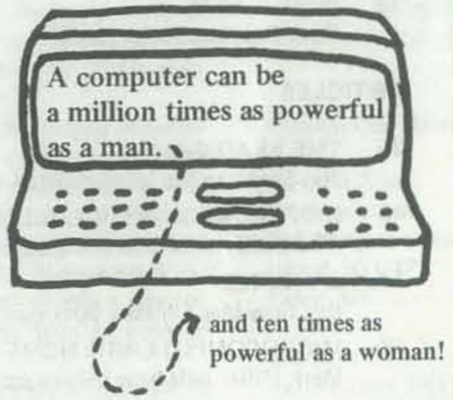
Is it possible to get some articles covering products out on the market or currently being developed so we could have an idea of what is worth spending our hard earned dollars on?

Finally, please Phyllis, I think the majority of the readers feel they can get along without having articles in the newspaper which cater to women.

Let's just stick to good old personal computing – by men *and* women. Thank you!

David Ritter
1006 S 8th
LaCrosse, WI 54601

We'll try to keep you up to date on new products – see the article on Heathkit computers in the last issue and the interview on the PET in this issue. So far the response to the articles on 'Women & Computers' has been favorable. Most people who wrote or spoke to me about the articles felt that it was time the issues were raised, problems acknowledged, and solutions sought. But we've no intention of belaboring the point.



For shame! And in your 'Women & Computers' issue too! For generic terms, 'human' isn't bad.

Judy Edwards, Faithful Fan
Northwest Regional Educational Lab.
Lindsay Building
710 S.W. Second Ave.
Portland, OR 97204

What can I say? Editing to eliminate inappropriate sex-biased phrases is not always easy: in such instances editorial changes may significantly modify the author's message. While I favor presenting information without sex bias, I also feel it inappropriate to drastically change an author's presentation. Often such cases can be discussed with an author, but I avoid making substantial editorial changes in reprinted material. However I'm reconsidering my options.

I wrote a version of the Reverse program described in Vol 5 No 5, in the Smalltalk language. I find it difficult to describe the program without having to describe the entire language first. I refer those interested in learning more about this wonderful language to *Creative Computing* Sept-Oct '75 and May-June '77, and the March '77 issue of IEEE's *Computer* magazine.

Bob Martinengo
487 James #202
Palo Alto, CA 94306

A letter to The Dragon:

As an enthusiastic calculator user and confirmed Reverse-Polish-Notation programmer, not to mention a Hewlett-Packard Applications Engineer, I would like to compliment you on your series of articles about SAM. I come into contact with many beginners who could benefit from such a treatment of calculator operation. There are two minor points that you might wish to consider for your future SAM articles.

First, an easy way to reduce your step count is to avoid clearing the stack before each new calculation. A little thought will convince you that a cleared stack is unnecessary, for the previous data and answers in the stack are simply pushed off the top by data from the new problem. Those familiar with HP calculators will note that our newer calculators no longer have the CLEAR STACK function.

My second comment is regarding the evaluation of polynomials. Now that you have taught how to figure out individual examples, perhaps you could teach a general method. For example the polynomial

$$a_0 + a_1 x + \ldots + a_{n-1} x^{n-1} + a_n x^n$$

may be rewritten as

$$a_0 + x(a_1 + \ldots + x(a_{n-1} + xa_n)\underbrace{\ldots)}_{\text{n-1 right parentheses}}$$

Starting from the inside, we can write this code as

```
INP x
UP
UP
INP $a_n$
x
INP $a_{n-1}$ }  Repeat n-1 times as i
+             }    goes from 1 to n-1
x             }
INP $a_0$
+
```

Thus there are 3(n-1) + 7 steps required to evaluate an n-th order polynomial by this method. Should there be coefficients of zero or one, fewer steps are needed. Consider the polynomial:

$$1 + 2x^2 + x^4$$

It may be written as

$$1 + 0x + 2x^2 + 0x^3 + x^4$$

which by our method is evaluated in 16 steps as follows

```
INP x
UP
UP
INP 1
x
INP 0
+
x
INP 2
+
x
INP 0
+
x
INP 1
+
```

The sequences (INP 1, x) and (INP 0, +) may be deleted, however, leaving only ten steps

```
INP x
UP
UP
x
INP 2
+
x
x
INP 1
+
```

I hope these ideas are useful to you.

Kenneth E. Newcomer
Applications Engineer
Hewlett Packard, Corvallis Division
1000 N.E. Circle Blvd
Corvallis, OR 97330

Let me introduce myself: my name's Paul and I'll be starting my senior year in high school in September. Because of that, I have to start thinking about what college I want to attend next fall. Since most colleges require that you apply for admission by late Fall or early Winter, I have to start thinking about *which* colleges to apply to. Why am I writing to you? I want to major in computer science, and I'm interested in finding out what the best schools for C.S. are in California. Since many of you are involved in education, I hope that you can give me some recommendations.

Many schools have programs in C.S., but some are unacceptable to me because of two things:

1) I'm interested in systems software (compilers, operating systems and the like) and many of the courses in this area are graduate courses. So I'd like the school to offer a M.S. in C.S.
2) I've been raised on interactive systems, so I'd hate to go to a school that does all its work on some IBM monster – I *hate* punch cards.

One place I've been considering that satisfies both constraints is U.C. Irvine. It does have a graduate school, and its computing facilities include a PDP-11 and a DECsystem-10. Have you heard anything about Irvine? Is it any good?

I'd be interested in finding out about any good C.S. schools in the West (although I'm not sure if there are even any in the West outside of California).

I'd really appreciate any help you can give me in my quest. If you don't have any info, perhaps you could direct me to someone who does?

All take and no give is no good, so I'll offer some comments about PC.

I didn't like the May-June issue. (I didn't say the comments would be good!) I enjoyed the article on 'Home Computing for Novices' (even though I'm not), as well as the Faire stuff, 'Fortran Man,' and the 'Don Quixote Starship' article. The majority of the other stuff just didn't appeal to me, but then you can't please everyone! If you stick to your intended aims, I'll probably love PC. Two things I did not like about your format change are the reduction of letters, and the elimination of most of those fantastic graphics. I've always enjoyed the letters part of *PCC*; it's neat to see other people are thinking and doing. There were still a few of those strange drawings that used to fill your pages (maybe you don't have room for them anymore?), but they were few and far between (cliche time, folks!).

I guess the main thing I don't like is that you are becoming more and more like the other magazines: straight laced and less fun-loving and irreverent. (I like Monty Python, which may help to explain my attitude.) (I also like parentheses) (What?) In my earlier *PCC*'s, I enjoyed all the descriptions of non-computer games, like Dungeons and Dragons. As a matter of fact, that article caused me to go out and buy the game. It's fantastic and amazing, but I've not been able to find anyone out here in the Livermore valley who plays.

I considered not renewing my subscription, but I guess I'll give you a chance. (How generous of me, eh?) I realize that this has not been a very complimentary letter, but I hope that you'll still help me. In any case, good luck and best wishes (really!). May the force be with you,

Paul Holbrook
6104 Crater Lake Ct
Pleasanton, CA 94566

For colleges and universities that offer at least an MS. in computer science, look into Stanford, the University of California at Berkeley, Santa Cruz, and San Diego as well as Irvine. Stanford offers free interactive computing to its students under its LOTs program; I think the other places have interactive systems as well. Take a look at the offerings of the University of Oregon in Eugene and Simon Fraser University in Vancouver. Irvine has a fine reputation in the field of computer science. If you're considering the East coast, investigate the University of Rochester and especially Carnegie Mellon University in Pittsburg.

We'll be reserving more space for letters, so keep writing. We hear your comments and will respond to them as best we can, but People's Computers *readers are such a varied crew that as you noted we'll have trouble pleasing everyone at once.*

Seeing as you peoples are interested in bio-feedback, computers, and music, I thought I'd put into code an idea I've been fostering concerning a ***Serendipitous Melging*** of the three. It's a program that's supposed to convert brainwaves straight into sound!

You take a brainwave pickup, amplify the signal to a nice useable voltage, do a little filtering to take out the AC hum, and feed it into an analog-to-digital converter. My program takes it from there, plays around with it, and finally sends the processed signal out to a digital-to-analog converter. You take that signal, filter it some to make it smoother and nicer, and put it thru a hi-fi- or guitar amp and into a speaker. The object of the program is to raise all of the frequency components of the brainwave signal up into the audible range while preserving the ratios between the various frequencies present in the brainwave. These frequency ratios would become *musical intervals*. For instance, if your brainwave contained jitterings at 5.5, 6.93 and 8.24 Hertz (cycles per second), MUSE would send out tones at 440, 554.4 and 659.2 Hertz, and you would hear a three-note, A-major chord!

I figure there must be someone who reads *People's Computers* and has the equipment to use this program. *I don't*, and I want to find out if it works!

Here's the stuff you need to run the unmodified version of MUSE: a brainwave pickup, a DC amplifier capable of increasing the brainwaves to a level which is close to the full range of your analog-to-digital converter, an active low-pass filter with a corner frequency of 20Hz., an 8-bit analog-to-digital converter (connected to an input port on the computer), an 8080-based computer with a 2MHz clock and 3/4 K of *full speed static* RAM, an 8-bit digital-to-analog converter (hooked to an output port), a speaker amplifier, and a speaker. The A-to-D must be of the type which does its conversion by itself (like the Cromemco) and not the software-driven type (like the Polymorphics). Preferably, it should do the conversion while the computer is busy, and have the data ready right when the computer asks for it. In any case, you should know the number of wait-states it causes (if any). There are some optionals: a high pass filter (to limit thumping) and a 1600Hz. low-pass filter (to cut out high-frequency distortion) on the output, and a low-frequency-limit high-pass filter for the input.

```
1000            0010 ; MUSE    Brainwave-to-sound conversion program.
1000            0020 ;                  by Steve Witham   6/26/77
1000            0030 ;
1000            0040 OPI EQU 80 ; Number of outputs per input.
1000            0050 OCIC EQU 80 ; Number of output cycles per input cycle.
1000            0060 TABL EQU 0F800H ; Waveform storage table.  Set at beginning
1000            0070 ;                 of Poly screen RAM. -- THIS MUST BE A
1000            0080 ;                 MULTIPLE OF DIVD!!!
1000            0090 DIVD EQU 8 ; Number of inputs per output cycle.  THIS MUST
1000            0100 ;            BE A POWER OF TWO!!!
1000            0110 DAC EQU 0 ; Digital-to-analog output port.
1000            0120 ADC EQU 1 ; Analog-to-digital input port.
1000            0130 ;
1000            0140 ; The following constants specify lengths of time to pause
1000            0150 ; between outputs.  They are different to compensate for
1000            0160 ; time taken up by different parts of the program.
1000            0170 TIM0 EQU 17 ; (312-54)/15
1000            0180 TIM1 EQU 14 ; (312-98)/15
1000            0190 TIM2 EQU 13 ; (312-123)/15
1000            0200 TIM3 EQU 12 ; (312-130)/15
1000            0210 ;
1000            0220 ;                   CYCLES
1000 21 00 F8 0230 MUSE LXI H, TABL ;  ---   Set up output pointer.
1003 0E 50      0240   MVI C,OPI ;      ---   Set outputs-per-input counter.
1005            0250 RSTI EQU $ ;       106 cycles so far.
1005 11 00 F8 0260   LXI D,TABL ;      +10   Set/reset input pointer and
1008 06 50      0270   MVI B,OCIC ;     + 7   output-cycle-per-input-cycle
100A            0280 ;                       counter.
100A 3E 0C      0290   MVI A,TIM3 ;     + 7  Pause after output+input+
100C            0300 ;                        output reset + input reset.
100C            0310 ;                 =130 cycles for o + i + o r + i r.
100C 3D         0320 LOOP DCR A ;         5   Count down for delay.
100D C2 0C 10 0330   JNZ LOOP ;         +10 = 15 cycles per delay loop.
1010            0340 ;
1010 7E         0350   MOV A,M ;          7   Get value from table.
1011 D3 00      0360   OUT DAC ;        +10
1013 23         0370   INX H ;          + 5   Move output pointer foreward.
1014 0D         0380   DCR C ;          + 5   One less output to do before
1015            0390 ;                        inputting.
1015 CA 1D 10 0400   JZ INPT ;          +10   Oh -- no more?  Then input now!
1018            0410 ;                  =37 cycles up to and including this 'JZ'
1018 3E 11      0420   MVI A,TIM0 ;     + 7   Pause for after output only.
101A C3 0C 10 0430   JMP LOOP ;         +10 = 54 cycles to do output only.
101D            0440 ;
101D            0450 ;                   37 cycles so far.
101D 0E 50      0460 INPT MVI C,OPI ;   + 7   Reset # of outputs per input.
101F DB 01      0470   IN ADC ;        +10
1021 13         0480   INX D ;         + 5   Advanced input pointer.
1022 7B         0490   MOV A,E ;       + 5  The input pointer contains important
1023 E6 07      0500   ANI DIVD-1 ;    + 7   information in its bottom bits.
1025 CA 2D 10 0510   JZ RSTO ;         +10   Have DIVD inputs (i.e., DIVD*OPI
1028            0520 ;                       outputs) been done since last
1028            0530 ;                       resetting output pointer?
1028            0540 ;                 =81 cycles up to & including the 'JZ'.
1028 3E 0E      0550   MVI A,TIM1 ;    + 7   Pause after output+input.
102A C3 0C 10 0560   JMP LOOP ;        +10 = 98 cycles to do output + input.
102D            0570 ;
102D            0580 ;                  81 cycles so far.
102D 21 00 F8 0590 RSTO LXI H,TABL ; +10   Reset output pointer.
1030 05         0600   DCR B ;         + 5   If OCIC output pointer resets
1031            0610 ;                       (output cycles) have been done then
1031            0620 ;                       OCIC*DIVD inputs have been done,
1031 CA 05 10 0630   JZ RSTI ;         +10   and one input cycle is complete.
1034            0640 ;                =106 cycles up to & including this 'JZ'.
1034 3E 0D      0650   MVI A,TIM2 ;   + 7   Otherwise pause for output+input+
1036 C3 0C 10 0660   JMP LOOP ;       +10   resetting output.
1039            0670 ;                =123 cycles for out + in + out reset.
```

MUSE as it stands has one problem: rotten transient response. If the input brainwave pattern *changes*, it will take eight seconds for the new pattern to completely take over the sound output. During this period, the old pattern/sound fades out while the new one fades in. Sorta like playing a piano with the pedal down. This can be changed. However, any increase in transient response (decrease in fade-in/out time) must be accompanied by a proportional loss of low frequency response. The table shows how to modify the program and circuits to trade off low-frequency response vs. transient response.

Low-frequency cutoff (Input high-pass freq.)	Transient response	Output high-pass freq.	DIVD	TIME
2.0 Hz.	8.0 sec.	160 Hz.	8	312
2.8 Hz.	5.6 sec.	224 Hz.	8	223
4.0 Hz.	4.0 sec.	320 Hz.	4	312
5.6 Hz.	2.8 sec.	448 Hz.	4	223
8.0 Hz.	2.0 sec.	640 Hz.	2	312
11.2 Hz.*	1.4 sec.	896 Hz.	2	223

*With the low frequency response at 11.2 Hz., you might want to increase the frequencies of the input and output low-pass filters to 25 and 2000Hz., respectively.

To use the table, pick the LF response /transient response pair you want to use. Set the filters as shown. If you have an assembler, you can stick DIVD straight into the EQU statement in line 0090. Otherwise, subtract one from the number shown in the table and put it in the second byte of the ANI instruction on line 0500 (location 1024 Hex.). Now you have to deal with that cryptic "TIME" column in the table. TIME is supposed to be the number of machine cycles between output strobes. It finds its way into the program via the constants TIM0 thru 3. (Notice the conspicuous 312 in those EQU statements?) Here's the complete set of formulas for figuring them out:

```
TIM0=INT((TIME-DLAY-54)/15+.5)
TIM1=INT((TIME-DLAY-98)/15+.5)
TIM2=INT((TIME-DLAY-123)/15+.5)
TIM3=INT((TIME-DLAY-130)/15+.5)
```

Here, DLAY is the number of wait states for the ADC. If your assembler has a divide operation for figuring out constants, you can make the computer do the work, but you'd probably have to type the expressions like this:

```
TIM0 EQU TIME-DLAY-54+7/15    Etc. . .
```

The '+7' is for rounding.

* IF you get this working . . .
* IF you have a software ADC . . .
* IF you have problems making filters . .
* IF you know how to do FFT's, FF un-T's, and/or digital filtering . . .
* IF you have slow RAMS . . .
* IF you have *any* comments, suggestions, questions, etc. . . .
* THEN write me!

Steve Witham
168 Painter Rd
Media, PA 19063

Is it okay if I try to monopolize your magazine? MUSE, by the way, can also be used as a voice-frequency-raiser or lowerer, but don't tell anybody! I want to be the first to try it! (If you hook it to your brainwaves, does that make it a consciousness raiser?)

I know you didn't want to continue the San Andreas biz, but couldn't you include this one as a fitting (and robotic!) finish?

I've also included some more comix by my friend Bob. As you can see, they're closer to his line – space stuff. By the way, did you like the previous batch? (Software Sam, Et Al?) I'm not saying you *should* have, but . . .

Here's a good slogan (suitable for buttons and T-shirts) for non-conformist computer hackers (and aren't we all?): 'Boot the System!'

Are you going to print that harmony program from the Kemeny and Kurtz book? I saw it in a bookstore once, and it looks like the algorithm is perfect for hooking to TONEGEN. If you're not printing it, can you Xerox it for me? (Is that nice?) Are you going to print other composer-type programs?

Steve Witham
168 Painter Rd
Media, PA 19063

Sure, monopolize away!
Your San Andreas finale will probably appear in the next issue. As to the drawings sent in your last two letters, several comments: First, use very black ink on smooth white paper; clear, well-delineated lines reproduce best. Second, plan drawings to fit in one of our standard sized areas for illustration; there's a tendency to put lots of detail into large pictures which gets lost if we reduce the size for publication. Third, we especially like stuff that somehow fits the subject being illustrated (although random illustrations are sometimes a nice change of pace); cartoons are welcome if they're funny – but alas, funny ones are hard to find!

For your spacey friend, how 'bout illustions for Don Quixote Starship? The next installment will appear in our November /December issue. Have Bob take a look at Volume 5, Numbers 4-6 to get a sense of what DQS is about.

I'm especially interested in science fiction and space simulations. I think that 'Star Trader' runs rings around most 'Star-Trek' programs. Maybe if someone would write a program where the Klingons (or Romulans) were more fairly represented. By this I mean: the ability to fire photon torpedoes; better shields; more mobility; the use of a cloaking device (at the cost of moving or something); the ability to fire on Star Bases. Then maybe I might change. Only after the 'Star Trader' had been written to its fullest potential and I got sick of it (fat chance). Perhaps if 'Star-Trek' could be played with another player as the bad guy (you don't know how fun it is to fire on a Star Base until you try it).

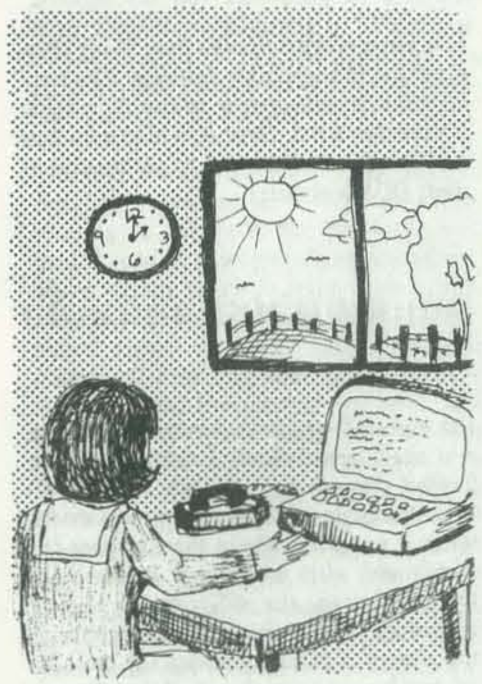
I had an idea about a game that may give some of your helpers an idea. It is about creating a monster with a killer nose. Some tactics I thought of to thwart him include: mothball bombs; blow up a pepper factory; parachute a group of plastic surgeons. I was also wondering if you knew about any programs like 'Hurkle' that allow the hurkle to move around when you get close?

Phil Dolan
7415 Portland Ave. S.
Richfield, Minnesota 55423

The Computer Conference

An Altered State of Communication?

BY JACQUES VALLEE, ROBERT JOHANSEN, AND KATHLEEN SPANGLER



Using ordinary telephone lines, people can now join an invisible network and attend a conference that runs continuously, 24 hours a day, for as long as the participants want. After analyzing some 5,000 hours of such computer conferences, researchers at the Institute for the Future in California believe that this unique medium can create an altered communication state. By enabling people to escape the normal bounds of time and space, computers may thus provide an opportunity to create and explore new patterns of human expression.

The authors of this article are researchers at this Institute for the Future, in Menlo Park, California. The Institute is dedicated to systematic and comprehensive studies of the long-range future. We plan to publish more articles on the Institute's work, particularly in the area of computer conferencing.

This article is reprinted with permission from the 1975 issue of The Futurist, *a journal of forecasts, trends and ideas about the future.* The Futurist *is published by the World Future Society, an association for the study of alternative futures, whose membership numbers about 25,000. For more information about the Society and its activities, contact the World Future Society, 4916 St. Elmo Avenue, Washington, D. C. 20014, Telephone 301-656-8274.*

Most of us communicate intuitively. We greet each other every morning without any thought of the contracting muscles of our vocal cords, the atmospheric support of sound vibrations, or the semantic intricacies of our language—all of which are necessary for our natural, face-to-face communication process. Suppose, however, that we had to explain face-to-face communication to someone who had never experienced it. How would we explain, for example, the necessity to be within vocal and visual range of other people? What about the possibilities for "body language," for interpreting all of the subtle visual cues which accompany the vocal symbols of face-to-face communication? And how would you introduce vocal symbols to a person who has never depended on them to communicate? How do the social demand for immediate responses and our limited ability to remember words which vanish in the air define the nature of our communication?

Clearly, the task of explaining a communication process is staggering. Yet this is the task which we face in exploring the computer conference. Most of our intuitions about face-to-face interaction simply do not apply to this new and unusual form of communication. In computer conferencing, time and distance are dissolved. Visual cues no longer exist. Each person's "memory" of what has been said is accurate and complete. And everyone may speak at once or listen at leisure. With such features, it is not surprising that computer conferencing might actually establish an altered state of communication in which the realities of face-to-face communication are distorted and entirely new patterns of interaction emerge. Our research team at the Institute for the Future in Menlo Park, California, has often experienced this altered state of timeless, placeless, remote communication during the past two years, as we developed and experimented with a family of conferencing programs. Our computerized communication system, known as FORUM, functions as an interpersonal medium for a variety of activities, including planning and forecasting, group conferencing, joint writing projects, electronic notepads (in which messages are stored in a computer instead of on paper), social simulations, and questionnaires. The system allows geographically separated people to communicate either simultaneously or on a delayed basis. We call these two basic usage modes "synchronous" and "asynchronous" conferencing. Participants do not need any technical expertise or even previous experience with computers, though they use a standard computer terminal. All of these characteristics combine to create social conditions that differ from face-to-face communication in at least three important ways: (1) the physical environment; (2) fewer time and space limits; and (3) the various communication structures which are allowed.

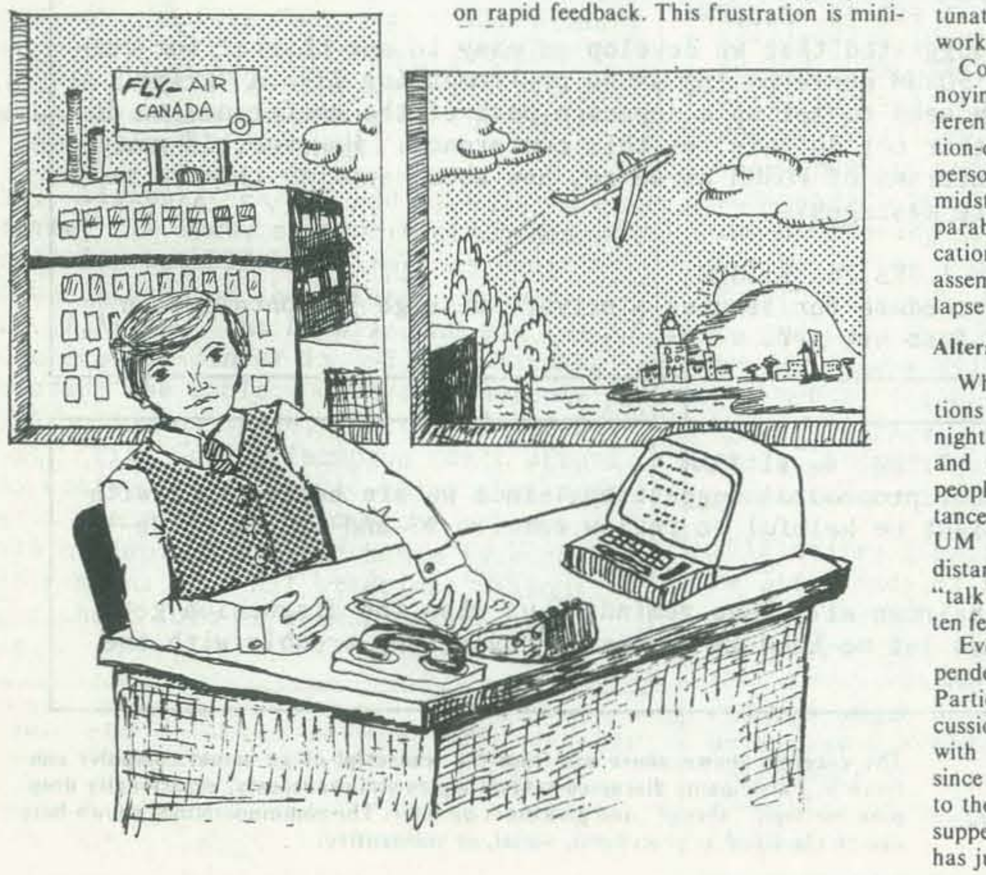
An Altered Physical Environment

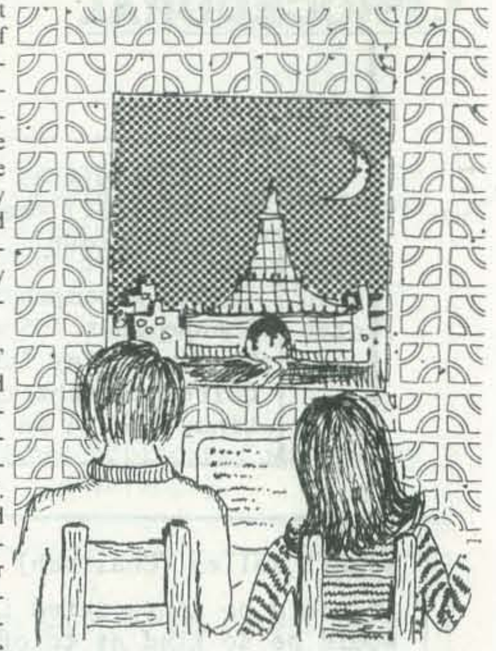
Unlike face-to-face gatherings, FORUM gatherings are characterized by physical isolation of each participant. Alone with his terminal, each computer conferee depends on an unseen computer to communicate with his colleagues. All "conversation" must be typed on a computer terminal with a standard typewriter keyboard. As a result, accessibility and reliability of terminals, typing skills, and writing skills—factors which are not even considered in face-to-face meetings—all influence communication in a computer conference. For example, a slow or uncertain typist will probably become more selective in the questions he answers and in making his own contributions. On the other hand, many users have found that typing allows them to "give more consideration and focus" to their statements. Expressing ideas through a keyboard is not always a negative factor: Ernest Hemingway reportedly preferred a typewriter for developing dialogues even though he returned to longhand for narratives and descriptions.

The remote keyboard situation hints at some interesting changes in the ritual of "meeting" people. In a computer-based conference, there are no gestures, facial expressions, or vocal cues like pitch, intonation, pauses, or stress. In face-to-face communication, these cues often regulate the flow of a discussion; they also convey emotional feelings and attitudes toward other participants. FORUM greatly narrows this field of information, and many emotional messages simply seem to disappear.

When the sole context for "meeting" someone is through an impersonal keyboard and an equally impersonal printout, the person at the other end might seem inaccessible—a mere extension of the machine. Fortunately, this is not entirely true. Many of the messages ordinarily expressed in body movement or voice tones are translated into written form, either implicitly or explicitly. One conferee reported that "relationships were established easily, personalities came across, conversations could be established." In short, people can become recognizable personalities, even when their only means of expression is the printout of a computer terminal.

The computer itself is invisible in the communication process, but it may intrude upon the discussion in a couple of ways. First, a heavily loaded computer network may transmit messages irregularly. The resultant delay can be frustrating and confusing, since satisfactory communication usually depends on rapid feedback. This frustration is minimal, however, compared to the experience of a system "crash," when the computer stops and the terminal automatically prints out a message such as "DRUM FULL" or "HOST DEAD." (The "HOST DEAD" message created considerable shock among many of our users who attended computer conferences for the first time. A British researcher pointed out to us that a more gentle announcement, such as "HOST PASSED AWAY," might be less traumatic.) Unfortunately, we have had no control over network access or reliability.





Computer system failures are always annoying, but a failure in the middle of a conference dealing with intellectual and emotion-charged issues is devastating. Each person is suddenly and totally isolated in midstream; frustration is intense. A comparable situation in face-to-face communication might be the violent disruption of an assembly by armed bandits, or a sudden collapse of the building.

Alterations of Time and Space

When people in widely separated locations can interact at any time of day or night, their "real world" concepts of time and space are drastically altered. Most people have already had their sense of distance altered by the telephone, but FORUM further reduces the consciousness of distance since it typically costs no more to "talk" across thousands of miles than across ten feet.

Even more striking is the unique "suspended time" of a computerized conference. Participants may enter and leave the discussion at will, without risk of losing touch with the meeting. Time zones disappear since discussion can proceed without regard to the fact that one user is about to eat his supper in London, while a California user has just arrived at his office. If the London

1. PROCEDURAL

2. SOCIAL

(195) Lipinski FRI 1 FEB 74 1:47 PM
Good bye all, have a nice weekend. I am going to do some work in the garden.

(196) Johansen FRI 1 FEB 74 2:24PM
I hope that 195 does not mean that this will be taken as a 9-5, Monday-Friday conference. Actually, the machine is usually quite pleasant to use on weekends, and everyone is free to continue use in an asynchronous fashion as we have been doing.

(198) Kollen (Chairman) FRI 1 FEB 74 2:43PM
It would be appreciated if participants who are logging into the system would be so kind as to offer comments concerning the agenda of this conference (the five points set down for discussion) and remarks about the discussion of the present point 1. Thank you!

(199) Johansen FRI 1 FEB 74 2:57PM
I am not sure what was meant by 198, Jim. Does this mean you don't think we are sticking to the topic, or does it mean you wish more people would make comments?

(201) Johansen FRI 1 FEB 74 5:32PM
Several people have suggested that we develop an easy to use channel for collecting responses to FORUM conferencing as we go--something like a "gripe Mode." At present we need to set up a separate part of the conference to do this, and I would rather not do this for this conference. However, if you do have comments/criticisms of FORUM as we go, how about sending them to me in the form of private messages?

(202) Johansen FRI 1 FEB 74 5:38PM
As a reminder, the procedure for sending a private message is contained in entry 41, or you can just hit a ?.

(206) Johansen SAT 2 FEB 74 11:29AM
If I could make another procedural suggestion: since we are now working with a basic agenda, it might be helpful to review entries 93 and 62, which describe that agenda.

I am sure our chairman will keep reminding us when the discussion gets off the track. Please let me know if anyone is having any trouble with the review process in FORUM.

The excerpts shown above are from the transcript of an actual computer conference. Participants discussed several topics simultaneously, occasionally dropping one topic "thread" and picking it up later. The communications shown here can be classified as procedural, social, or substantive.

3. SUBSTANTIVE

(192) Lipinski FRI 1 FEB 74 1:20PM
There is a danger of confusing the richest with the ultimate (see 189). In fact, the end of the richness scale would probably be face to face with complete visual and aural record, a very uncomfortable situation in some circumstances. Thus, for different transactions, different degrees of richness may be appropriate, and too much may be as bad as too little. Unless one considers what kind of meeting one runs, there is a danger that data will be collected across the scale of "richness"

(194) Lipinski FRI 1 FEB 74 1:39PM
There must have been reasons surely, beyond inadequate publicity, why the TV conferencing was not a roaring success (in view of the savings).

(197) Kollen (Chairman) FRI 1 FEB 74 2:33PM
The answer to Mr. Lipinski's question in 192 is yes we have collected data on how business trips are distributed across corporate activities. We have 10,000 questionnaires which have data on the following: (etc.)

(203) Johansen FRI 1 FEB 74 5:50 PM
To get back to the "spectrums of richness" question (I am afraid there are lots of different threads to this medium), wouldn't one end of the spectrum be telepathy, or complete "merging of the minds"? Though face to face certainly provides multiple imputs simultaneously, I see no reason to assume that this is the limit of communication richness.

(204) Lipinski FRI L FEB 74 6:00 PM
I agree. In fact, science fiction has 3-d vision established firmly as their communications of the future. In one book of Asimov, the heroine and the detective eat dinner in 3-d, the barrier bisecting the table. However, there is still no common space, now that I think of it. Pity Gordon Thompson can't attend this conference! I would think that merging of the minds is beyond the present state of implementation; hence face to fact remains, for all practical reasons, the richest form of communications we have.

(217)Kollen (Chairman) MON 4 FEB 7:09AM
Re 194. The Bell Canada Conference TV Trial was just that, a trial. It was not, and still is not, a market offering. It was conceived and conducted as an experiment; hence the comment about it not being a "roaring success" is not applicable as far as I can see.

(218) Lipinski MON 4 FEB 74 7:42AM
Re 194. I think you are too defensive. I hear (maybe Williams can comment) that the B.P.O. teleconference is not a great success either. Perhaps this is because we don't quite know why and how people communicate.

(219)Kollen (Chairman) MON 4 FEB 8:06AM
Re 218. Andy, I think that you may be right. Perhaps I overreacted to the words "roaring success" because I felt that there was no basis in E. Frohloff's remarks which warranted that particular criticism. Frohloff indicated that Conference TV was an experiment and not a service offering (as confravision in the U.K.)

colleague unexpectedly joins the discussion while our Californian is busily entering his ideas, this "presence" suddenly adds a dimension of intimacy which restores the awareness of space and time.

Freedom from the constraints of time and distance can naturally reduce the obligation to communicate. No one is physically present, demanding a response. No ringing telephone demands an answer. There is only the knowledge that a conference is in progress and is available, at will, through the terminal. There are, of course, a number of motivations for joining: a need for information, the need to solve a problem, a professional sense of duty, or simply the desire to "be in touch."

Clearly, there are both advantages and disadvantages to such "self-activated" communication. A participant who is asked a question feels less pressure to respond immediately than he would in face-to-face discussion. He can take time to consult a library, review his own thinking, and present a well-prepared response. Still, this same lack of pressure may be an annoyance for someone who is eager to pursue a topic with an indifferent or preoccupied colleague; however, we have found that direct questions through FORUM have generally received prompt replies. And conference growth curves, which measure the number of entries, show that the majority of conferences have constantly or positively accelerated growth rates—an indication that the momentum of the conference can generate pressure to communicate. Nevertheless, the balance between motivation and lack of demand is strikingly different from face-to-face interaction. Thus, the communication might also evolve quite differently.

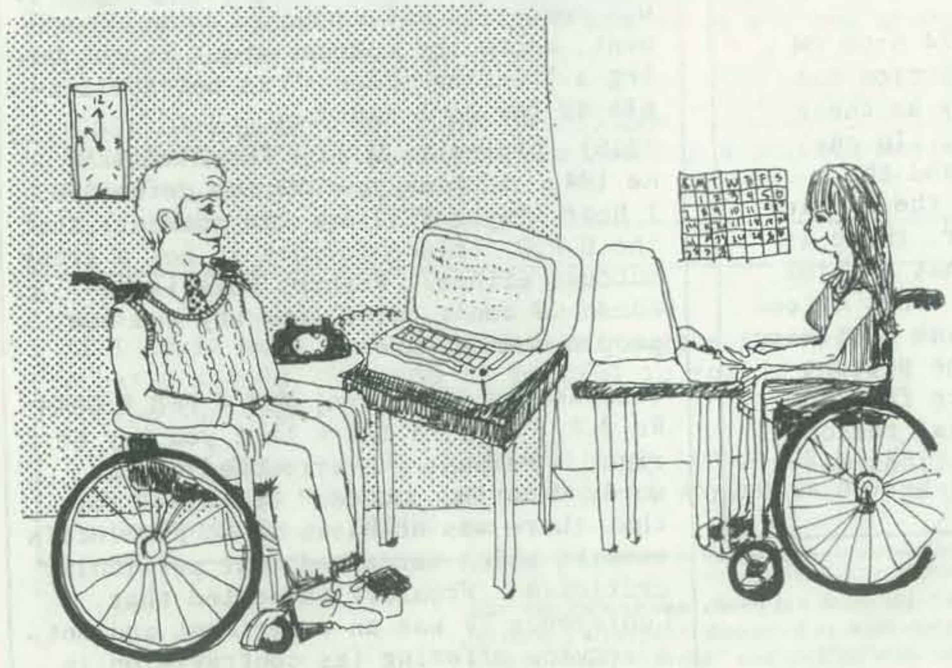
Altered Structures in Communication

Computer-based conferencing allows a great deal of control of communication structures. For example, users may send *public* messages, which are entered into the transcript and available to all, or *private* messages, which are sent to specific individuals and seen only by them. Functionally, the private message enables colleagues to "whisper" in the midst of a discussion without any breach of etiquette. In content, the public messages tend to be more formal than private messages, and more closely related to the discussion topic, while private messages include more personal interaction, sometimes quite unrelated to the main topic of group discussion.

Anonymous messages permit participants to state their views without divulging their identities—a possibility which does not exist in face-to-face meetings. Conferees have used this feature to express unpopular opinions, voice grievances, or make jokes in a way which is usually not possible.

A FORUM conference can vary from an open-ended discussion in which the topic is simply introduced and the discussion evolves without prescribed direction to a carefully preorganized discussion. In these more structured conferences, the FORUM program becomes a many-roomed meeting hall, dividing the conference into activities according to topic. For still more structured needs, FORUM will administer questionnaires or secret ballots and report the results.

In some ways, even the most unstructured computer-based conferences are more structured than face-to-face communication. FORUM discussions have been characterized by what appears to be a narrower range of topics, less diversion from the subject, and more explicit decision-making than in face-to-face conferencing. On the other hand, it is difficult to compel a FORUM user to direct his comments. It is impossible to shout down or interrupt any other person in the "meeting." All participants may "talk" at the same time; the computer simply records the entries according to the time at which the user began typing.



Mapping the Altered State

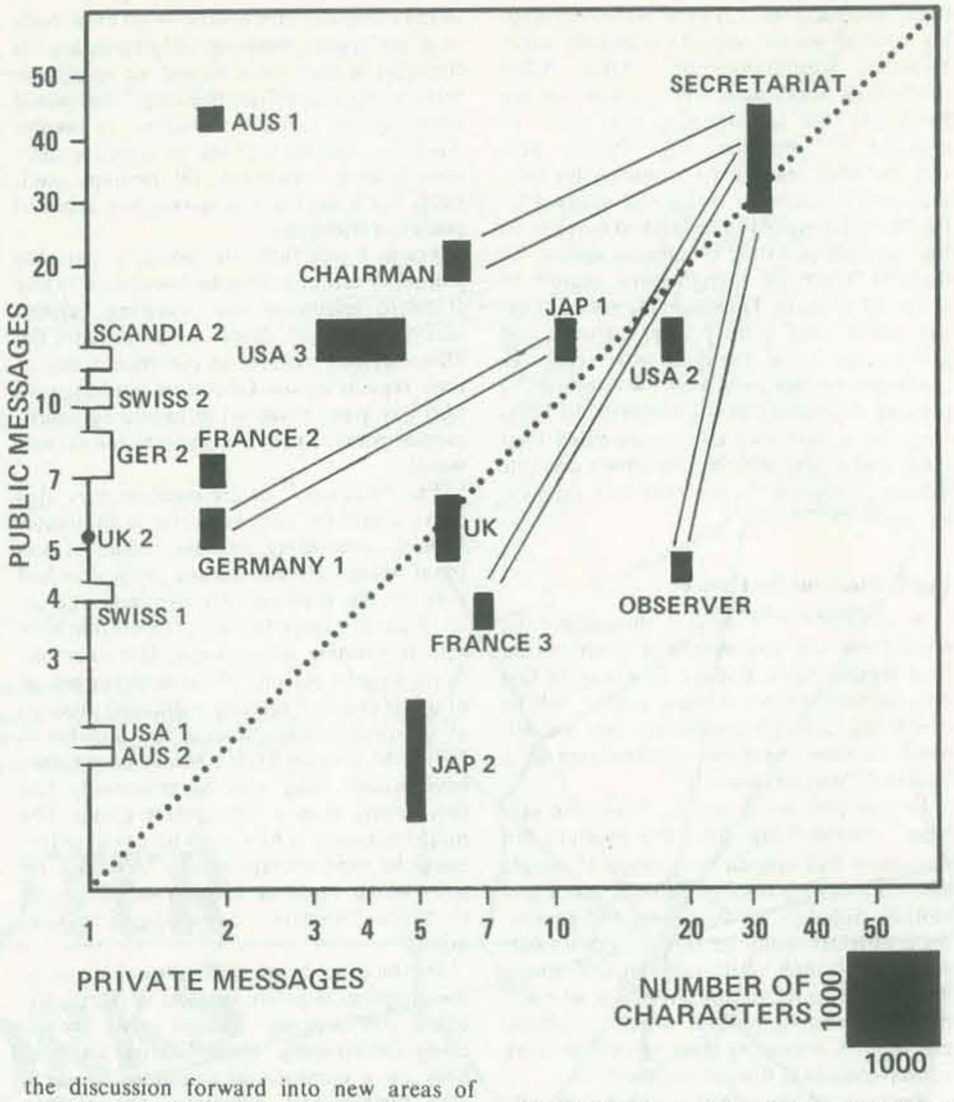
We have now begun to "map" the altered state of communications that arises from the special characteristics of a computer conference—physical isolation, dependence on the computer, suspension of time and space, reduced obligation to communicate, and a new set of communication structures. Each communication medium is a unique instrument with characteristics all its own. Because we are most familiar with face-to-face voice communication, we tend to make it a standard by which to measure other media. But we must be careful not to overlook the innovative patterns and opportunities of a new medium by clinging to our preconceptions of what communication really is. Just as it would be unfair to judge a piano by the narrow range of the human voice, it is misleading to evaluate computer conferencing as a simple substitute for face-to-face communication.

The social aspects of communications media have rarely been evaluated, and starting points are not easy to find. Perhaps as many as 50 researchers in the world are doing work on the social effects of different media in at least ten different locations. The theoretical basis for this work is rich, but scattered. The computer conferencing medium itself provides two powerful analytic tools for evaluating its social characteristics: (1) an up-to-date machine-readable transcript of every computer meeting is always available and (2) the computer can unobtrusively map interpersonal interactions to reveal patterns of communication among individuals, groups, and subgroups. Each of these points deserve elaboration.

1. A complete transcript of every computer conference is always available, current, and machine-readable. This transcript is automatically recorded exactly as it is typed, and members can review the record by subject, author, and date—during and after the conference. The possibilities for analyzing the content of the discussion are thus greatly improved over most other media. Using one analytic technique, we have classified entries by content, identifying them as regulatory comments dealing with the group process, comments on the substantive topics in the conference, humor, novel ideas, and similar classifications. In this way, we can evaluate a group's ability to focus on a particular task, and we can also determine where the time actually went.

PARTICIPATION MAP FOR A SIMULATED COMPUTER CONFERENCE

This sample participation map was constructed with statistics *gathered directly* in a simulation of a computer-based international conference. In this test, the work of an International Telephone and Telegraph Consultative Committee Study Group was simulated by 18 graduate students at San Jose State University, assisted by several technical experts. The students played the roles of eight national delegations from which previous position papers were available. An analysis of user behavior with respect to negotiation and information exchange was then conducted. Rectangles represent each participant, indicating the number of private messages and public messages each has sent. The sides of the rectangles are proportional to the verbosity, defined here by the average length (in characters) of messages in private and public mode. This type of map can help define roles of participants; it also enables us to observe coalitions and subgroups and to track individual participation characteristics from one conference to another.



The transcript also makes it possible to track specific discussion topics over time. We have thus identified a strong tendency for "threads" or "chains of thought" to occur in the conference transcript. These topic threads are frequently labeled ("re comment 13,"), but the tie is sometimes only implicit, requiring readers to review the earlier proceedings to find out what has been said on a particular topic. The review process is supported by the FORUM program: a participant can, for example, request the computer to search for any entries which mention a particular word, such as "energy." In general, our analysis of the topic threads shows that it is possible to discuss several topics at the same time, occasionally dropping one thread and then picking it up again later.

In addition to tracking the content of discussions, topic threads enable us to analyze the role that the different participants play. We find that some persons tend to introduce many new ideas, while others are best at developing them; still others function as synthesizers. The roles can vary greatly among persons and conferences, but we have noticed an apparent tendency for the "provocative" and "synthesizing" roles to be mutually exclusive. The provoker seems to push the discussion forward into new areas of thought, while the synthesizer ties the loose strands together. By examining the patterns of a FORUM conference, one can easily identify both key persons and key ideas.

2. The computer itself can unobtrusively map many dimensions of the interaction that may or may not be evident from the transcript. The ability to map these interaction patterns within a conference may be the most powerful analytic tool inherent in any communications medium. This capability of the FORUM program means that the detailed coding and painstaking observation of interpersonal communication that social psychologists must typically carry out in analyzing small groups can be done automatically here, without disturbing the normal communication process. Comparative participation rates, growth curves, daily activity, and other related indicators create new dimensions for assessing group interaction. Private message statistics, for example, may indicate the formation of subgroups, cliques, or coalitions. Such statistics even allow us to trace individual participation characteristics from one conference to another, perhaps as a function of topic and task.

In addition to individual characteristics of participation, we can also evaluate group characteristics with growth curves. When plotted for the content categories, for example, these curves can indicate if and when the conference has made a transition from the procedural questions inherent in any meeting to the solution of substantive issues.

It is difficult to think of another medium in which an analysis of group interaction can be automatically and unobtrusively generated with this level of detail. At the same time, the privacy of the conference is not violated. The statistics about interaction can be compiled independently of the content of the conference; conferees must grant their permission before we can make any comparison of personal interaction and content.

We have evaluated over 25 conferences using these and more traditional analytic techniques (including interviews and questionnaires). In general, our user groups have had the following characteristics: (1) little familiarity with computer systems; (2) a genuine need to communicate with each other; (3) group sizes ranging from 3-20, but averaging about 5; (4) tasks which were relatively unstructured; (5) time periods averaging several weeks; and (6) primarily asynchronous communication. After 5,000 conference hours with these groups, we are convinced that long-running field tests—as opposed to laboratory experiments—provide the most realistic environment for fully exploring conference styles and usage. FORUM was designed to be learned quickly, so that new users would be able to master its features after an introductory period of about 15 minutes. However, the styles of usage could vary greatly after persons and groups are more familiar with computer conferencing and with their own abilities to present themselves in the medium. In long-term tests, attitudes can be sampled over time, and evaluations become more credible as users integrate the medium with their everyday lives.

Implications for the Future

A scant 100 or so persons throughout the world now use computerized conferencing on a regular basis. But the time may be fast approaching when far more people will be conferring through computers and we will begin to view computer conferencing as a "natural" way to interact.

In this new environment, "invisible colleges" may develop, since this medium can introduce and coordinate groups of people who may or may not have been in touch previously. Scholars, businessmen, and government officials would be able to interact outside the normal limits of time and space; they would no longer need to spend so much time exchanging journal articles, memos, and reports, arranging meetings, or traveling to conventions in distant places.

Perhaps we can enhance group creativity through a new communications style, forged in the computer conference. With everybody at a conference thinking and expressing his thoughts in multiple streams, we might observe a process of "fast thinking" that would enhance our collective abilities to resolve conflicts, deal with crises, or improve decision-making capability. Or perhaps computer conferencing will spawn new types of poetry or literature.

From a practical viewpoint, a portable computer terminal may be hooked up to any standard telephone line, enabling persons immobilized with illness, or away from the office for any reason, to continue many of their regular duties. Computer conferencing also has great potential in providing handicapped persons with a channel to the outside world.

The "coolness" of the medium may also prove useful for such activities as encounter sessions, counseling, and discussions of personal values. Psychotherapy may also find uses for the medium: for instance, the altered environment for self-presentation may help in defining self-concepts (for example, in relating to persons of the opposite sex or of other races). Certainly a detailed analysis of self-presentation processes is possible in FORUM, and the FORUM communication environment may also be potentially less threatening than a face-to-face group. The major question is how such an environment could be used therapeutically to obtain results which could be transferred effectively to more "normal" communication situations.

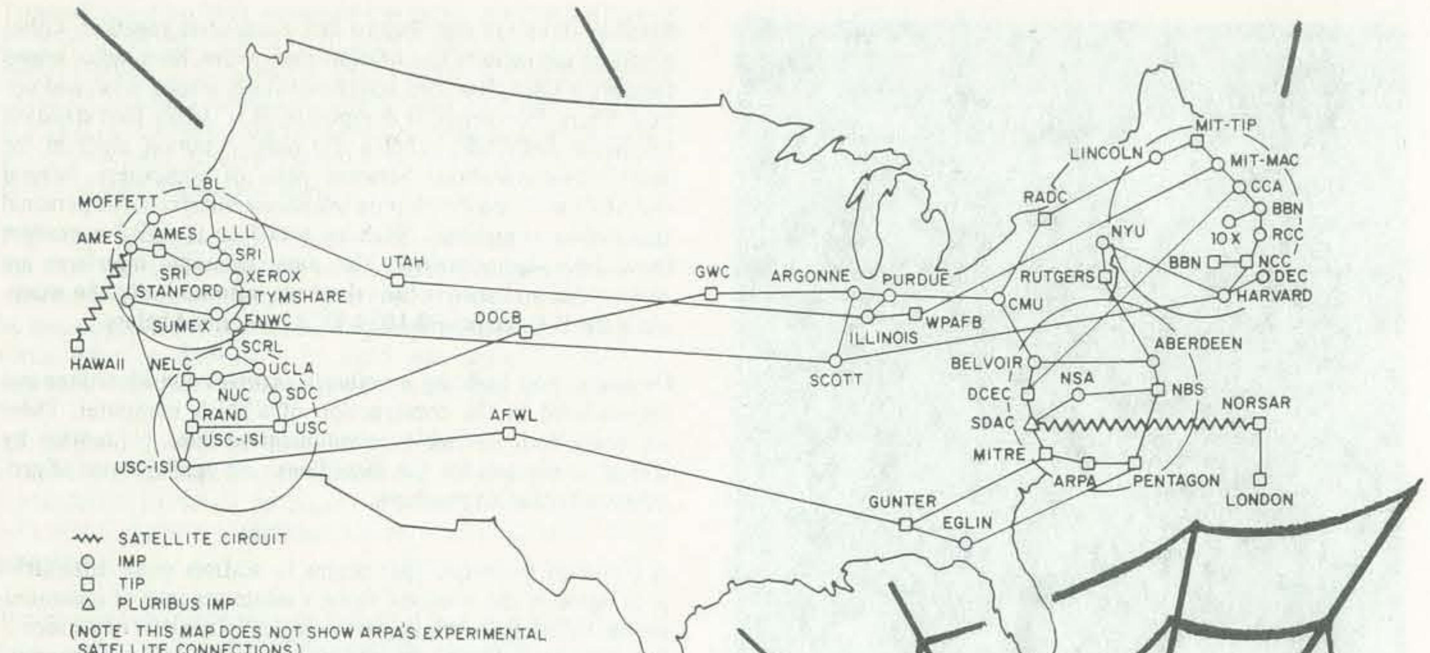
On the other hand, we are not oblivious to the potential negative impacts of computer-based conferencing. Though costs are already encouraging (about $15 per terminal hour on a commercial computer network, with further cost reductions anticipated), computer conferencing is not yet a medium for the masses. And a type of electronic elitism is certainly a possibility as long as terminals and network access remain the privilege of a few.

Could computer communications replace much—or all—face-to-face contact? At present, "human contact" usually means being together "in person." For some people, the mere thought of a communication medium in which human bodies (or even voices) are irrelevant is frightening. Isaac Asimov, in his novel *The Naked Sun*, and E. M. Forster, in his 1929 story *The Machine Stops*, offer nightmarish projections of a future in which electronic communication replaces human contact as we now know it. Our research team has examined computer-based conferencing as a supplement to face-to-face communication, not as a replacement, but long-term negative possibilities deserve attention, if only so they can be avoided.

Our studies to date indicate that computer conferencing has unique potential for enhancing the exchange of ideas among people. In current field tests, we are exploring its usefulness in bargaining and negotiation, conflict resolution, crisis management, and some educational applications. However, as should be clear from this article, our work should only be viewed as a foot placed in an interesting door. We are convinced that this medium will change quickly and that it should not be evaluated by narrow criteria. We believe as well that the potential of computer-based communication remains largely unexplored.

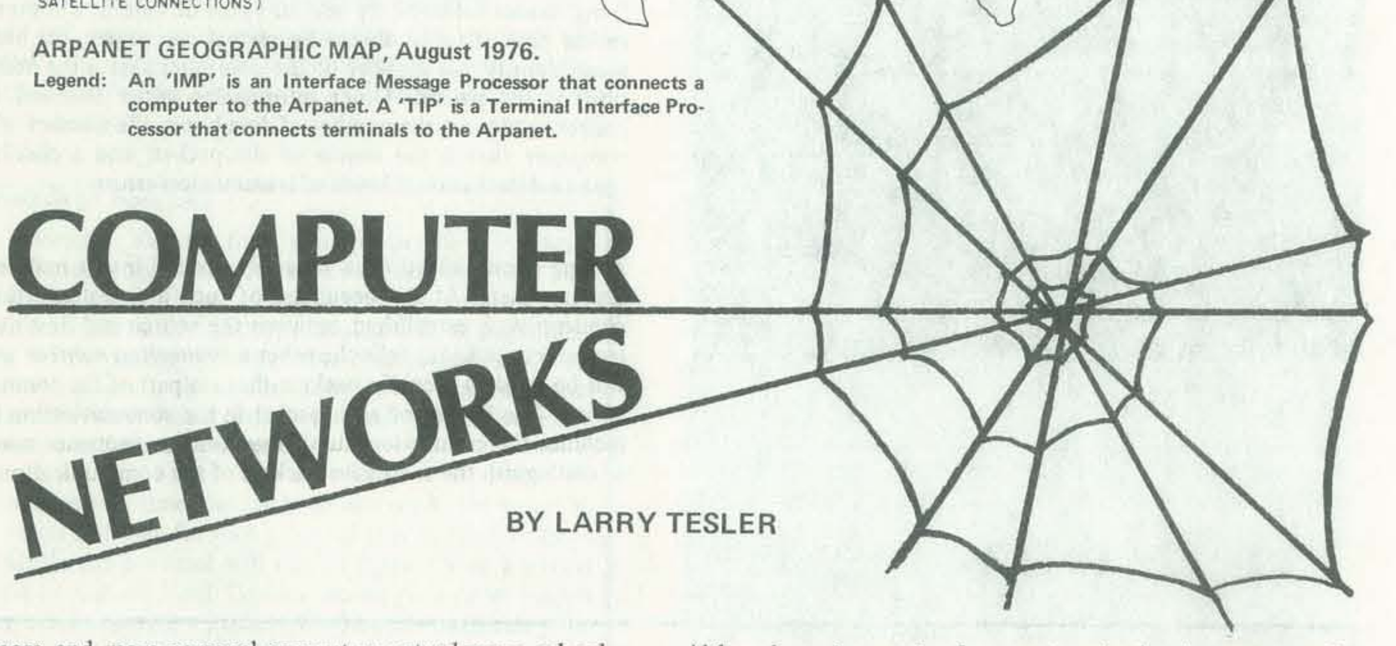
• • •

The authors, Robert Johansen, left, Kathleen Spangler and Jacques Vallee. The authors are researchers at the Institute for the Future, 2740 Sand Hill Road, Menlo Park, California 94024. The FORUM system has been developed by a team composed of Roy Amara, Hubert Lipinski, Ann McCown, Richard Miller, Thad Wilson, and the authors of this article. This research is supported by the Department of Computer Research at the National Science Foundation (under Grant GJ-35 326X). The authors wish to thank Arthur Hastings for suggesting that computer conferencing might represent an 'altered state.' PLANET-1, a simpler version of FORUM, is now available on the TYMSHARE, Inc. computer network.



ARPANET GEOGRAPHIC MAP, August 1976.

Legend: An 'IMP' is an Interface Message Processor that connects a computer to the Arpanet. A 'TIP' is a Terminal Interface Processor that connects terminals to the Arpanet.



COMPUTER NETWORKS

BY LARRY TESLER

As more and more personal computers enter homes, schools, and other institutions, it is useful to link them together in networks so that people can send electronic mail (computer telegrams) and can access information in community memories (computer libraries). As an added benefit, software can be distributed quickly over telephone lines.

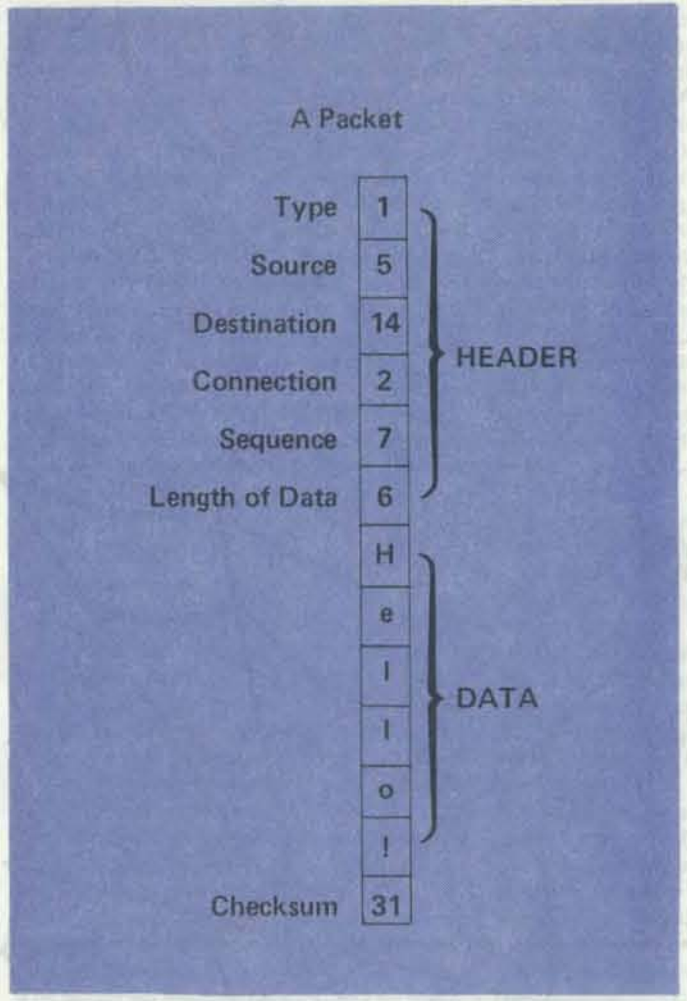
A computer network is a community of interconnected computers. The computers may have different speeds and memory capacities and be made by different manufacturers. For example, the Arpanet of the US Department of Defense interconnects more than 150 computers owned by over 60 universities, government offices, and industrial contractors as far apart as Massachusetts and California. The computers are made by 13 different manufacturers and use more than 20 different operating systems. (An operating system is a relatively permanent program that schedules other programs to run, and provides services to other programs such as controlling external devices.)

Although a given pair of computers in the Arpanet may be very different and may be thousands of miles apart, they are able to communicate with each other because two basic requirements have been satisfied: *path* and *protocol*, that is:

1) There is a communication *path* through the network from one computer to the other, and

2) All the computers in the path are programmed to communicate according to an agreed upon *protocol*.

Networks of computers may be linked to other networks through *gateway* computers. For example, the Arpanet has a satellite link from Alexandria, VA to a gateway in London, England and one from Sunnyvale, CA to a gateway in Honolulu, Hawaii. When the protocols of two networks are different, the gateway between them must know both protocols in order to relay information from a computer in one network to a computer in the other.

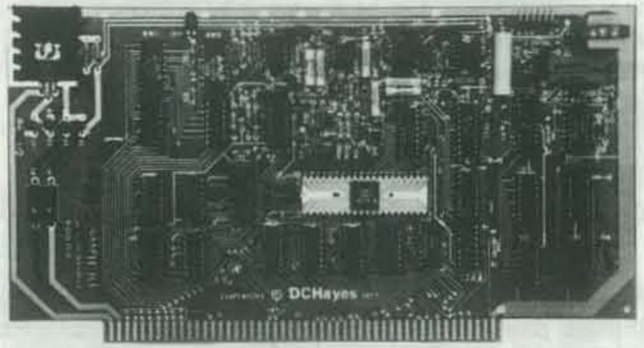


Satellite links are one way to link computers together. Other media in use include line-of-sight microwave, ham radio, leased telephone lines, standard telephone lines, strung wire, and optical fibers. For personal computers, it is likely that standard telephone lines will become the most common medium for short communications between pairs of computers. Several manufacturers are developing telephone interfaces for personal and hobby computers. Such an interface is called a *modem* (modulator/demodulator). The most desirable interfaces are able to dial and answer the telephone automatically. An example is the D.C. Hayes 80-103A DCA, described below.

Designing and building a network involves considerations not encountered in the construction of a single computer. There are noisy and unreliable communication lines, contention by several computers for the same lines, and malfunctions of geographically distant machines.

A common technique that begins to address these difficulties is to agree in the protocol upon a minimum unit of communication called a *packet*. A packet typically consists of an identifying header followed by several bytes of data. If a communication path is to be shared by several computers, the header must identify the number of the computer that is the destination of the packet. Other information often encoded in a packet header are the number of data bytes, the number of the computer that is the source of the packet, and a *checksum* used to detect certain kinds of transmission errors.

A long communication is generally divided into a number of small packets. At the beginning of such a communication, a *connection* is established between the source and destination computer, and each tells the other a *connection number* which will be used to identify packets that are part of the communication. The header of each packet in the communication then includes the connection number as well as a sequence number to distinguish the successive packets of the communication.



Retail price is $279.95; bare board with manual, $49.95; manual only, $7.50. Shipments are 4 to 8 weeks from receipt of order. No kit production is planned at this time. Further information available on request from D. C. Hayes, P.O. Box 9884, Atlanta GA 30319.

80-103A Data Communications Adapter

Data Communications Adapter (DCA), is an Altair compatible asynchronous serial interface incorporating a fully programmable frequency shift keyed (FSK) modem.

Programmable freatures include auto dial and answer, originate answer mode, data rate 110-300 BPS, echo suppress tone generator, error detection and self test for complete internal verification. Other features are fully buffered bus, outputs drive over 25 Altair bus loads; complete digital modulation and demodulation with precision filter – requires no adjustments or fine tuning; option for interrupts on ringing, transmit register empty, receive register full; switch selectable address. The double sided PC board with plated holes, gold plated card edge connector, solder mask both sides, and silk screened component designation is available as an assembled unit or as a bare board.

To send a multi-packet message from computer S to computer D, S sends all the packets to D, and D sends back *acknowledgments* ('acks') to S. Acks are special short packets stating the sequence numbers of regular packets that have been received. If S fails to receive an ack for a certain packet within an agreed upon time limit, it retransmits that packet. When the ack is received, S may discard the packet.

There can be many reasons for an ack not to arrive at S within the time limit. The packet could be lost or garbled on the communication path (packet 3 in the diagram). The ack could be lost or garbled (ack 4 in the diagram). Bottlenecks along the communication path or in D could have unduly delayed receipt and acknowledgment (packet 5 in the diagram).

Let us look at each of these cases. After packet 3 is lost, retransmission (packet 3R) restores order. After ack 4 is lost, retransmission causes D to receive a duplicate packet (packet 5R), but S receives a duplicate ack (ack 5R), which it simply ignores.

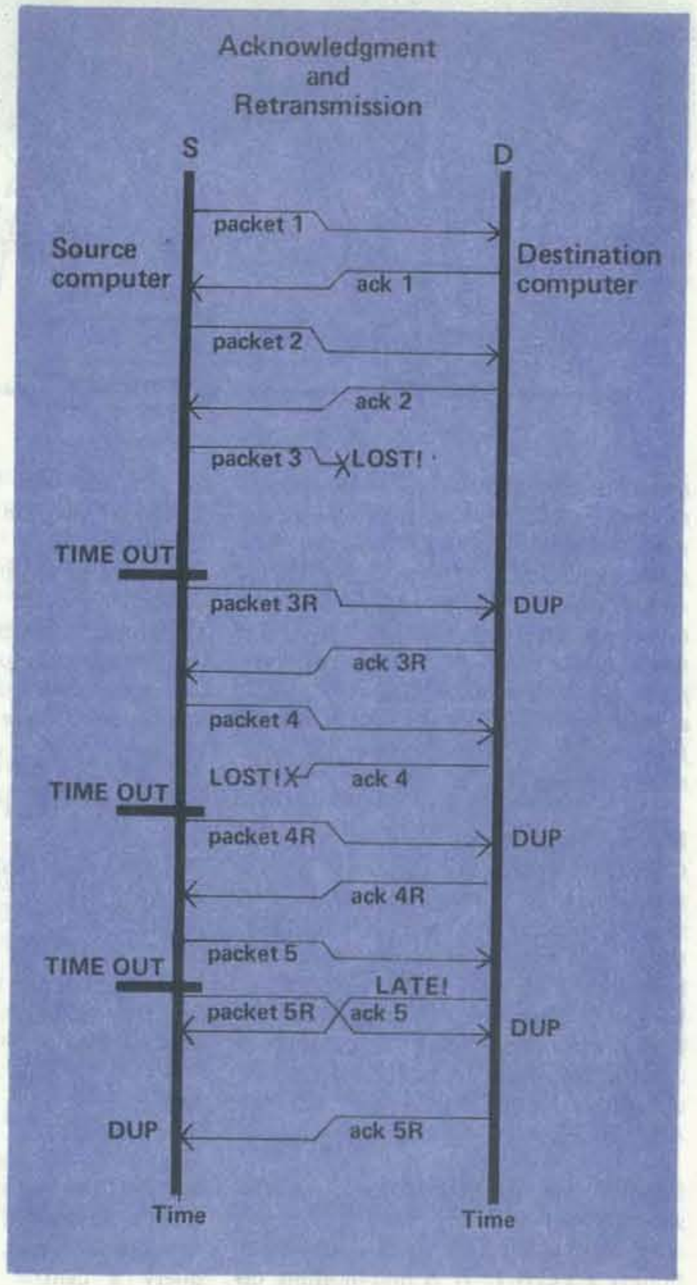
The packet/ack protocol just described has been implemented with variations on different networks depending upon the speeds and capacities of the computers, the topology of the network, the quality of the communication lines, the average and peak traffic, the average and peak communication delay, and other factors.

Where to go from here . . .

An interesting exercise for a group of people is to specify an assumed set of conditions on a network, then have subgroups of one to three people each define a protocol for those conditions. The protocol should include a specification of the packet types (regular data packet, ack, start connection, etc.), the header format, and the rules that each computer should follow to send, acknowledge, retransmit, and discard packets.

After a certain time, say, an hour or a week, the group reconvenes and then for each proposal tries to think of cases in which the protocol will fail. Is there a way a packet could be forever lost? Could a sender go into an endless loop retransmitting a packet? Could a communication be unduly slow because the timeouts are too long and too frequent? Could the communication lines be overrun with duplicate packets because the timeouts are too short? Is the header so long that little line capacity is devoted to data? If the communication is over a private line between the source and destination computers, those are some of the possible problems. If several computers contend for the same shared line or the same radio frequency, collision problems must be addressed as well.

Expect weaknesses in every proposal. Can you devise a systematic way to verify the soundness of any proposal? It is possible to simulate many of the aspects of a computer network in a computer model on one computer. It is much harder but more instructive to actually build a network of several computers. The best way to further the development of electronic mail and community memory is to join an existing network in your geographical region.



Bibliography

Cerf, V.G. and Kahn, R.E., 'A Protocol for Packet Network Intercommunication,' *IEEE Transactions on Communications*, vol. Com-22, no. 5, May 1974.

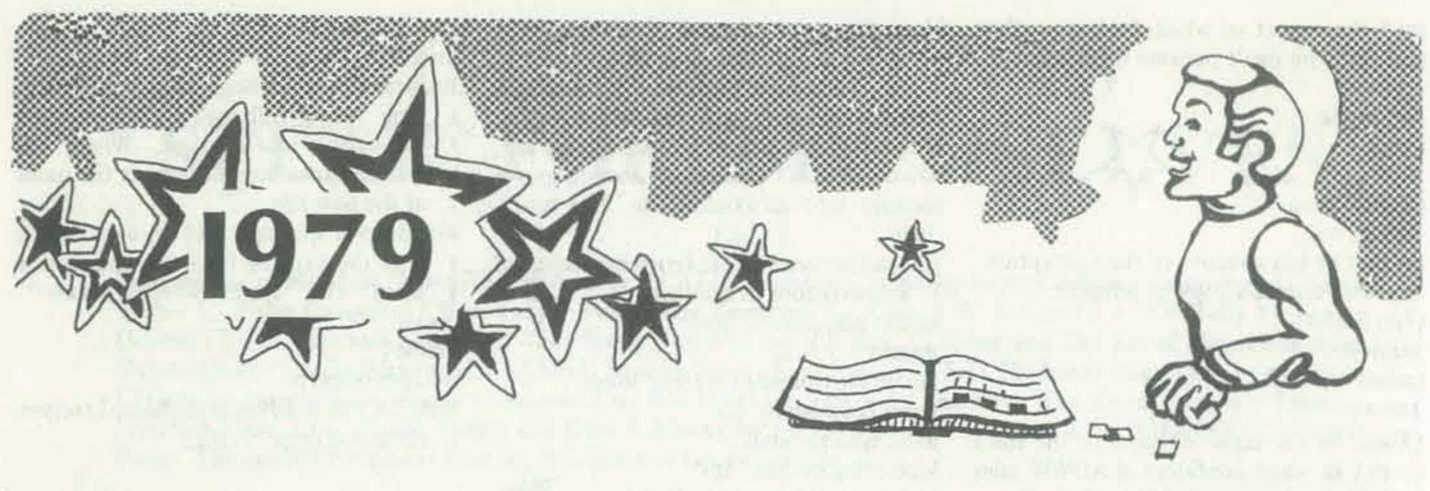
Farber, D.J., 'A Ring Network,' *Datamation*, February 1975.

Crowther, W.R. et al. 'Issues in Packet-Switching Network Design,' *Proceedings of the National Computer Conference*, May 1975.

Metcalfe, R.M. and Boggs, D., 'Ethernet: Distributed Packet Switching for Local Computer Networks,' *Communications of the ACM*, vol. 19, no. 7, July 1976.

'Community Information System', *Northwest Computer Club Newsletter* (10/76); reprinted in *People's Computer Company* Vol 5, No. 5, Mar-Apr 1977.

PCNET 1979



BY DAVE CAULKINS

Dave Caulkins has worked on computers for longer than he cares to remember, with one short digression into the biofeedback business. He is currently involved with electronic mail systems at Cabledata Associates. In the past he has worked on a number of distributed computer systems, including the ARPANET, a microprocessor based satellite system for the military, and a large office automation system.

In the July-August issue of *People's Computers* (page 31) there was an announcement of the beginning of the Personal Computer NETwork (PCNET) in the Palo Alto, California area. A feel for how such a network might be used can be gotten from the following bit of sci fi (more sci than fi; all the capabilities covered below are available today on big computers and/or big systems like the ARPANET) set two years in the future.

Imagine that you are looking over the shoulders of a family using a personal computer connected to (among other things) a PCNET. It is mid-morning on Saturday, the 29th of July, 1979. Linda Smith is making up menus for the next few days' meals. She sits at the personal computer keyboard and types.

In the text that follows, a line beginning with '(' indicates commands typed by a human; a line beginning with ')' indicates the responses of the computer system.

```
(getfile
)name?
(recipes
)recipes.current.03june79.0843
(show desserts 'chocolate'
```

This command causes the computer to search the desserts' section of the file for all entries with 'chocolate' in the title.

```
)chocolate bavarian cream
)chocolate cake
)chocolate icebox pudding
)chocolate mousse
```

The recipe she was thinking of is not in the file; she calls to her husband, 'Jim, do you remember that chocolate zucchini upside down cake we had last week?' Jim says 'Yeah, it was at the Henderson's dinner party.'

Linda says 'Thanks' and continues typing.

```
(mail
```

This command calls the mail handling system.

```
)send?
(y
```

'Yes', 'OK' or the name of a mail system function other than 'send' would also have caused the computer to take the correct action.

```
)to?
(mary henderson
```

The computer gets Mary Henderson's address from an 'address book' file; if her address was not in this file the computer would ask for it, or for permission to query a central PCNET address file.

```
)copies to?
```

Linda's message is not important and she wants no copy for herself or anyone else. She types:

```
(n
)subject?
(chocolate upside down cake recipe
)message?
(Mary: Could you please send me your
(  recipe for that delicious chocolate
(  zucchini upside down cake we had at
(  your party?
(Linda#
```

The '#' character signals the computer that this is the end of the message.

```
)trying . . . . .
```

The above message appears when the computer initiates a process that takes a period of time of more than a few seconds. In general, the computer-to-human interface operates in 'humble servant' mode; the system goes to considerable lengths to keep the user fully informed and happy. It avoids making demands and being unresponsive, cryptic or arrogant.

```
)mary  henderson  40.45N83.45W419–
)  555–1213  msg  delivered  10:11AM
```

The string of characters after the name is Mary Henderson's PCNET address.

Linda goes back to using the recipes file (it was not taken out of the working space while she used the mail system). When she finishes she stores the new menu file and leaves the computer.

Jim is the secretary of a local stamp collecting club; he sits down at the keyboard to arrange the next meeting.

```
(mail
)send ?
(y
)to?
(stamplist:
```

'Stamplist' is the name of a file containing the names of all the club members. The ':' at the end tells the system that this is such a list and not someone's name.

```
)copies to?
(n
)subject?
(stamp meeting – 9 August
)message?
(The next meeting of the Fernwood Phil-
(  atelic Society will be at my house,
(  7:30PM 9 August. See you there.
(Jim#
)trying . . . . .
)john  anders  40.43N83.32W419–554–
)  1223 msg delivered 10:28AM
)bill  baker  40.16N83.12W419–554–
)  3758 waiting–addressee mailbox un-
)  available
)joel  cairo  40.32N83.12W419–555–
)  9476 msg delivered 10:31AM
)tom hartman 40.08N83.24W419–555–
)  6593 msg delivered 10:33AM
)  .
   .
   .
```

And so on through the entire Stamplist file. As indicated above, the system gives a status report on each message, either a positive indication of delivery or a reason why this has not occurred. If a message cannot be delivered the system will keep trying.

Jim is employed by the Acme Digital Widget Works (ADWW). He is collaborating on a sales projection report with a colleague at ADWW, Tom Wilson. Jim types:

```
(getfile
)name?
(salesproj.draft
)trying . . . .
)not a local file; please give me a data
)  base address where it might be.
(adww
)trying . . . .
)salesproj.draft.26july79.0927 ready
```

Jim notices that the most recent change date and time (26july79.0927) is more recent than his last reference to the file; Tom must have been working on it. Jim reviews what Tom has done and makes further changes of his own. Then,

```
(putfile,
```

The ',' after putfile signals the system that additional qualifying sub-commands follow the 'putfile' command.

```
(to remote,
```

At this point in time Jim suffers a slight attack of paranoia; the changes he has put in the file include some information on ADWW sales that competitors would love to see. He decides to encrypt the file, to convert it into a pseudo-random data stream which can be read only by someone else with the same encryption key.

```
(encrypt
)remote database name?
(adww
)access password?
```

Jim enters a code word allowing him access to write a file at ADWW; it is not printed or displayed.

```
)encryption key?
```

Jim enters a key sequence that will be used to encrypt the file before it is sent out by the mail system; it is not printed or displayed.

```
)name?
(*
```

The '*' character tells the system to use the same name for the reviwed file, changing only the date and time.

```
)trying . . . .
)salesproj.draft.29july79.1103
)  stored in data base at acme digital
)  widget works
)you have new mail
```

Jim types:

```
(mail
)send?
(read
)new messages are:
)14 Subject: reply to msg 29 July 79 1011
)     To: Linda Smith
)     From: Mary Henderson
)     Received: 29 July 79 1058
)13 Subject: reply to msg 29 July 79 1031
)     To: stamplist:
)     From: Joel Cairo
)     Received: 29 July 79 1046
)12 Subject: you may have already won
)     our contest!
)     To: Jim Smith
)     From: The On-Line Digest
)     Received: 29 July 79 0614
```

Jim says, 'Hey Linda – Mary's sent a reply to your message. Give me a few minutes to finish up here and you can read it out.' Linda responds,'OK. We should really consider getting a second terminal and another 64K of memory; one of us is always waiting to use it and the response time is getting long again.' Jim says, 'Yeah, I guess so, but I'm not sure we can afford the $150 this month. Besides, we seem to get more and more junk mail. There's another of those cheapie early morning special promos in the mailbox.' He types:

```
(delete 12
(type 13
)13 Subject: reply to msg 29 July 79 1031
)     To: stamplist:
)     From: Joel Cairo
)Hey guys: I have to work late that night
)  –OK to change the time to 9PM?
)Joel
(answer 13
```

The 'answer' command gets all the necessary addressing information from the message being answered; all Jim must type is the text.

```
)message?
(OK by me; anyone else have objections?
(Jim#
)trying . . . .
(john  anders  40.43N83.32W419–554–
(  1223 msg delivered 11:12AM
)bill  baker  40.16N83.12W419–554–
)  3758 msg delivered 11:13AM
)  .
   .
   .
```

And so on, as before.

Jim is about to turn the system over to Linda when he remembers his paranoia of a few minutes ago. Tom Wilson will be more than slightly annoyed if he tries to read the report of which he is co-author and finds he can't because it has been encrypted.

Jim types:

```
(send
)to?
(Tom Wilson
)copies to?
```

So that he has a record of the encryption key, Jim directs a copy to himself.

```
(Jim Smith
)subject?
(salesproj draft
)message?
(Tom: In my latest changes to the file I
(  put in some confidential ADWW sales
(  info; so I encrypted the whole thing.
(  Use key #2304 from the ADWW book
(  to decrypt it.
(Jim#
)trying . . . .
)tom  wilson  41.30N81.35W213–321–
)  4849 msg delivered 1148
```

Linda sits at the keyboard and types:

```
(show messages from 'henderson'
)14  Subject: reply to msg 29 July 79 1011
)     To: Linda Smith
)     From: Mary Henderson
)     Received: 29 July 79 1058
(type 14 no header
)Ingredients: 4 eggs, 1 cup sugar, 1/4
)  pound semi-sweet chocolate, . . . . and
)  so on through the rest of the recipe.
```

Linda decides to add the recipe to her recipe file; she pauses and says to herself, 'Damn. I don't remember how to put a message into an exisitng file.' She types:

```
(help
)the active systems are (currently selected
)  abbreviations in quotes):
)audio play/record 'audio'
)games
)home environment system 'home'
)local file system 'file'
)mail system 'mail'
)operating system 'ops'
)remote resource access 'rem'
)TV play/record 'TV'
)word processing 'word'
)for more information, type system name
)  and ?
(mail?
)mail system commands are:
)send
)answer 'ans'
)file into message 'f-msg'
)message into file 'msg-f'
)read
)type
)for more information or sub-commands,
type command and ?
(msg-f?
)to transfer a message into a new file,
)  type 'msg-f' followed by the message
)  number(s); hit return. When the
)  system types 'name?' give it the name
)  of the new file.
)to insert a message into an existing file
)  type the existing file name and section
)  when the system types 'name?'.
(msg-f 14
)name?
(recipes desserts
)msg 14 put in desserts section of recipes.
)  current.03june79.0843
```

Conclusions

This little story gives my ideas of how future personal computer/communication systems might look. I (along with many others) believe that there are no technically mandated boundaries between computers and communications. There is no reason why personal computer users cannot use a PCNET-like communications system to access resources many miles away almost as easily as those physically connected to their own systems.

We've talked about the idea of the 'cheep' computer, and here's the first, expected to be available by the time you read this. You'll be hearing lots more about the PET in these pages and elsewhere in the near future.

Commodore's PET is a self-contained, factory-assembled unit that contains a 6502 microcomputer, keyboard, CRT display (40 columns, 25 lines), 1000-baud tape cassette, and memory. For $595 you get 4K of user memory (or 8K for $795) plus the 14K needed by an 8K BASIC interpreter, a 4K operating system, a 1K diagnostic routine, and 1K machine language monitor. The PET's expanded 8K BASIC contains strings, integers and multiple dimension arrays. It has high precision (10 significant digits), floating point numbers, and direct memory access through PEEK and POKE.

The system weighs 44 pounds, is 16.5 inches wide, 18.5 inches deep and 14 inches high – about the size of a portable TV but a somewhat more awkward shape to handle. The 73-key calculator-style keyboard is upper case only; a calculator style numeric keypad is included. The 64 ASCII characters are available without using a shift key; the shift key makes 64 graphic and reverse field characters accessible from the keyboard. The graphic characters can be used to play games, plot, or draw pictures.

In late July, a motley crew interviewed Chuck Peddle, father of the PET, at Commodore. The interview panel included True Seaborn and Dennis Allison of IEEE's Computer *magazine, Don Inman and LeRoy Finkel of* Calculators/Computers, *stockbroker Jon Krass, vice president of E.F. Hutton and Company, and yours truly, Phyllis Cole of* People's Computers. *Here are some facts to add to the many rumors that are floating about.*

THE MARKET

Q: Chuck, how does Commodore view the PET's place in the computer field?

A: The major philosophical difference between what Commodore has attempted and anything that's happened until now is that we've tried to make a product that is merchandisable by a normal retailer to the ultimate consumer. That's why the PET has the built-in CRT, the built-in cassette, and the lower-cost, calculator-type keyboard. For a sale to be made by an inexperienced retail clerk to an inexperienced customer the unit has to have immediate perceived value. The only way to have immediate perceived value is for the unit to do something the customer wants as soon as it's plugged in. People have been taught that computers are difficult to operate, that computers are things to be afraid of, not things to get warm and friendly and cuddly with. Therefore, what we've tried to do is to package the unit in such a way that it's as close to warm and friendly as we can get it; but it has perceived value as a thing that does something.

The analogy between the way the PET will be marketed and the stereo market, I think, is very, very close. In K-Mart stores stereos are sold by retail clerks who have no real understanding of the product. We expect the PET will be successfully sold in a similar fashion.

Q: Does such an approach presuppose some mass-media advertising?

A: Oh, absolutely. It is our opinion that the Average American Public has no concept that personal computing is here. News about personal computing is starting to break in the media, but it has got to be mass-advertised and mass-merchandised. We expect to spend a significant amount of dollars relative to the size of our corportation on advertising, such as television, that is non-traditional for the hobbyist market.

Q: Does that mean you're going to go into retail marketing directly?

A: We're already in retail marketing directly. The Mr. Calculator stores are a legitimate test market. That doesn't imply that we're not dealing with dealers – we are. At the same time, there will be at least one major retailer handling the product on a nationwide basis simultaneously.

Q: You're thinking about September delivery?

A: That's to the mass retailer. We will not necessarily have the PET to smaller retail stores at that time.

Q: How do computer stores fit into your marketing plan?

A: We feel that the two types of stores are complementary. Stereos became something you could buy in the discount stores, but you still have the specialty stores. So we feel that the well-financed dealer who understands how to do specialized merchandise is going to be a very profitable guy. As a matter of fact he is going to take over, if you want a better analogy than stereo, exactly the same place as the appliance sales store did. He is going to service his community.

Q: Do you see that analogy as going further, do you see this becoming an appliance that is going to be widely used?

A: That is our goal. We're doing everything we can to make it happen. In other words, the product's technical direction and marketing direction is to be a consumer item, with a secondary strong market emphasis on small business applications. But in addition to that, it is our intention to find ways to make this product quite useful to insurance salesmen, doctors, real estate salespeople – the classic professionals, people who have money, but more than that, people who are considered 'thought leaders' in their community. Suppose all the doctors and all the insurance salespeople put the PET in their offices and start to use it. Because they've used it in the office, they're familiar with the product, and now can use it in their home. Next Mr. John Q. Public, who buys things because doctors have them, will in fact start buying the product.

Q: What about the area of education?

A: We're using three approaches to the educational market this year. First, we're trying to support development of innovative programs for schools by educational 'thought leaders.' Second, anyone planning to use the PET in a course this fall will get a high priority allocation. Third, we're setting up a royalty program to pay royalties for software developed for the PET.

We also plan to demonstrate the PET at teacher education seminars and we're encouraging educators to write programs for both the home and school. Even though it's too late to sell to schools this year, we expect some action immediately.

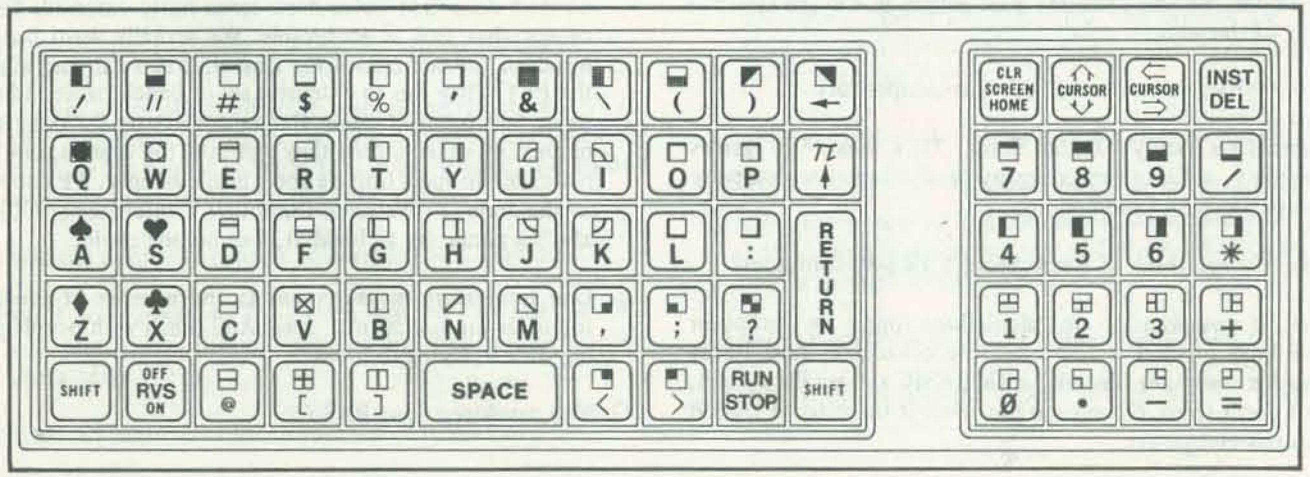
As professional educators start writing and trying out PET programs in schools we'll negotiate royalty agreements with the authors, then mass-merchandise the materials directly to the consumer.

Q: Can you give any price estimates on the educational packages you'll be selling?

A: The first year out, we are going to be able to get $20 for a decently packaged product. After that, I think the volume package is going to drop down to the $10 range.

Q: What can we say that is going to encourage people to write programs for the PET?

A: Suppose we had *really good* educational home programs right now, aimed at re-educating adults on how to compute interest rates, that kind of practical thing. I think such programs would sell rather quickly. But frankly, the aspect of education I consider most important, far broader than teaching children, is adult education. Everyone is missing



this fine market, educating the individual. We won't neglect that area: that is where the PET can be truly revolutionary.

Q: What sales volume do you expect to attain by Christmas?

A: By the Christmas period we will be manufacturing 5,000 a month in the United States; not long after that we'll be at the same production level in our Japanese plant.

Q: Are you doing all your manufacturing in the U.S.?

A: Nobody in the United States makes black and white CRTs anymore, you have to buy those things out of Taiwan. Cassettes are not made in Japan anymore, the only place to buy cassettes is out of Hong Kong and Korea. The semiconductors are, of course, made in this country, transported to Asia to be packaged, and transported back. The keyboards are Japanese, and will continue to be because the Japanese tend to do a very good job in that area. The housings, PC work, and assembly work will all be done in the United States for the U.S. market, in Europe for the European market, and in Japan for the Japanese and the Far East market.

Q: Are you going to make a very heavy marketing thrust in this country and then hit Europe or will both markets be tackled simultaneously?

A: You can't do it simultaneously. We have already opened Europe up with the Hanover Fair and we got very positive response. We will probably start selling in Europe after the first of the year.

Q: A year from now, who will be your competitors?

A: There'll be Tandy's Radio Shack, TI, a major U.S. games company, a European company, and a Japanese company competing with Commodore.

Q: What do you think of Radio Shack's TRS-80 computer?

A: Well, if you want to do calculations forget the computer and buy yourself a $30 scientific calculator. And Radio Shack's computer has only a 4K BASIC and no file system, so I don't think the system has what it takes to be classed as a real computer.

SOFTWARE

Q: What can you tell us about software distribution?

A: We are setting up either a division of the company or another company to publish software, most often on a royalty basis. We will only sell on a royalty basis software which has been *installed* and *tested* in a *financially rewarding environment.* In other words, the product has either got to have been installed in a scholastic situation or in some type of retail environment; the software publisher will test market programs strictly for the home. And we don't even talk to people until they've met those particular tests.

Q: What are the ranges on royalties for PET software?

A: We're going to try and make the successful author a good financial deal based on what the market for the particular product is. We have to set the prices of the cassette fairly low, because otherwise you get a high rate of theft. That means that the royalty percentage is fairly low, but hopefully it's compensated for by the fact that the volume is significantly higher.

Let's talk about the person who is writing programs for small businesses, particularly specialized applications. We'd like to see that author make a *better* than good living by writing three program packages a year.

Q: Can you tell us what software projects are currently in the works, either in house or out of house?

A: The answer is no, I can't. Not because I'm trying to avoid it, it's just that there are a whole series of software projects which people have started outside because of our official non-recognition. We do, by the way, offer an internal service where we will tell people that other people are working on similar products. Interested people should contact Terry Laudereau.

Q: What about software repair?

A: Which kind? If it is operating systems software, you buy what you buy. And there will be no warranty: the unit does what it does. But we've done some fairly extensive testing to see that that is acceptable. We actually went out and hired an outside consulting firm who came in and audited the PET. They ran an extensive set of benchmarks and tried to in fact break it using standard software breaking techniques. In other words they got into the system, saw that it worked to spec, outside spec, inside of spec. We also went to the June '77 National Computer Conference (NCC) and said 'all come, try to break it,' but no one could.

One interesting measurement is the number of problems found in the last month: zero. And that's with people playing with it, trying to break it.

Q: Who developed your BASIC?

A: Rick Wyland at MicroSoft wrote the PET BASIC. We climbed on the backs of all the MITS users who had problems, all of the 6800 users.

Q: Are you going to stick with BASIC as your primary language?

A: We did some things to BASIC in the I/O area to make BASIC a lot more flexible for growth. We designed the machine and software so the first-time user could walk up to the machine and operate it comfortably, and so the novice programmer could program. But we did the I/O structure for the sophisticated programmer. I think you'll find that with intelligent peripherals (which is the *only* kind Commodore will build), that you'll find that the structure is going to let you do a lot of I/O things.

We have a logical file structure operating system that really lets you operate at the logical file level; it lets you get in and play with that IEEE bus, right down to almost anything you want to do on it.

I want the business systems analyst who wants to write a good program using our floppy and our printer to be able to feel very comfortable sitting down and writing a well done program, where he doesn't feel that the language is inhibitive. And if I have done that, then I have met my goal.

Q: Are you going to come out with any materials supporting the machine language on the 6502?

A: We'll have an assembly language manual. There is a monitor which is currently not in the software; it will be available on tape.

Q: What about text editing?

A: We have a string editor and we have some editing capability already in it. Now the question is whether or not that's a legitimate text editor; I think more work needs to be done.

ADD-ONS

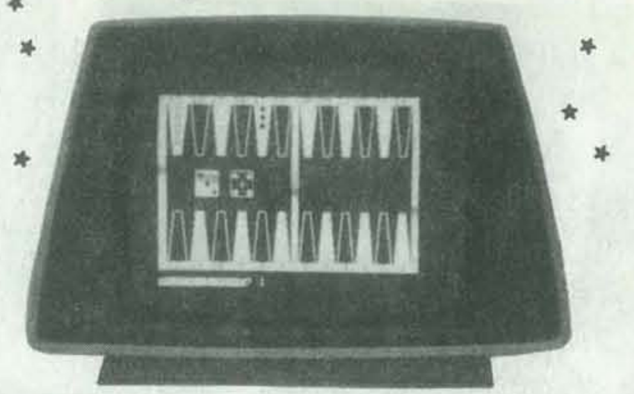
Q: Do you have any dates or costs of a printer for the PET?

A: We are negotiating with the printer manufacturer, we expect adapting our program to take a month or so. We hope to show the printer at WESCON, in September, and make it available to customers shortly thereafter.

Q: Will the printer be capable of handling PET graphics?

A: We'll do graphics to the extent that we can, but the CRT and printer use different sizes of dot matrices for printing characters.

Q: What are your plans for floppy discs for the PET?



A: We're negotiating with two mini-floppy companies right now, but few mini-floppy manufacturers can come close to meeting the demand we foresee. So we're going to wind up coming out with a traditional mini-floppy, priced high, for those people that need it, and then get to work developing a low cost mini.

Q: Do you have any plan for when you would like to bring out your low priced mini-floppy?

A: Yes, as soon as I can get it designed and into production. If we could come out with a really low cost product it would blow the mind of the world!

Q: While we're discussing peripherals, is there anyone that is making IEEE bus peripherals that are price compatible with the PET?

A: In terms of price compatibility, what do you want? If you're talking about a consumer application then at this point the answer is no. But we expect to see other companies producing price compatible PET peripherals.

Q: I have the impression that a color bit-map display is low on your list of priorities, compared to the mini-floppy and the printer. Is that right?

A: Yes; first we want to provide support for the small business. Second we want to give people traditional computing capability, such as disk sorts and the ability to print. In addition to the printer and mini-floppy a third peripheral will be something that is an interesting, fun type of thing. And I have my own choice for that which I think has higher priority than a color display.

Q: What is involved in adding more memory to the PET?

A: There's a box for memory expansion; it is brought out the side, and you plug in a connector. Commodore will sell the box. In that box you can put APL if we sell it, or any memory expansion that turns you on.

Q: What is involved in adding and replacing ROM's?

A: For a short time the ROM's are on sockets until we are satisfied with their performance because they are fairly new components. But after that, replacing the ROM means un-

soldering and soldering, you don't want to do that. That's why we brought the bus out: do your outboard stuff outboard. This is not a unit to be tinkered with. By the way, you'd blow the built-in diagnostics out of the water the minute you change ROM's.

Q: Is Commodore interested in facilitating use of the PET over personal computer networks?

A: Yes; we want to provide a straightforward way to provide communication facilities for the PET. And that doesn't mean using an RS232 interface. Eventually we'll come out with a box equipped with what's needed.

Q: What if I want to hook on my own keyboard to the PET?

A: That would be very difficult. You'd have to get into the guts of the software, and it was not intended that you be doing that. By making the PET a consumer item, by making it so any TV repairman can fix it, we really have cut down your ability to screw with it. I hate pushing my competition, but if you really feel like you have to get in and mess around with your computer, buy an Apple.

Q: What's the PET II going to be?

A: PET II will be a higher priced product, with dual cassettes built in and a fullsized keyboard; it will be aimed at the small business market in particular, and will be priced accordingly. By the way, one of the tests we're running is a keyboard laid out not like a typewriter keyboard at all. Literally we are going to lay the keyboard out abcdefg and see what happens.

Q: Why did you choose the IEEE 488 bus?

A: The PET is not a *product*, it's the beginning of a *sales opportunity*. In order to make that real, you need standardization: my electrical plug has got to plug into all electrical outlets. There were four alternatives: we could go IBM compatible bus, which was kind of ridiculous; we could go S-100 bus, which was ill conceived and is very expensive; we could create our own, but that meant getting everyone to agree with us to promote standardization; or we could choose the 488. We chose the 488 IEEE bus because it was the best alternative: it was developed by many good people who put a lot of time and effort into trying to define a plug compatible standard.

Q: Can the PET operate as a slave as well as a master?

A: No, it's a master only.

SERVICE

Q: Will service contracts be available?

A: Not from the factory, but yes they will definitely be available, just like you were buying a TV. We put a great amount of attention into that trying to make the PET easily serviceable.

Q: Do you think we're likely to see price competition between service contractors?

A: We've set the unit up so that it can be repaired by the individual repairman, so I think you are going to see the same factors evolve as in television service.

Q: Will the service manual be available to retail customers? If so, for how much?

A: Yes, but the price hasn't been set yet.

Q: What about board replacement or trade-in costs?

A: All of the traditional things that you would expect to be able to do relative to a consumer appliance will be available. But we haven't set prices yet, they really have to wait. Remember, we had to pick a price in order to get out in the market and start talking. We need to get a little more product history before we start setting such prices; but there will be board repair centers.

Q: If I go down to Mr. Calculator and buy a PET and three weeks later my kid puts gum in someplace or pours syrup down it, what happens?

A: Your retailer or his subcontractor will fix it. Everyone who retails our product must be capable of servicing it before he introduces it; that's one of the requirements. That's true for all retailers, large and small. A retailer cannot have the product unless they can service it.

Q: Who is going to train retail service personnel?

A: Commodore will initially train service people, then the retailers themselves will take it from there. We'll be conducting service schools in late August and early September.

Q: What about a PET user's group?

A: We will try to set that up probably using some of the Hewlett-Packard conventions. That seems to have worked out fairly well for them, but we foresee the local retailer playing a major role in interfacing between Commodore and consumers.

THE RETAILER

Q: Could you say some more about the role of the computer retailer?

A: Computer retailers are a sociological phenomenon more than anything else. They're the places where the consumer can go to tell of troubles and get advice. And of course, the social experience of trading information is fine for the retailer, who sells a $5, $10, or $15 item at every turn.

A large department store won't provide the facility for letting you try out programs before buying them or for test marketing your programs. That's where the retailer comes in.

I think the retail channel is going to be the commmunication vehicle for users. Your local merchant is the person closest to the problems of users, who is capable, has a reason to do business with the customer on an individual basis, and is in a position to recommend software for Commodore's royalty program.

A retail distribution channel is developing for all of the products that can be dreamed of by the next five years. And that retail distribution channel is not a flim-flam-thank-you-ma'am type of market. Automobile dealers are substantial members of their community. They are valued people, they make a lot of money, they belong to the appropriate country clubs, and they're valued members of the community. We would like to see computer dealers acquire equal social status.

PURCHASING A PET

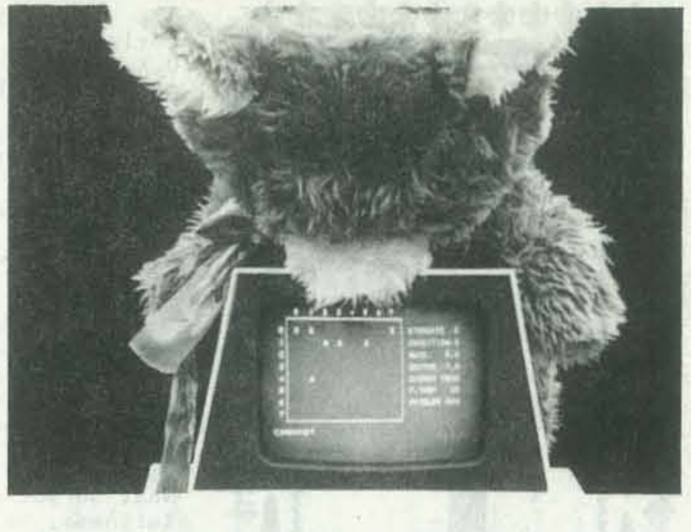
Q: What instructional materials will accompany a PET?

A: We've written an instruction manual; we also hope to provide an interactive instructional tape to teach thė consumer how to use the PET and to introduce programming in BASIC.

Q: Are educational discounts available?

A: Commodore has traditionally found a way to sell to the educational community at a price that is favorable to education. But we haven't set a program up yet, and probably won't get to it this school year.

Q: Does that mean that anyone in education wishing a system now should send in $595?



A: Yes, that's right – or $795; plus 6% sales tax in California. And quite frankly, if you're a dealer, that's the way you get PETs too. We're not discriminating in that respect, there are some dealers that want to get started on writing software or getting their act together and so forth, and they're paying, retail.

Q: What about discounts for quantity orders, say buying clubs or computer clubs or anything like that?

A: Commodore has been a traditional mail order house, and the mail order house will probably still sell mail orders, but aside from that, we sell to retailers. So if your local club can convince your local retailer to give them a discount, then you should do that. But you should understand the corollary to that. People want to become dealers in order to get a discount, and that just doesn't happen with us.

Q: Suppose a retail dealer wants to be a PET dealer. Today he would like a commitment from Commodore – 'In October I'd like to be a PET dealer.' What's he do?

A: For now, Sam Bernstein's handling it, but the only way we even talk to dealers right now is with cash deposits up front. We're not trying to be difficult, we're just saying that if you want to do business with us, you must have financial stability; to measure that, we want your money.

Q: Is the fastest way to get a system to send Commodore a check?

A: It's the only way, right now. However we encourage people to wait and do business with their local retailer. For those wishing to order a system now, orders will be filled in 90 days and are FOB Palo Alto, or money will be refunded. Checks should be made payable to Commodore and sent to:

Commodore Business Machines, Inc.
901 California Ave.
Palo Alto, CA 94304
(415) 326-4000

PILOT

PILOT is a language that's easy to learn and easy to use. *People's Computers* regularly features PILOT articles and programs. Here are 2 computer-assisted instruction (CAI) programs in PYLON, a first cousin of PILOT; they were written by Ellen Nold and Sallie Cannom of Stanford University for use in freshman English composition classes.

On this page is what happened when a student named Chris 'ran' a program named 'Mustard'. Each line that Chris typed is preceded by '>'.

```
 Hi.  What's your name?
>Chris
 O.K., Chris, let's talk about breaking out of the stereotypical
 perception that results in dull thinking and dull writing.

 Find pencil and paper and draw a hand.
 When you're finished, press the RETURN key.
>
 O.K.  Now you've drawn a hand.  How detailed is it?
>Not very
 Is it an evocative, thought-provoking representation?
>Hardly!
 Why?
>Too little detail, too little attention paid to its creation
 Most hands have four fingers and a thumb, so that's a
 basic representation - a kind of stereotype.  But past
 that, it's the differences rather than the similarities
 that make your picture interesting.

 Look at your own left hand as if for the first time.
 What do you see?
>Softness, hardness, hilss and valleys, wrinkles
 All right.  What else?
>colors --gold rosy flesh tones, the blue of veins
 And?
>nails -- long, short, chipped, many shpaes
 You could write a lot more just about your left hand -
 not counting inferences or speculations about the kind
 of person you are, or how your particular hand came to be.

 Here's a poem:

      HANDS

      Hands can work
      Hands can play
      Hands are used
      Every day.

 Not exactly spellbinding is it? But why?
>Very pedistrian information put into dull phrases
  Chris, using the details:
 Softness, hardness, hilss and valleys, wrinkles
 colors --gold rosy flesh tones, the blue of veins
 nails -- long, short, chipped, many shpaes,
 you could write a much more interesting poem about hands.
 Want to?
>OK
 O.K.  Here are five blank lines for you to use.

    HANDS

      by Chris

>Soft flesh climbs bony riges
>Rosy hued ridges, golden valleys
>Tipped by brittle jagged ovals
>The winding blue rivers
>Bring life to all

 Was that fun?  Maybe you'd like to do more with that poem later,
 or perhaps write another one.

 Practice in avoiding stereotypical thinking
 will increase your poetic perception and expression.

 To write perceptively means to see perceptively.

 For an exercise in seeing, try SAGE.
 Bye for now, Chris.
```

PROGRAMS BY ELLEN NOLD & SALLIE CANNOM

For more information on PILOT, see the PILOT interpreter in this issue, and articles in our back issues.

PILOT for the 8080 was developed by John Starkweather under contract to the National Library of Medicine. It is available at low cost electrically, on paper tape, and in several other media. For information, contact Bill Ford, Lister Hill Center for Biomedical Communication, National Library of Medicine, 8600 Rockville Pike, Bethesda, MD 20014.

Last but not least, there's the PILOT Information Exchange run by Gregory Yob, PO Box 354, Palo Alto, CA 94301. He maintains a library of materials in PILOT, source code in several languages, and a membership list with a brief note as to who is doing what.

◆ ◆ ◆

A summary of some PILOT commands:

R: Remark indicates that the text following on the same line is a comment about the program, and is not part of the program.

T: Type prints out on the terminal the text following on the same line.

A: Ask causes the computer to pause and wait for user input.

M: Match checks to see if what the user typed in for the last Ask command matches any of the items listed after the M: command. If there is a match, the match flag is turned on; if there is no match, the match flag is not turned on.

J: Jump causes a branch to the label named: that command is executed next.

Y: and N: Yes and No test the match flag. 'JY:' means 'jump if the match flag is yes, or turned on.' if the Y or N command occurs by itself, the 'T' command is used. 'N:' means 'type if the match flag is no, or turned off.'

E: End stops execution of that part of the program in which it occurs.

C: Compute permits evaluation of at least some arithmetic expressions.

U: Use calls subroutines.

MUSTARD

```
R: MUSTARD  by Ellen Nold and Sallie Cannom  Stanford  9/73
T: Hi.  What's your name?
*Name A:
T: O.K., @name@, let's talk about breaking out of the stereotypical
T: perception that results in dull thinking and dull writing.
T:
T: Find pencil and paper and draw a hand.
T: When you're finished, press the RETURN key.
A:
T: O.K.  Now you've drawn a hand.  How detailed is it?
A:
T: Is it an evocative, thought-provoking representation?
A:
T: Why?
A:
T: Most hands have four fingers and a thumb, so that's a
T: basic representation - a kind of stereotype.  But past
T: that, it's the differences rather than the similarities
T: that make your picture interesting.
T:
T: Look at your own left hand as if for the first time.
T: What do you see?
*See1 A:
T: All right.  What else?
*See2 A:
T: And?
*See3 A:
T: You could write a lot more just about your left hand -
T: not counting inferences or speculations about the kind
T: of person you are, or how your particular hand came to be.
T:
T:
T: Here's a poem:
T:
T:       HANDS
T:
T:       Hands can work
T:       Hands can play
T:       Hands are used
T:       Every day.
T:
T: Not exactly spellbinding is it? But why?
A:
T:  @name@, using the details:
T: @see1@
T: @see2@
T: @see3@,
T: you could write a much more interesting poem about hands.
T: Want to?
A:
M: |NO|UH U|LATER|UGH|
J Y: @qod
T N: O.K.  Here are five blank lines for you to use.
T:
T:     HANDS
T:
T:       by @name@
T:
A:
A:
A:
A:
A:
T:
T: Was that fun?  Maybe you'd like to do more with that poem later,
T: or perhaps write another one.
*Go T:
 T: Practice in avoiding stereotypical thinking
T: will increase your poetic perception and expression.
T:
T: To write perceptively means to see perceptively.
T:
T: For an exercise in seeing, try SAGE.
T: Bye for now, @name@.
T:
T:
END:
```

CURRY

```
R: Curry by Ellen Kold and Sallie Cannom  Stanford 2/74
T: Hi there!  What's your name?
*Name A:
T: O.K., @named@, this program is about inferences -
T: discovering how we make them and whether they're warranted.
T: If you'd like to learn about the differences between
T: reports, inferences and judgements, do Marjoram first.
T: Would you like to try Marjoram?
A:
M: |NO|UH U|LATER|
J N: @Z@
T: Good. Let's go on.
T: Here's a short tale.  On the basis of the words in the story,
T: you decide whether or not the statements which follow it are
T: true (T), false(F), or neither true nor false (N).
T:
*Go T:
T: Little Jack Horner sat in the corner
T: Eating a Christmas pie
T: He put in his thumb and pulled out a plum
T: And said, "What a good boy am I!"
T:
T: Jack had a thumb.
A:
M: |T|TRUE|
T Y: You're right!
T N: "Jack had a thumb" is a true statement.  The text said
T N: that he put in his thumb.
T:
T: Jack was sitting in a corner while he was eating.
A:
M: |T|TRUE|
T Y: Right, @named@.
T N: You think I'm trying to trick you, @named@?
T N: The statement above is true: the text says that
T N: Jack sat in a corner eating.
T:
T: Jack was eating a plum pie.
A:
M: |N|NEITHER|CAN'T TELL|
T Y: Right again, @named@.
T N: You can't tell from the text.  Jack may have found
T N: his plum in a raspberry pie.  "N" is correct.
T:
T: Jack was sitting on a chair.
A:
M: |N|NEITHER|CAN'T TELL|
T Y: Can't stump you on this one, can I?
T N: @named@, all the text says is that Jack sat in a corner.
T N: It doesn't say what he sat on.  "N" is correct.
T:
T: Jack was a good boy.
A:
M: |N|NEITHER|CAN'T TELL|
T N: Jack said he was a good boy, but his statement does not
T N: establish the truth or falsity of his "goodness."
T N: Again, "N" is correct.
T Y: Right.
T: If you'd like to do Jack Horner again, type OVER.
```

```
T: If you'd like to go on, type ON.
A:
M: |OVER|
J Y: @go@
T:
T: Why do you think we use fairy tales to test whether or not
T: you make hasty inferences?
A:
T: You're more likely to have set assumptions about
T: material with which you are familiar.
T:
T: Try this story:
*OH T:
T: Mary had a little lamb.  Its fleece was white as snow.
T: And everywhere that Mary went, the lamb was sure to go.
T: It followed her to school one day; that was against the rule.
T: It made the children laugh and play to see a lamb at school.
T:
T: Mary's lamb had white fleece.
A:
T: True.  The text says that the lamb's fleece was white.
T:
T: Mary went to school regularly.
A:
M: |N|NEITHER|CAN'T TELL|
T Y: Right, @named@!
T N: The text says that Mary went to school one day.
T N: We can't tell what she did the rest of the time.
T N: "N" is the answer.
T:
T: The lamb had a ribbon around its neck.
A:
M: |N|NEITHER|CAN'T TELL|
T Y: Correct.
T: The statement is neither true nor false.  We make a
T: fallacious inference to assume that because no ribbon
T: is mentioned it doesn't exist.
T:
T: Mary is a boy.
A:
M: |F|FALSE|
T Y: Right, @named@.
T: The text says "He followed her."  The feminine pronoun
T: signifies that Mary is a female.
T:
T: Mary is a woman.
A:
M: |N|NEITHER|CAN'T TELL|
T Y: Good answer.
T: T You can't tell.  Mary is feminine, but she is not called
T: by any referent which signals her age.  "N" is right.
T:
T: Maybe Mary liked to go to school.
A:
M: |T|TRUE|
T Y: You're right.
T: Any "maybe" statement not contradictory to the text
T: is a true statement.
T:
T: Is "Maybe Mary didn't like to go to school" a true statement?
A:
```

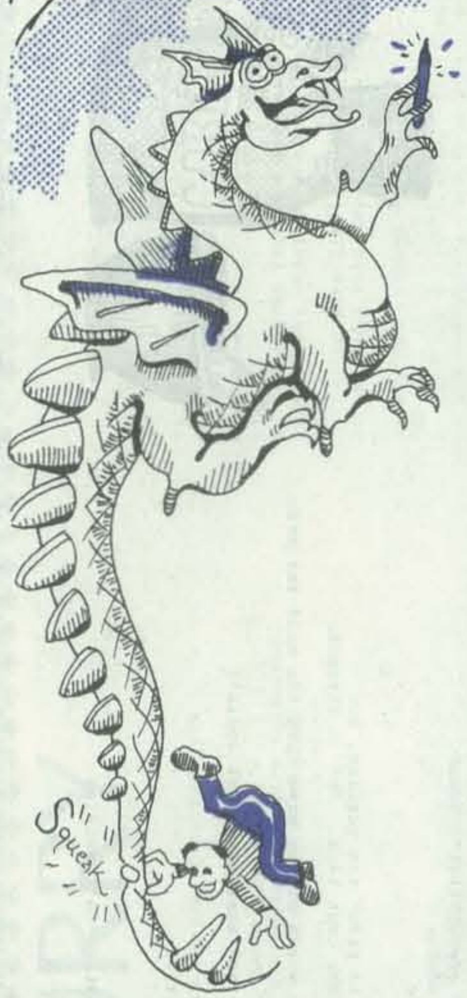
```
M: |Y|YES|
T Y: Yes is right, @named@.
T N: The statement is true because it doesn't contradict the text.
T:
T: Maybe Mary's lamb didn't have white fleece.
A:
M: |F|FALSE|
T Y: Right.
T N: The statement is false because it contradicts the text which says,
T N: "Its fleece was white as snow."
T:
T: If you had trouble with Mary, perhaps you'd like to go back over
T: the questions.  If so, type OVER.  If you'd like to go on to the final
T: text, type ON.  If you'd like to stop, type OUT.
A:
M: |OVER|
J Y: @oh@
M: |OUT|
J Y: @E1@
T: I'm glad you're going on, @named@.  Here's the last example:
T:
T: Little Red Riding Hood's grandmother was ill, and Red decided
T: to take her a basket of goodies.  The big bad wolf saw Red walking
T: through the woods on the way to her grandmother's house, and he
T: ran ahead to the grandmother's house and ate the grandmother.
T: Though Red didn't recognize the wolf when she first arrived
 T: at her grandmother's house, she screamed in time to be rescued
T: by a nearby woodsman.
T:
T: Red's mother fixed the basket of goodies for Red to take.
T:
M: |N|C NEITHER|CAN'T TELL|
T Y: Right.
T N: Can't tell from the text who did the fixing.  "N" is correct.
T:
T: Red recognized the wolf in time to be rescued.
A:
M: |N|NEIGHE   THER|CAN'T TELL|
T Y: Correct!
T N: The text says she screamed in time to be rescued.
T N: We don't know if she recognized the wolf or not.
T:
T: Red was taking goodies to her grandmother's house
T: when the wolf saw her in the woods.
A:
M:   |N|NEIG THER|CAN'T TELL|
T: You've got it, @named@.
T N: The text says she "decided to take" the goodies, but
T N: we don't know if, in fact, she took them.  "N" is correct.
T:
T: Maybe Red saw the wolf in the woods at the same time the wolf saw her.
A:
M: |T|TRUE|
T: True is right.  This is a "maybe" statement that doesn't
T: contradict the text.
T:
T: Red didn't like her grandmother.
A:
M: |N|C NEITHER|CAN'T TELL|
T Y: Right, @named@.
T N: "N" is right because the text doesn't mention
```

```
T N: anything about the subject.
T:
T: Red ran all the way from her house to her grandmother's house.
A:
M: |F|FALSE|
T Y: T Right again.
T N: The text says that the wolf saw her walking,
T N: so the statement is false.
T:
T: Maybe Red's grandmother wasn't really ill, but was
T: testing Red's affection for her.
A:
M: |F|FALSE|
T Y: Again, false is the correct answer.
T N: False is correct because the text says the grandmother is ill.
T:
T: Here's the last one.
T: Red was rescued by a passing woodsman.
A:
M: |N|NEITHER|CAN'T TELL|
T: Neither is right.  The text says the woodsman was nearby.
T: We don't know if he was passing nearby.
*E1 T:
T: What you've done in the past few minutes is to practice being
T: conscious of your own inferential leaps.
T: Your ability to reason accurately affects your skill as
T: as a reader and as a writer.
T:
T: If you want to work on slanted reports, try MACE.
T: If you're interested in questions that cannot be resolved,
T: try CHILI.
T:
*E T: Bye for now, @named@.
T:
T:
T:
End:
```

TINY LANGUAGES

SECTION 1: DRAGONSTUFF
BY BOB ALBRECHT

THE DRAGON



Once upon a time, when *People's Computers* was still a newspaper called *People's Computer Company*, we proposed a little language called Tiny BASIC. As many of you know, lots of people sent us letters, listings, paper tapes and other good stuff, and Tiny BASIC was born. Brothers and sisters of Tiny BASIC popped up everywhere. We now have Texas Tiny BASIC, Palo Alto Tiny BASIC, Itty Bitty Tiny BASIC, NIBL, Notso Tiny BASIC and . . . well, it boggles a dragon's mind!

More recently, *PC* (or whatever it's called now) brought you Tiny PILOT. Tiny PILOT is good for stuff that Tiny BASIC doesn't do well, and Tiny BASIC is good for stuff that is difficult to do in Tiny PILOT.

It's Tiny Language time again! This time, Dennis Allison and I and (we hope) lots of you people out there, want to design a language that

- Is good for Tiny BASIC type problems and is also good for Tiny PILOT type problems.
- Is designed to be most useful to elementary school kids (at home or at school) and also useful to teachers and parents of elementary school kids.
- Can be implemented in about 4K bytes of ROM with extensions possible in RAM.
- Runs on a personal computer whose advanced chip technology controls a color TV with simple graphics.

So, we're looking for

PARTICIPATION
COOPERATION
ENTHUSIASM . . .

YOUR ideas on how to design a language for kids. We especially want to hear from people who WATCH LOTS OF KIDS LEARNING HOW TO PROGRAM IN VARIOUS LANGUAGES (All kinds of kids – *not* just carefully selected 10 year old geniuses).

The project will generally follow the pattern which produced Tiny BASIC, but will be considerably enlarged, as follows.

STARDATE 1. In the next two or three or four issues of *People's Computers* we will design the language itself. On these pages I will propose a few ideas; on the next page, Dennis will share his wisdom.

STARDATE 2. Two things will happen.

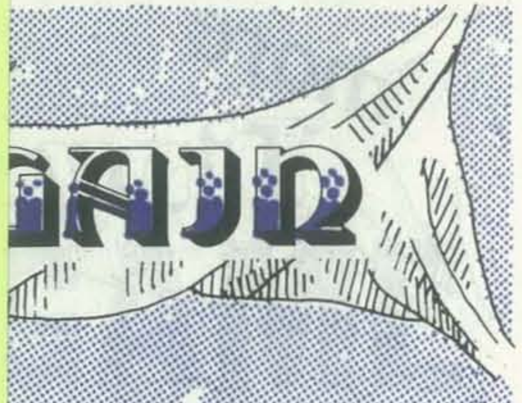
- People like me (authors of books) will write a 'teach yourself' style primer to help people learn the language.
- People like Dennis will design an Intermediate Language (IL) to interpret the Tiny Language.

STARDATE 3. Again, we hope, two things.

- People like me write a 'teach yourself' style primer to help people learn how to program in the IL.
- People like *you* out there will implement the IL and the Tiny Language on REAL COMPUTERS.

And now, here are a few Dragon Thoughts about a Tiny Language which for now I will call **DRAGONSQUEAK**. (I promise that I won't really call it DRAGONSQUEAK. I *was* going to call it PEOPLE'S LANGUAGE 1, or PL 1 for short, but someone told me there already is a

The Tiny Language extravaganza will be jointly presented by People's Computers *and* Calculators/Computer *(Dymax, Box 310, Menlo Park, CA 94025). We expect to obtain grants and hardware as prizes to further the project. For starters, our good friend Anonymous has donated $1000 to promote the development of Tiny Languages!*



pos-
ore
ON-

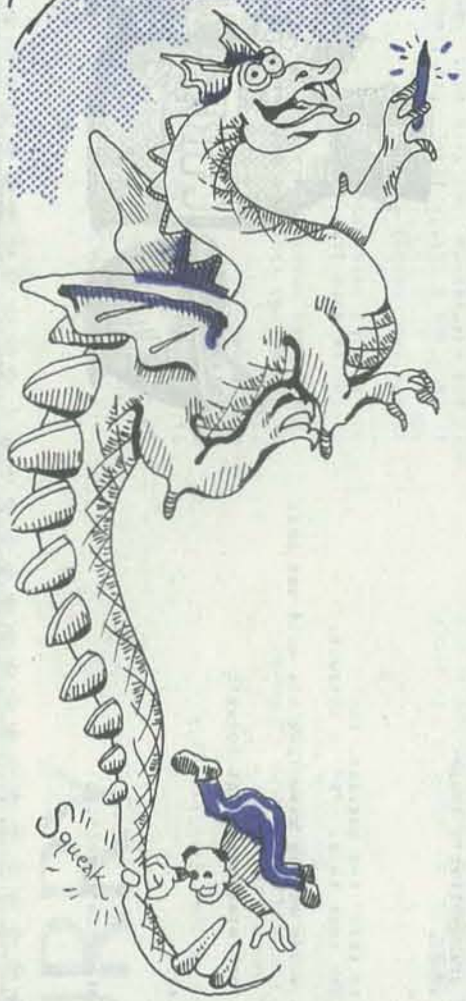
om-
om-

use

kid
and
e to

For Section 2
Heavy Stuff

by Dennis Allison
turn the page . . .



SECTION 1: DRAGONSTUFF
BY BOB ALBRECHT

THE DRAGON

Once upon a time, when *Pe*
Computers was still a newspaper
People's Computer Company, we
posed a little language called Tiny B
As many of you know, lots of peopl
us letters, listings, paper tapes and
good stuff, and Tiny BASIC was
Brothers and sisters of Tiny B
popped up everywhere. We now
Texas Tiny BASIC, Palo Alto
BASIC, Itty Bitty Tiny BASIC, N
Notso Tiny BASIC and . . . well, it
gles a dragon's mind!

More recently, *PC* (or whatever it's
now) brought you Tiny PILOT.
PILOT is good for stuff that Tiny B
doesn't do well, and Tiny BASIC is
for stuff that is difficult to do in
PILOT.

It's Tiny Language time again
time, Dennis Allison and I and (we
lots of you people out there, want
sign a language that

- Is good for Tiny BASIC type pro
and is also good for Tiny PILOT
problems.
- Is designed to be most useful
mentary school kids (at home
school) and also useful to teache
parents of elementary school kid
- Can be implemented in abou
bytes of ROM with extensions p
in RAM.
- Runs on a personal computer
advanced chip technology cont
color TV with simple graphics.

The Tiny Language extravaganza will be jointly presented by People's Computers *and* Calculators/Computer *(Dymax, Box 310, Menlo Park, CA 94025). We expect to obtain grants and hardware as prizes to further the project. For starters, our good friend Anonymous has donated $1000 to promote the development of Tiny Languages!*

Dr. Dobb's Journal Volume One

Now you can purchase a single bound volume containing all ten issues of the first year of **Dr. Dobb's Journal.** Complete contents are listed on the back of this issue. An order form is below.

Dr. Dobb's Journal of
COMPUTER
Calisthenics & Orthodontia
Running Light Without Overbyte

A Reference Journal
for Users of
Home Computers

Volume One

People's Computer Company

PCC's Reference Book

Here's just some of the useful information you'll find in **PCC's Reference Book**:

- ★ Hundreds of companies and computer stores selling hardware, software, peripherals and offering all sorts of services are listed with their addresses.
- ★ Nuts-and-bolts and survey articles on software, hardware, applications, robots, and the future, just to name a few, for the experienced and the not-so-experienced user of microcomputers.
- ★ The complete, documented source and object code for a 2K Tiny BASIC.
- ★ A massive index of the articles from the major hobbyist magazines, plus information on the magazines in the field, hobbyist clubs, newsletters and professional societies.
- ★ Bibliographies on different areas so you can investigate them further, including a special computer music bibliography.

⟶

An order form for these books is at right.
Or contact:

People's Computer Company
1263 El Camino Real
P. O. Box E
Menlo Park, CA 94025

people's computers Subscription Form

Please send me a one-year subscription to *People's Computers* magazine (published bi-monthly) for $8.

□ Payment enclosed □ This is a renewal (Please attach your label)

□ Bill me

NAME ______________________

ADDRESS ______________________

CITY/STATE ______________ ZIP ________ 34

(Foreign rates available on page 2)

1263 El Camino Real • Box E • Menlo Park, CA • 94025

people's computers Subscription Form

Please send me a one-year subscription to *People's Computers* magazine (published bi-monthly) for $8.

□ Payment enclosed □ This is a renewal (Please attach your label)

□ Bill me

NAME ______________________

ADDRESS ______________________

CITY/STATE ______________ ZIP ________ 34

(Foreign rates available on page 2)

1263 El Camino Real • Box E • Menlo Park, CA • 94025

Books

□ Please send me ____ copies of **Dr. Dobb's Journal, Volume One**, a complete, bound reprint of the original ten issues, for only $13.00 (California residents please add 78¢ sales tax) per copy.

□ Please send me ____ copies of **PCC's Reference Book**, for a mere $5.95 plus 95¢ for shipping and handling (California residents, please add 35¢ sales tax) per copy.

□ Payment enclosed □ Bill me

Name ______________________

Address ______________________

City/State ______________ Zip ________ 35

1263 El Camino Real • Box E • Menlo Park, CA 94025

pos-
ore
ON-

om-
om-

use

kid
and
e to

For Section 2
Heavy Stuff

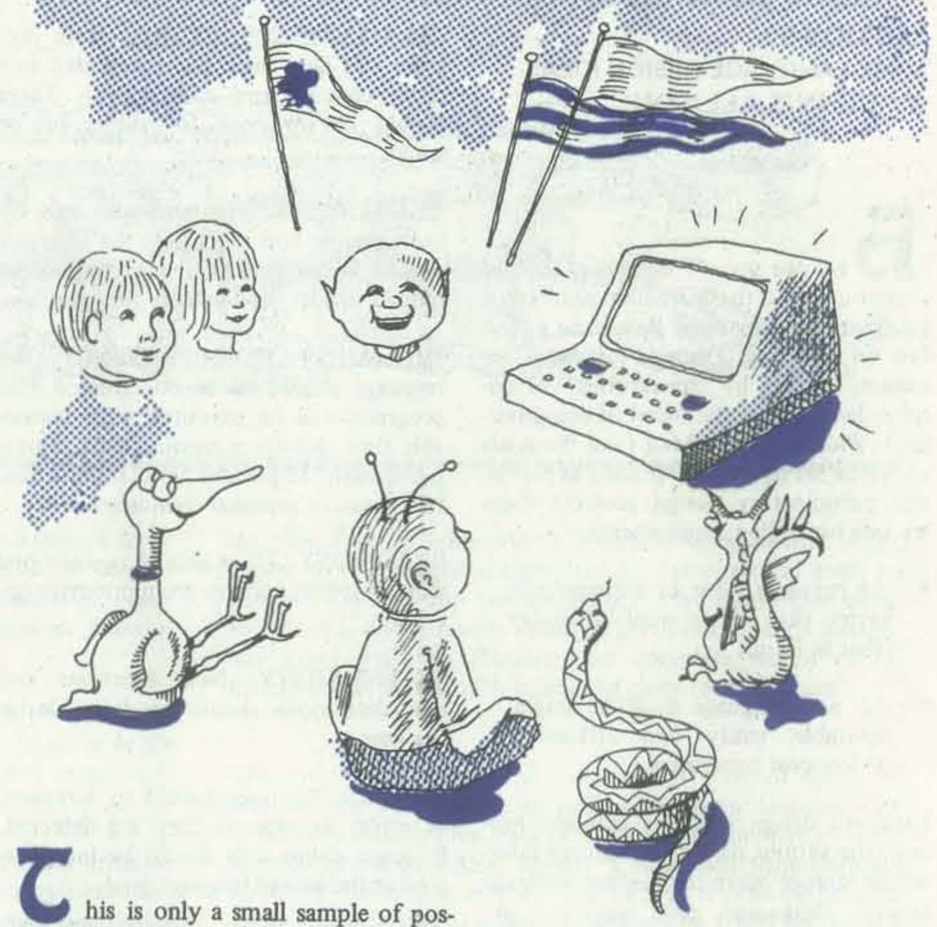
by Dennis Allison
turn the page . . .

The Tiny Language extravaganza will be jointly presented by People's Computers *and* Calculators/Computer *(Dymax, Box 310, Menlo Park, CA 94025). We expect to obtain grants and hardware as prizes to further the project. For starters, our good friend Anonymous has donated $1000 to promote the development of Tiny Languages!*

PL 1 . . . oh well . . . when you spend most of your time in a cave guarding treasure you get out of touch!)

- Superficially, **DRAGONSQUEAK** looks kinda like Tiny BASIC.
- However . . .
- *All* variables are *string* variables. There are no other kinds of variables.
- But you can still do arithmetic on strings that are numbers . . . **DRAGONSQUEAK** does ASCII arithmetic on *integers* (as many digits as you want).
- We also have one-dimensional string arrays.
- **DRAGONSQUEAK** has BASIC-like statements such as PRINT, INPUT, LET, GO TO, IF, etc. More about them next time.
- We have lots and lots and lots of interesting, useful, and powerful *functions*. For example,

RND(A,B)	Random interger from A to B
GCF(A,B)	Greatest common factor of A and B
SWAP(A,B)	Swap the values of A and B
SQZ(A)	Squeeze blanks (spaces) out of A
MATCH(A,B)	Moving windows match (I'll explain next issue).
MATCH(A,B,*)	Find out if A is a substring of B, using the single character (* is used as an example) to separate substrings inside B. Instead of * you may use any character, such as comma, colon, etc.
ELE(+,A)	Element of a set. Find out if the single first character (+ is used as an example) is an element of A.



This is only a small sample of possibilities. Next time I'll provide a more detailed naive description of **DRAGONSQUEAK.**

GOOD NEWS:
I am *not* a computer scientist or a computer language designer; I'm just a computer user.

BETTER NEWS:
I have watched thousands of kids use computers and continue to do so.

BEST NEWS:
The time *finally* has come! Now us kid watchers and teachers and parents and dragons are ready to design a language to meet *kid's needs*! Join us!

**For Section 2
Heavy Stuff**

**by Dennis Allison
turn the page . . .**

more Tiny Languages

SECTION 2: SOME LANGUAGE DESIGN IDEAS

BY DENNIS ALLISON

Here we go off designing another contribution to the incredible number of programming languages. Before we go too far, we had best examine just what we expect to gain by contributing yet another language to the babel of computerland. Whatever is decided (and the goals are to be set by you the readers as part of this participatory design project) there are two overriding requirements:

- the language must be substantially better than those available today (that is, Basic)
- the new language must be implementable easily and efficiently on low-cost computers.

Language design is a very complex process; the various parts of a language interact in strange ways to produce inconsistencies. Languages, good languages, are often personal products. Committees compromise too easily. Just as a camel may be described as a horse designed by committee, languages designed by committee have lumps and bumps in strange places. And they may not really satisfy their desired design goals.

One can wheel out the standard language design motherhood criteria. Trite as they may be, they do provide some level of guidance:

INFERABILITY. A language should be inferable in the sense that a knowledge of the various constructs should allow one to infer the form of other constructs.

CLARITY OF STRUCTURE. Both program and data must be represented in a clear, concise, and natural way. There should be no need for subterfuge or tricks.

READABILITY. Programs are read by both people and machines; the language should be so constructed as to aid the human reader understand the problem.

IMPLEMENTATION EFFICIENCY. The language should be so constructed that programs can be executed with reasonable time and space requirements. This is particularly important with today's limited resource personal computers.

PARSIMONY. There should be only one way of specifying any given primitive operation.

EXTENSIBILITY. New functions and new data types should be definable by the user.

ERRORS. The user should be informed of errors as soon as they are detected. Program debug aids should be incorporated at the source language level.

MODULARITY. The language should support the development of hierarchical structures of both program and data. Such structuring is important because it models the way people solve problems.

INTERACTIVE INTERFACE. . Interactive systems have substantially superior human factors when compared to traditional batch-oriented compile-load-run systems. Careful attention must be paid to the human interface.

In the limited view of the user, a programming language is just a tool, a convenient shorthand. In a larger sense it is a vehicle for problem solution. The ability to abstract concepts and manipulate them as objects is fundamental. It is here that the traditional programming languages are most cumbersome. Basic, because of its lack of any kind of procedure mechanism, is particularly poor. As of late, the trend in language design has been to provide more and better mechanisms for structuring. Smalltalk, the Xerox PARC Learning Research Group's language, provides a particularly nice mechanization of abstraction of both function and data.

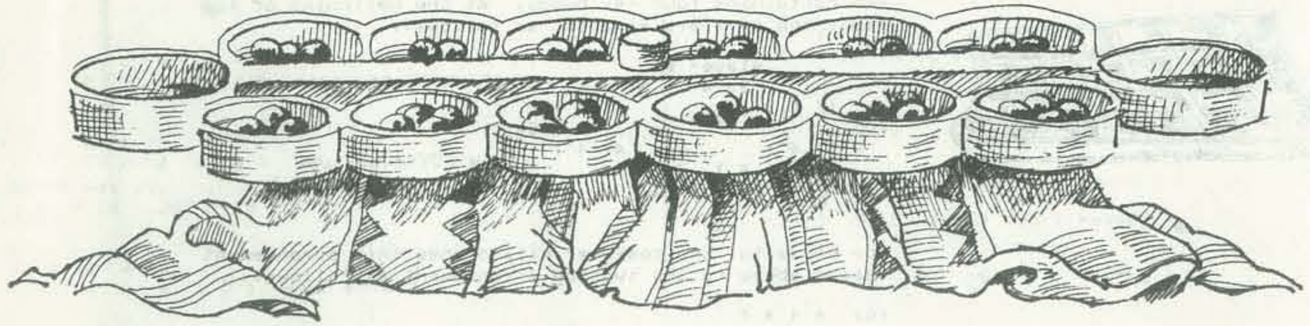
Just how one can utilize a language to aid problem solving is a topic of considerable interest. One would expect an effective programming language to reflect, in some way, the psychology of problem solution. Just how do we solve problems and what sort of mechanized tools make sense.

In future issues we will look at a variety of programming languages, and their features, discuss possible goals for a new language, and embark on the language design and its implementation.

But this is to be a collective effort: let us hear from you.

THE BEAD GAME

BY JON STEDMAN



An *awari* (or *oware*) board from Ghana.

As a gift to readers of People's Computers *Jon Stedman sends a listing of his BASIC program of mini-KALAH, a kind of Mancala game. The game involves players transferring beads from one cup to other cups, hence the name 'Mancala' from an Arabic word meaning 'transferring'.*

Jon wrote his program for homebrew computer users. He notes that it's easy for a human to beat the computer in his game, and that two human players make the game more interesting.

One of the ancient games that has survived in many areas of the world is known as Mankal'ah in Egypt, Pallanguli in India, Kalah in Syria, and as Wari in Africa. These games, generically known as 'MANCALA' games, have been played for thousands of years in Egypt where boards were found carved into the stones of the pyramid of Cheops. Perhaps it is the world's oldest game since it can be played using pebbles and holes in the ground.

The micro-processor is the latest toy medium produced by our technological society. The homebrew programmer has been developing games for this new toy at a prodigous rate. Star Trek is probably the best known of these space war computer games. Now the manufacturers are joining in with bells and whistles on their new T.V. electronic games. Where is this gaming craze leading us? My guess is the next step will be the competitive sport in computer games. Haven't we seen this competitive phenomenon happen to other toys such as 'pong'?

I can envision a situation where game contestants would match their pet computer strategies against one another in an arena controlled by a game-referee monitor. Tournaments would evolve so that programs submitted to a competition could be batched together and run automatically by the referee monitor. Round Robin results would be produced for each contestant along with his standing and complete record of his matches. Fun for all, with prizes and glory to the winner!

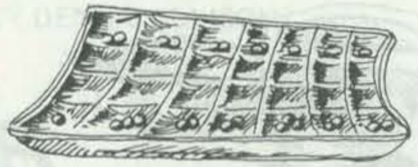
This visionary exercise was conducted at a recent computer seminar held at the Lawrence Hall of Science in Berkeley. Rob Shurtliff instructed his class of student programmers to write their programs in BASIC to play a simplified form of mini-KALAH. The enthusiasm was tremendous for the assignment. After a few weeks of intense programming effort by the students, a contest was held to determine whose program played the best game of mini-KALAH.

Some far-sighted manufacturer might see the fun and profit in this idea and produce a game referee monitor and sponsor such competitions. Future possibilities might include popular card games such as poker or bridge. Of course there are the more complex games such as go and chess that are currently challenging the theoretical scientist in artificial intelligence research. But, there is no reason not to challenge the amateur programmer with the next level of program abstraction. Current A.I. developments are essentially just heuristic searches over a very large choice-space.

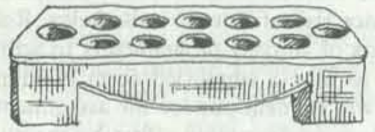
The following program is an example of mini-KALAH. I invite you to play this game on your homebrew computer and to develop your own program strategies. YOUR MOVE=?!

EXCERPTS FROM A 2–GAME MATCH WITH THE COMPUTER AS OPPONENT

```
kalah   21-May-77          10:33 AM
Game Instructions? yes
** ** The Bead Game of Kalah ** **
This game is played between two players.
The board is composed of ten -10- cups which contain beads.
Two of the ten cups are initially empty and represent the
score cups where the beads accumulate as each player scores
points.  Both players have four cups each with each
cup containing four -4- beads.  At the beginning of the
game, the cups are arranged as follows...

          player #2
           2( 4 )  3( 4 )
    1( 4 )            4( 4 )
(0)                        (0)
    1( 4 )            4( 4 )
        2( 4 )  3( 4 )
          player #1


For breveity, the computer will change this arrangement
viewed above to the two row arrangement as follows...

(0)  4 4 4 4
     4 4 4 4  (0)

The first row represents the score cup (0) and the four
cups  of your opponent.  The numbers specify the content
of each of his cups.  Likewise, the second row is your -4-
cups and a score cup.  Remember that both player's cups are numbered
(1 thru 4) from left to right.

The rules of the game are...

1) players alternate taking turns.
2) A turn consists of choosing a cup with at least one bead
   by entering the cup# and -return-.  The beads of this cup
   are distributed (one per cup) in a counter-clockwise
   direction.  Points are scored, when a bead (distributed in
   the above manner) falls into a score cup.
3) If a player has -no- beads remaining in any
   of his cups when their turn comes, then the game is
   over and all beads remaining in the opponents cups are
   added to the players score cup. (The computer does this)
4) Play continues until one player ends (as in rule 3).
5) A match consists of two games. Each player takes turn
   making the first move.  A game won scores one match point
   for the winning player.  A game tied scores 1/2 point
   for each player.  An even match will automatically start a
   re-match.

If you happen to make a wrong choice of cups and discover
your error before the -return- then just proceed to enter
a second digit and then -return-.  This illegal choice will
be rejected, allowing you to make a new choice.

---GOOD LUCK---

This game requires two players.
Would you like the computer to be your opponent? yes
What is your name? JON


( 0 )  4  4  4  4
       4  4  4  4  ( 0 )  computer move= 4


( 0 )  4  5  5  5
       4  4  4  0  ( 1 )  your move=? 1
```



1. A *mweso* board from Uganda.



2. An *awari* board from central West Africa.

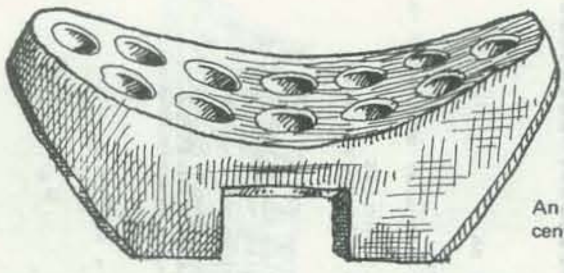
```
( 8 )  0  1  11  0
       0  1  2  2  ( 7 )  computer move= 3


( 8 )  0  1  11  0
       0  1  0  3  ( 8 )  your move=? 4
illegal move, choose again.
? 3


( 9 )  1  3  1  1
       1  2  1  4  ( 9 )  computer move= 4




( 17 ) 0  0  0  0
       1  0  0  1  ( 13 )  computer move= 4


( 17 ) 0  0  0  0
       1  0  0  0  ( 14 )  your move=YOU WIN!
The score was ...
computer = 14   JON = 18


( 0 )  4  4  4  4
       4  4  4  4  ( 0 )  your move=? 1


( 1 )  0  4  4  4
       5  5  5  4  ( 0 )  computer move= 4




( 16 ) 2  0  0  0
       0  0  1  0  ( 13 )  your move=? 1


( 17 ) 0  0  0  0
       1  0  1  0  ( 13 )  computer move= 3


( 17 ) 0  0  0  0
       1  0  0  1  ( 13 )  your move=YOU WIN!
The score was ...
computer = 13   JON = 19
Kalah Match Score...
computer = 0 JON = 2
Do you want a re-match? no
Thank you. Play again sometimes.  BYE!
```

A *gabatta* board from Ethiopia.

EXCERPTS FROM A 2-PERSON, 2-GAME MATCH

```
kalah   21-May-77          10:49 AM
Game Instructions? no
This game requires two players.
Would you like the computer to be your opponent? no
What is your name? JON
And the name of your opponent? PHYLLIS
O.K. JON and PHYLLIS, may the best person win!


( 0 )  4  4  4  4
       4  4  4  4  ( 0 ) JON's move=? 1


( 0 )  4  4  4  4
       0  5  5  5  ( 1 ) PHYLLIS's move=? 3


( 1 )  5  5  0  4
       1  5  5  5  ( 1 ) JON's move=? 1




( 17 ) 1  0  0  0
       0  0  0  0  ( 14 ) JON's move=PHYLLIS WINS!
The score was ...
 JON = 15   PHYLLIS = 17


( 0 )  4  4  4  4
       4  4  4  4  ( 0 ) PHYLLIS's move=? 1


( 1 )  0  4  4  4
       5  5  5  4  ( 0 ) JON's move=? 3




( 14 ) 1  0  0  0
       2  0  0  0  ( 15 ) JON's move=? 1


( 14 ) 1  0  0  0
       0  1  1  0  ( 15 ) PHYLLIS's move=? 1


( 15 ) 0  0  0  0
       0  1  1  0  ( 15 ) JON's move=? 3


( 15 ) 0  0  0  0
       0  1  0  1  ( 15 ) PHYLLIS's move=PHYLLIS WINS!
The score was ...
 JON = 15   PHYLLIS = 17
Kalah Match Score...
JON = 0  PHYLLIS = 2
Do you want a re-match? no
Thank you. Play again sometimes.  BYE!
```

mini-KALAH



An *awari* board from central West Africa.

In BASIC-PLUS, the $ symbol indicates a string variable and % an integer variable; variables not explicitly typed are assumed to be floating point variables.

```
 5 ! Program-id.         mini-KALAH
10 ! Author.             Jon D. Stedman
15 ! Installation.       UNIX  U.C. Berkeley
16 ! Language.           Basic-Plus
20 ! Date-written.       May 16, 1977.

50 dim k%(10), a$(5), p1$(9), p2$(9)
55 data 4,3,2,1,4,3,2,1,3,4,1,2,3,4,2,1,3,4,2,1,0

65 input "Game Instructions"; a$
75 if left(a$,1) = "y" then gosub 610
80 print "This game requires two players."
85 input "Would you like the computer to be your opponent"; a$
95 c%=0: if left(a$,1) = "y" then c% = 1
96 input "What is your name";p1$
97 if c%=1 then 100
98 input "And the name of your opponent";p2$
99 print "O.K. ";p1$;" and ";p2$;", may the best person win!"

100 !   play match of kalah
101 m1=0: m2=0
105 gosub 390   !initial board
106 let t%=1: w%=0
110 gosub 145    ! play game 1
115 gosub 390    ! initial board
120 let t%=2: w%=0
125 gosub 145    ! play game 2
135 if m1 = m2 and m1 < 2.0 then 105
136 print "Kalah Match Score..."
137 if c% =1 then print "computer =";m1;p1$;" =";m2
138 if c% =0 then print p1$;" =";m1;" ";p2$;" =";m2
139 input "Do you want a re-match";a$
140 if left(a$,1) = "y" then 100
141 print "Thank you. Play again sometimes.  BYE!": stop

145 !  play game of kalah
146 restore
150 gosub 360    ! print current board status
155 gosub 200    ! get player move

160 if w%<>0 then 185 ! check for winner
175 gosub 420    ! make player's move
176 t% = t% + 1: if t% > 2 then t% = 1
180 goto 150      ! loop
185 gosub 465    ! compute score and print
190 return


200 ! get next player's choice
201 x% = k%(1)+k%(2)+k%(3)+k%(4): y% = k%(6)+k%(7)+k%(8)+k%(9)

202 if x% = 0 and t%=1 then w% = 1
203 if y% = 0 and t%=2 then w% = 2
210 if w%>0 then 355
```

```
280 if c%=1 and t%=1 then gosub 570 else input n%
305 let n% = int(abs(n%))
310 if n% >= 1 then if n% <= 4 then 330
320 print "illegal move, choose again."
325 goto 280
330 if t% = 2 then n% = 10 - n%
331 if k%(n%) = 0 then 320
355 return

360 !  print current board status
361 print: print
366 print "(";k%(10);")";tab(6);k%(9);k%(8);k%(7);k%(6)
367 print tab(6);k%(1);k%(2);k%(3);k%(4);" (";k%(5);") ";
371 if c%=1 then if t%=1 then print " computer move="; else print " your move=
374 if c%=0 then if t%=1 then print p1$;"'s move="; else print p2$;"'s move=";
385 return

390 !    initial board array
395 for n%=1 to 10
400       let k%(n%)=4
405 next n%
410 let k%(5)=0: k%(10)=0
415 return

420 !  make player's move
425 let x%=k%(n%): k%(n%)=0
430 for a%=1 to x%
435       let y%=n%+a%
440       if y%>20 then let y%=y%-20
445       if y%>10 then let y%=y%-10
450       let k%(y%)=k%(y%)+1
455 next a%
460 return

465 ! compute winner score and print
470 if w%=1 and x%=0 then k%(5)=k%(5)+y%
480 if w%=2 and y%=0 then k%(10)=k%(10)+x%
525 if k%(5)<>k%(10) then 541
530 print " DRAW GAME!"
535 let m1=m1+0.5: m2=m2+0.5
540 goto 560
541 if k%(5) > k%(10) then w%=1 else w%=2
545 if w% = 1 then let m1 = m1 + 1  else m2 = m2 + 1
555 if c%=1 and w%=1 then print "YOU LOSE!"
557 if c%=1 and w%=2 then print "YOU WIN!"
558 if c%=0 then if w%=1 then print p1$;" WINS!" else print p2$;" WINS!"

560 print "The score was ..."
562 if c% =1 then print "computer =";k%(5);"  ";p1$;" =";k%(10)
563 if c% =0 then print " ";p1$;" =";k%(5);"  ";p2$;" =";k%(10)
565 return
570 !  computer strategy
571 s% = k%(5)+k%(10)
574 if s%>15 then 580
```

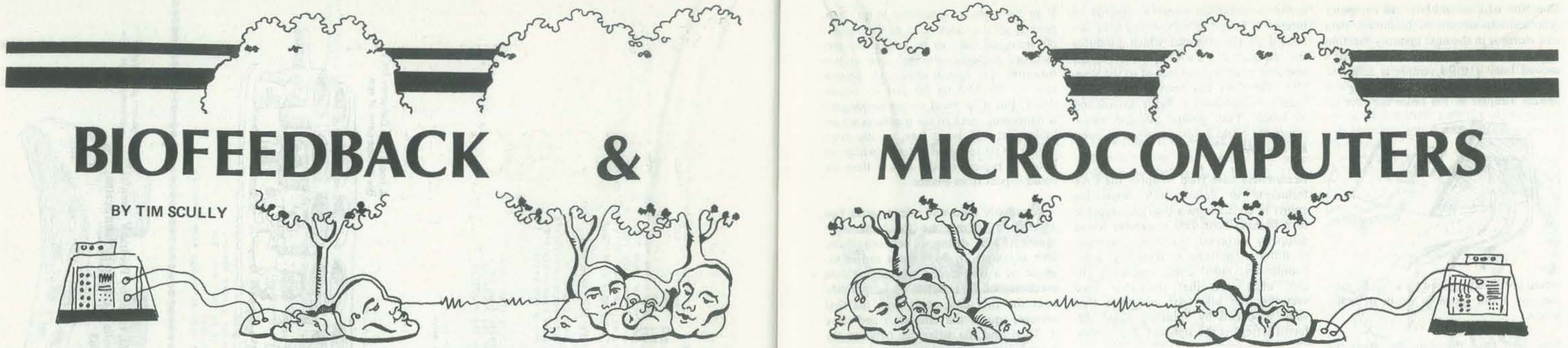
```
575 read n%
576 if n%=0 then gosub 996
577 if k%(n%)=0 then 575
578 print n%
579 return

580 if s% < 20 then 584
581 gosub 900
582 if n% = 0 then gosub 935
583 if n% <> 0 then 585
584 gosub 870
585 print n%
586 return

610 ! instructions for game of kalah
615 print  "** ** The Bead Game of Kalah ** **"
620 Print "This game is played between two players."
625 print "The board is composed of ten -10- cups which contain beads.
630 print "Two of the ten cups are initially empty and represent the"
635 print "score cups where the beads accumulate as each player scores "
640 print "points.  Both players have four cups each with each"
645 print "cup containing four -4- beads.  At the beginning of the"
650 print "game, the cups are arranged as follows...": print

655 print "        player #2"
660 print "         2( 4 )  3( 4 )"
665 print "     1( 4 )            4( 4 )"
670 print "(0)                         (0)"
675 print "     1( 4 )            4( 4 )"
680 print "         2( 4 )  3( 4 )"
685 print "        player #1"
690 print: print
695 print "For breveity, the computer will change this arrangement"
700 print "viewed above to the two row arrangement as follows..."
710 print: print "(0)  4 4 4 4"
711 print         "     4 4 4 4  (0)"
712 print: print
715 print "The first row represents the score cup (0) and the four"
720 print "cups  of your opponent.  The numbers specify the content"
726 print "of each of his cups.  Likewise, the second row is your -4-"
728 print "cups and a score cup.  Remember that both player's cups are numbered"
730 print "(1 thru 4) from left to right."

740 print: print "The rules of the game are..." : print
745 print "1) players alternate taking turns."
750 print "2) A turn consists of choosing a cup with at least one bead"
755 print "   by entering the cup# and -return-.  The beads of this cup"
760 print "   are distributed (one per cup) in a counter-clockwise"
765 print "   direction.  Points are scored, when a bead (distributed in"
770 print "   the above manner) falls into a score cup."
775 print "3) If a player has -no- beads remaining in any"
780 print "   of his cups when their turn comes, then the game is "
785 print "   over and all beads remaining in the opponents cups are"
790 print "   added to the players score cup. (The computer does this)"
800 print "4) Play continues until one player ends (as in rule 3)."
810 print "5) A match consists of two games. Each player takes turn "
815 print "   making the first move.  A game won scores one match point"
820 print "   for the winning player.  A game tied scores 1/2 point"
825 print "   for each player.  An even match will automatically start a "
826 print "   re-match." : print
830 print "If you happen to make a wrong choice of cups and discover"
835 print "your error before the -return- then just proceed to enter"
840 print "a second digit and then -return-.  This illegal choice will"
845 print "be rejected, allowing you to make a new choice."
850 print: print "---GOOD LUCK---": print
855 return

870 ! get cup with highest count
875 x% = k%(1): n% = 1
```

```
880 if x%<=k%(2) then let x% = k%(2): n% = 2
885 if x%<=k%(3) then let x% = k%(3): n% = 3
890 if x%<=k%(4) then let x% = k%(4): n% = 4
895 return

900 ! set cup to pass one
905 let n% = 0
910 if k%(4) > 1 then n%=4
915 if k%(3) > 2 then n%=3
920 if k%(2) > 3 then n%=2
925 if k%(1) > 4 then n%=1
930 return

935 ! set cup to set up pass next turn
940 n%=0
945 if k%(1)<>0 then if k%(2)>2 then n%=1
950 if n%<>0 then 995
955 if k%(3)<2 then 975
960 if k%(1)>1 then n%=1
965 if k%(2)>0 then n%=2
970 if n%<>0 then 995
975 if k%(4)=0 then 995
980 if k%(1)>2 then n%=1
985 if n%=0 then if k%(2)>1 then n%=2
990 if n%=0 then if k%(3)>0 then n%=3
995 return

996 restore
998 read n%
1000 return
```

A board from Dahomey for the Yoruba game of *ayo*.

REFERENCES:

Games of the World, edited by Fredric V. Grenfeld
Scarne's Encyclopedia of Games, by John Scarne
Time, June 14, 1963



BIOFEEDBACK & MICROCOMPUTERS

BY TIM SCULLY

PART I

Tim Scully has been designing biofeedback equipment and doing biofeedback research for many years. He is now working on a doctorate in psychology; his dissertation project involves researching and developing biofeedback systems and techniques for use in drug rehabilitation.

Tim is also teaching a computer class to fellow inmates at a Federal penitentiary. Although prison resources are scarce and he is not allowed to solicit donations, he is hopeful of somehow eventually acquiring a computer system for the prison.

Tim has received permission from prison officials to work on a project to modify a computer system which will allow a cerebral palsy patient to communicate. He plans to make his work available to the public to allow others to duplicate his efforts.

What does your computer have to do with your feelings? Will your computer ever be able to understand what you are thinking? Research in biofeedback and physiological monitoring with microcomputers may soon provide answers to such questions.

Biofeedback. That's learning to control *bio*logical processes by *feed*ing information about the process, from a measuring instrument, *back* to the student. Physiological monitoring is simply measuring biological processes and perhaps recording and analyzing them.

Biofeedback has become a rewarding area for research, an accepted part of medical practice and a popular educational/entertainment process. There are some good reasons for this: it turns out to be possible to learn at least some voluntary control of any body process that can be measured. The learning process can be as brief as a dozen hours for many body processes, and most interesting of all, control of your body's process brings with it some control over your own consciousness (mood and cognitive mode).

The biological processes most commonly measured in biofeedback training are skin temperature, muscle tension, skin resistance and brainwaves. Each is measured by a specialized instrument and each varies with changes in consciousness. Skin temperature at the fingertips, for example, is for many people a sensitive indicator of mood and stress. If you are feeling safe, relaxed and happy, your fingertips are probably warm. But if you are feeling uncomfortable and stressed, your body may respond as though in a life or death situation: with the fight or flight response. Part of this response is the withdrawal of blood from the hands and feet so that the vital muscles and organs are prepared for running or fighting. The result of reduced blood circulation in your hands is that they get cold. This process can be triggered by worry over giving a speech or by being caught in rush hour traffic. Biofeedback training in handwarming is done to teach relaxation and stress management skills.

We are all aware of using our muscles to move ourselves around and to move objects. We are less aware of the complex patterns of muscle tension which shift with our changing moods. These patterns are superimposed on the gross muscle action necessary for movements. When we are under stress, for example, we use bands of tense 'muscle armor' to protect ourselves from real or imagined threats. A sensitive instrument, the electromyograph (EMG), can pick up the electrical signals of muscle action. The EMG is used in biofeedback training in muscular control. Such training may be simple deep relaxation training, or it may be more complex retraining of muscles damaged by trauma, stroke or cerebral palsey.

The electrical resistance of our skin also changes with mood. You've probably noticed that sweaty hands can be an indication of stress or arousal. Sweat contains salt and conducts electricity well. We are less conscious of more subtle changes that take place in the permeability of cell membranes in our skin which produce smaller but rapid changes in skin resistance with variations in mood and arousal. Such a change in skin resistance is sometimes called a galvanic skin response (GSR) and GSR is one of the principal measures in the traditional lie detector.

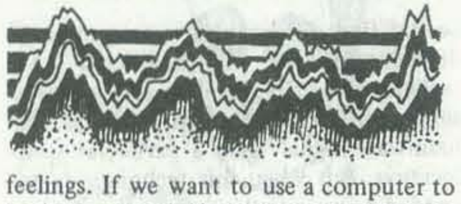
The electrical activity of the brain leaks out onto the scalp and the signals which can be measured there are sometimes called brainwaves (more properly EEG or electroencephalograph signals). You've probably heard of alpha waves. Alpha is the brainwave frequency range from 8 to 13 Hz and may appear during eyes-closed relaxation or meditation. Brainwave signals are complex, constantly changing and varying depending on which part of the scalp they are picked up from.

Trying to measure events in consciousness by looking at the electrical signals leaking through the scalp is a little like trying to find out what is happening inside your computer by analyzing the signals that leak out through its cabinet. It is possible, but involves sorting out signals of interest from many unrelated signals which can be considered to be 'noise'.

This same problem of 'signal to noise ratio' exists for any effort to relate any other physiological measure such as muscle tension or skin resistance to events in consciousness. Although the events in consciousness in which we may be interested do influence each physiological measure, other influences also modify each physiological measure. This means that we are faced not only with the problem of decoding the body's language, but also with the problem of sorting out the language from the background noise. This is a little like trying to learn a new language at a cocktail party.

COMPUTER ANALYSIS

Modern information theory and computers can come to our rescue. A computer can be used to look for the signals of interest buried in background noise and can be used to look for physiological response patterns typical of different thoughts or feelings. If we want to use a computer to analyze a signal, the first step is to get information about the signal into the computer. Brainwaves are wave-like; they are tiny voltages that vary from moment to moment. Just as water waves on the ocean average out to sea level, brainwave voltages (as they are usually measured) average out to zero volts in the long run. It is the tiny variations, first positive and then negative, in voltage that we want to analyze.

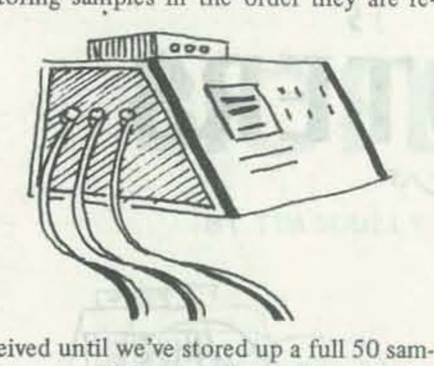


An A/D converter, or analog to digital converter, is used to sample the brainwave signal at regular intervals. Each time it samples the brainwave signal it measures its voltage and converts that measurement into digital form – a form which the computer can accept. The A/D converter might sample 50 times each second, thus providing the computer with 50 numbers each second. Each number is an instantaneous voltage level of the brainwave signal. If we've chosen a fast enough sampling rate, we could ask the computer to use these numbers to draw a graph of the brainwave signal, and the graph would look very much like the original brainwave signal. That's a good test to see if we are giving the computer enough information about the brainwave signal.

Let's consider a classical example of computer analysis of brainwaves. One of the major contributing factors in our brainwave production is the electrical activity of the brain resulting from the processing of sensory data. A computer can be used to pick out the part of the brainwave signal that is contributed by our response to a particular sensory stimulus, such as a flashing light. This response will be buried in the total brainwave signal made up of many contributions.

Suppose we flash a light in a person's eyes once each second and at the same time measure and digitize brainwave data from her and send this data to our computer. The experiment is easiest to run if we let the computer control when the light flashes and when the A/D converter samples. The light is flashed and at the same instant the first sample of brainwave data is digitized and stored in the computer's memory, at the beginning of a block of memory set aside for this experiment.

One 50th of a second later the computer gets another sample of brainwave data and stores it in the next memory location in the block we'd set aside. One 50th of a second later a third sample is collected and stored, and we continue sampling and storing samples in the order they are received until we've stored up a full 50 samples, or a full second's worth of brainwaves.

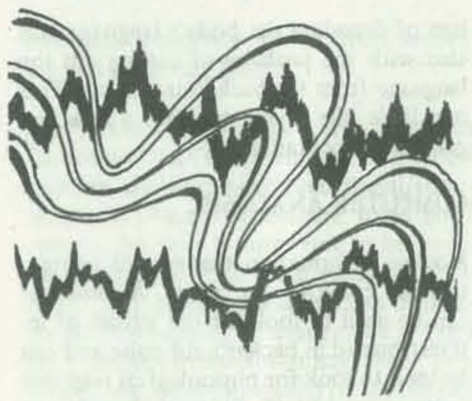


Now let's have the computer flash the light again and begin collecting a new batch of samples of the brainwave signal. The first sample from the new batch should be added to the first number stored in our block of memory. The second sample gets added to the second number stored in memory and so on. At the end of two seconds we will have collected two complete sets of 50 brainwave samples and we'll have added them together. If we were to have the computer stop collecting new data at this point and go through those 50 memory locations and divide all the numbers found there by two and then print out the result as a graph, we'd get a picture of the average of the two one-second segments of brainwave data.

In the usual experiment of this type we wouldn't stop with only two flashes of the light. We'd go on to average together as many as 40 or 50 flashes and their brainwave responses. The result of this averaging process is that the brainwaves that are unrelated to the flashing light tend to average out toward zero, while any brainwave response which is caused by the flashing light will tend to be reinforced by each additional flash that we add to the average. We've found a way to extract the signal from the background noise.

The technique we've just described is called visual evoked response averaging. A special purpose computer called a 'computer of averaged transients' (CAT) is often used for this type of work. A CAT is useful in situations where a response of interest is buried in background noise unrelated to the stimulus which produces the response. For a CAT to be useful, the response must be time locked to the stimulus. The CAT has been used to study brainwave responses to lights, sounds and to touch. Your general purpose microcomputer could be programmed to operate as a CAT.

Researchers have tried to apply the CAT technique to detect more interesting events in consciousness than our response to a flashing light. One researcher found distinctly different brainwave responses to different patterns of lines (*e.g.* polar coordinates, radial lines, concentric circles, etc.).[1] Another researcher tried averaging the brainwave responses that were produced by speaking aloud different letters of the alphabet many times; he did find different patterns typical of different letters.[2] The traditional CAT technique usually involves looking at a single physiological event, such as the brainwave signal from a particular scalp location. But when this technique is expanded to include the recording and separate averaging of many different physiological signals, some very interesting possibilities develop.



RESPONSE PATTERNS

As we've seen, the power of the CAT lies in its ability to pick a weak response out of strong background noise by averaging many responses. Suppose we do this for many physiological measures; the result is a set of response patterns, one for each measure. Now we can ask our computer to compare a single response, in real time, for each physiological measure with its corresponding set of stored average response patterns.

If an event we are interested in isn't happening, we'd expect most of the current physiological data to be unlike our prerecorded patterns, although one or two measures may appear similar to the pattern we are looking for just by coincidence. But if an event we are looking for *is* happening, each of the measures we are looking at will be at least a little more similar to its recorded pattern (which we know is typical of the event) than we could expect from chance.

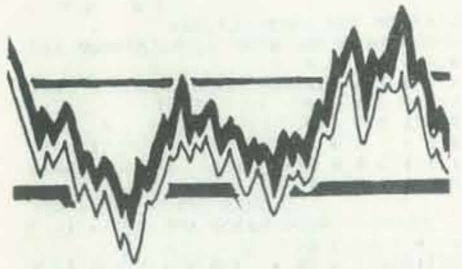
A commonly used way of comparing two patterns is to measure the 'root mean square (RMS) distance' between them. This is simple to do. In our earlier example of a one second segment of data consisting of 50 samples, we simply subtract the first data point from the first average, square the difference and store it. Then find the difference between the second data point and the second average point, square that and add it to the last square. If we keep this up for all 50 points, we'll end up with the sum of the squared differences between the individual points. Then we divide by the number of points (in this case 50) and take the square root of the result. That's the RMS distance; it'll be small if the patterns are similar.

If we measure a large number of different physiological events at once (let's say that we measure brainwaves from many different scalp locations) then we can sum the RMS distances from each scalp location's prerecorded average and get the same kind of improvement in signal to noise ratio that we got from averaging many repetitions of the event. We can use a kind of 'spatial averaging' in place of averaging over time.

SPATIAL AVERAGING

This idea of spatial averaging has been tried out. In a series of experiments at Stanford Research Institute,[2] a computer was taught to identify the word an experimental subject was thinking. Before you get excited about 'mind reading' by your computer though, you should know the limitations of this experiment.

The first, and perhaps most important, limitation was that the word had to be thought on command from the computer. This is because of the precise timing required for comparing the stored patterns with the sample brainwaves; small timing errors reduce accuracy considerably. The next limitation is the size of the computer's vocabulary; in the first series of experiments it was the words put, schoolboy, coughdrop, tip and had. These words were recognized correctly by the computer about 2/3 of the time by the comparison of EEG signals from 4 scalp locations with stored patterns for each word.



Another big limitation is the speed of identification of words. In the early SRI experiments the major part of the computing was done off-line and the words were not identified in real time. In later experiments real time identification was achieved for a vocabulary consisting of the words right, left, up, down, near, far and stop, but the accuracy of identification dropped to 55%, even though 5 channels of EEG were analyzed.

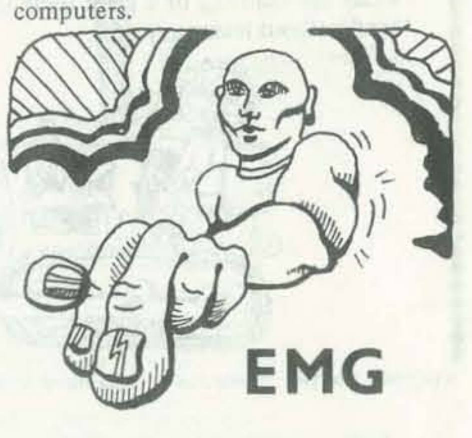
Meanwhile, the idea of spatial averaging is being tried out in other applications. One very interesting project involves computer-aided instruction (CAI).

Pierre St Jean is an instructor in the Social Sciences Department at Algonquin College in Ottawa, Ontario. He is also researching methods for making CAI more effective through computerized physiological monitoring. He's using a Nova minicomputer to present CAI material on a CRT terminal in printed form. From time to time the program asks the student a question. If the answer the student types back on the terminal .is correct, more new material is presented. If the answer is wrong, an alternative and more detailed presentation of the material is made and then more questions are asked.

That's traditional CAI. Pierre has added a physiological monitoring system which allows the computer to receive and analyze physiological data from its student while the CAI process is happening. Pierre's system presently monitors 4 channels of brainwaves, and one channel each of skin temperature, EMG (muscle tension), GSR/BSR (skin resistance) and EOG (electrooculograph, for measuring eye movements).

Pierre is in the process of finishing up software which will enable the NOVA to handle all this physiological data while it is supervising CAI. The idea behind this is to look for patterns of physiological response (which may be different for different students) typical of good or poor performance at the CAI task. Some day it may be possible for the computer to distinguish between wrong answers given because of lack of understanding and those resulting from poor attention. Eventually we may be able to recognize the physiological signals that correlate with times when a student can learn well from reading, other signals may indicate that a film would be the best way to learn, others might lead the computer to suggest to the student that it is time for a 10 minute exercise break or a brief meditation period.

We don't know yet if patterns like these can be reliably detected, but the existing research data are promising. The strategy for studying these patterns of physiological response will involve extensive use of computers.



The hardware which Pierre St Jean is using was built for him by Aquarius Electronics in 1974, with some recent additions. Experience from the design of this, and other, computerized physiological monitoring and biofeedback systems has led to the design of an S-100 compatible set of plug-in printed circuit modules for interfacing physiological data to microcomputers.

Part II of this series will describe more recent research using microcomputers to look for physiological correlates of emotional states. We'll also talk about other approaches to relating physiological and psychological events, primarily biofeedback. The essential idea behind biofeedback training is the use of a sophisticated system of instruments to improve the sensitivity and range of our internal perception abilities temporarily so that we can learn more about ourselves and about how to 'shift gears' mentally and physiologically from one state of consciousness (mood and mode of cognition) to another.

FOOTNOTES

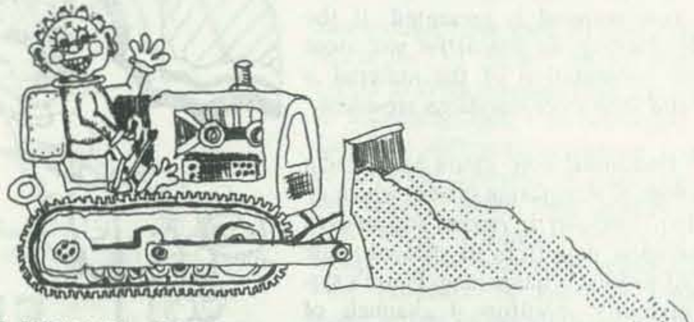
[1] Clynes, Manfred 'Sentics: Biocybernetics of Emotion Communication,' *Annals of the New York Academy of Sciences* 220:57-131 (1973).

[2] Pinneo, L.R. and Hall, D.J. 'Feasibility Study for Design of a Biocybernetic Communication System,' Stanford Research Institute final report (1975).

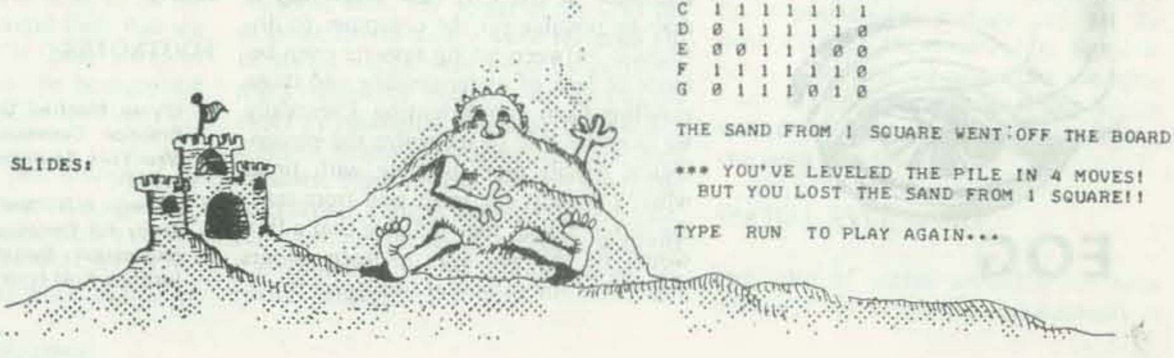
SANDPILE

BY MAC OGLESBY

Grab the controls of a giant bulldozer – try to level the pile of sand without losing any of it.



```
WELCOME TO THE SANDPILE!  YOU HAVE A BULLDOZER AND A PILE OF SAND.

    !             *
&L MMM          * * *
@VVVVV@=(     * * * * *

TOP VIEW AT START:

A  0 0 0 0 0 0 0
B  0 1 1 1 1 1 0
C  0 1 2 2 2 1 0
D  0 1 2 3 2 1 0
E  0 1 2 2 2 1 0
F  0 1 1 1 1 1 0
G  0 0 0 0 0 0 0

THE NUMBERS TELL SAND DEPTH AND THE LETTERS IDENTIFY ROWS.

WANT COMPLETE INSTRUCTIONS? YES

THE SANDPILE IS CENTERED ON A BOARD OF 49 SQUARES (7 BY 7).
YOU START WITH ENOUGH SAND TO COVER 35 SQUARES TO DEPTH 1.
YOUR JOB IS TO LEVEL THE ENTIRE PILE TO DEPTH 1 WITHOUT
LOSING ANY OF IT.  SAND WHICH IS PUSHED OFF OR WHICH SLIDES
OFF THE BOARD IS CONSIDERED LOST.

AT EACH TURN YOU DRIVE THE DOZER ALONG A ROW OF YOUR CHOICE.
YOU ALSO CHOOSE DIRECTION (> OR <) AND BLADE ANGLE (\ OR /).
AT EACH PASS THE TOPMOST LAYER OF SAND IS PUSHED DIAGONALLY
AHEAD AND ONTO THE NEARBY SQUARES.

*** SAND SLIDES OVER ONTO NEARBY SQUARES IF THE DIFFERENCE
IN DEPTH IS 2 OR MORE.

ROW:  A - G  ? C
DIRECTION:    > OR <  ? <
BLADE ANGLE:    1=/  2=\  ? 1

TOP VIEWS AFTER MOVE 1
  1) AFTER THE DOZER PUSHES SAND:

A  0 0 0 0 0 0 0
B  1 2 2 2 2 1 0
C  0 0 1 1 1 0 0
D  0 1 2 3 2 1 0
E  0 1 2 2 2 1 0
F  0 1 1 1 1 1 0
G  0 0 0 0 0 0 0

  2) AFTER THE SAND SLIDES:

A  0 1 1 1 1 0 0
B  1 1 1 1 1 1 0
C  0 1 2 1 1 1 0
D  0 1 1 2 1 1 0
E  0 1 2 2 2 1 0
F  0 1 1 1 1 1 0
G  0 0 0 0 0 0 0
```



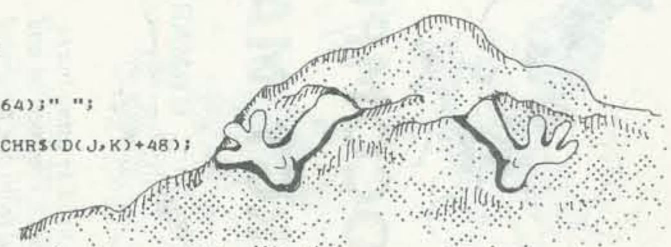
```
ROW:  A - G  ? E
DIRECTION:    > OR <  ? <
BLADE ANGLE:    1=/  2=\  ? 2

TOP VIEWS AFTER MOVE 2
  1) AFTER THE DOZER PUSHES SAND:

A  0 1 1 1 1 0 0
B  1 1 1 1 1 1 0
C  0 1 2 1 1 1 0
D  0 1 1 2 1 1 0
E  0 0 1 1 1 0 0
F  1 2 2 2 2 1 0
G  0 0 0 0 0 0 0

  2) AFTER THE SAND SLIDES:

A  0 1 1 1 1 0 0
B  1 1 1 1 1 1 0
C  0 1 2 1 1 1 0
D  0 1 1 2 1 1 0
E  0 0 1 1 1 0 0
F  1 1 1 1 1 1 0
G  0 1 1 1 0 1 0


ROW:  A - G  ? D
DIRECTION:    > OR <  ? >
BLADE ANGLE:    1=/  2=\  ? 2

TOP VIEWS AFTER MOVE 3
  1) AFTER THE DOZER PUSHES SAND:

A  0 1 1 1 1 0 0
B  1 1 1 1 1 1 0
C  0 1 3 2 2 2 1
D  0 0 0 1 0 0 0
E  0 0 1 1 1 0 0
F  1 1 1 1 1 1 0
G  0 1 1 1 0 1 0

  2) AFTER THE SAND SLIDES:

A  0 1 1 1 1 0 0
B  1 1 1 2 1 1 0
C  0 1 1 1 1 1 1
D  0 1 1 1 1 1 0
E  0 0 1 1 1 0 0
F  1 1 1 1 1 1 0
G  0 1 1 1 0 1 0


ROW:  A - G  ? B
DIRECTION:    > OR <  ? <
BLADE ANGLE:    1=/  2=\  ? 2

TOP VIEWS AFTER MOVE 4
  1) AFTER THE DOZER PUSHES SAND:

A  0 1 1 1 1 0 0
B  0 0 0 1 0 0 0
C  1 2 2 2 2 1 1
D  0 1 1 1 1 1 0
E  0 0 1 1 1 0 0
F  1 1 1 1 1 1 0
G  0 1 1 1 0 1 0

  2) AFTER THE SAND SLIDES:

A  0 1 1 1 1 0 0
B  0 1 1 1 1 1 0
C  1 1 1 1 1 1 1
D  0 1 1 1 1 1 0
E  0 0 1 1 1 0 0
F  1 1 1 1 1 1 0
G  0 1 1 1 0 1 0

THE SAND FROM 1 SQUARE WENT OFF THE BOARD!

*** YOU'VE LEVELED THE PILE IN 4 MOVES!
  BUT YOU LOST THE SAND FROM 1 SQUARE!!

TYPE  RUN  TO PLAY AGAIN...
```

```
WELCOME TO THE SANDPILE!  YOU HAVE A BULLDOZER AND A PILE OF SAND.

    !             *
&L MMM          * * *
@VVVVV@=(     * * * * *

TOP VIEW AT START:

A  0 0 0 0 0 0 0
B  0 1 1 1 1 1 0
C  0 1 2 2 2 1 0
D  0 1 2 3 2 1 0
E  0 1 2 2 2 1 0
F  0 1 1 1 1 1 0
G  0 0 0 0 0 0 0

THE NUMBERS TELL SAND DEPTH AND THE LETTERS IDENTIFY ROWS.

WANT COMPLETE INSTRUCTIONS? NO

ROW:  A - G  ? F
DIRECTION:    > OR <  ? <
BLADE ANGLE:    1=/  2=\  ? 2

TOP VIEWS AFTER MOVE 1
  1) AFTER THE DOZER PUSHES SAND:

A  0 0 0 0 0 0 0
B  0 1 1 1 1 1 0
C  0 1 2 2 2 1 0
D  0 1 2 3 2 1 0
E  0 1 2 2 2 1 0
F  0 0 0 0 0 0 0
G  1 1 1 1 1 0 0

  2) AFTER THE SAND SLIDES:

A  0 0 0 0 0 0 0
B  0 1 1 1 1 1 0
C  0 1 2 2 2 1 0
D  0 1 2 2 2 1 0
E  0 1 1 1 1 1 0
F  0 1 1 1 0 1 0
G  1 1 1 1 1 0 0


ROW:  A - G  ? B
DIRECTION:    > OR <  ? >
BLADE ANGLE:    1=/  2=\  ? 2

TOP VIEWS AFTER MOVE 2
  1) AFTER THE DOZER PUSHES SAND:

A  0 0 1 1 1 1 1
B  0 0 0 0 0 0 0
C  0 1 2 2 2 1 0
D  0 1 2 2 2 1 0
E  0 1 1 1 1 1 0
F  0 1 1 1 0 1 0
G  1 1 1 1 1 0 0

  2) AFTER THE SAND SLIDES:

A  0 0 1 1 1 1 1
B  0 0 0 1 1 1 0
C  0 1 1 1 1 1 0
D  0 1 2 2 2 1 0
E  0 1 1 1 1 1 0
F  0 1 1 1 0 1 0
G  1 1 1 1 1 0 0
```

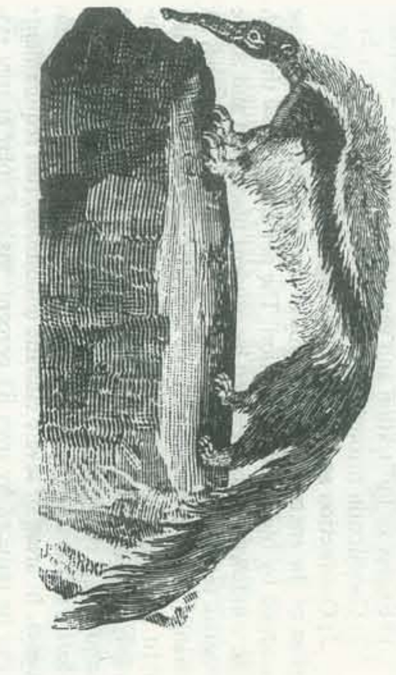
```
ROW:  A - G  ? D
DIRECTION:    > OR <  ? <
BLADE ANGLE:    1=/  2=\  ? 1

TOP VIEWS AFTER MOVE 3
  1) AFTER THE DOZER PUSHES SAND:

A  0 0 1 1 1 1 1
B  0 0 0 1 1 1 0
C  1 2 2 2 2 1 0
D  0 0 1 1 1 0 0
E  0 1 1 1 1 1 0
F  0 1 1 1 0 1 0
G  1 1 1 1 1 0 0

  2) AFTER THE SAND SLIDES:

A  0 0 1 1 1 1 1
B  0 0 1 1 1 1 0
C  1 1 1 2 1 1 0
D  0 1 1 1 1 1 0
E  0 1 1 1 1 1 0
F  0 1 1 1 0 1 0
G  1 1 1 1 1 0 0


ROW:  A - G  ? D
DIRECTION:    > OR <  ? <
BLADE ANGLE:    1=/  2=\  ? 2

TOP VIEWS AFTER MOVE 4
  1) AFTER THE DOZER PUSHES SAND:

A  0 0 1 1 1 1 1
B  0 0 1 1 1 1 0
C  1 1 1 2 1 1 0
D  0 0 0 0 0 0 0
E  1 2 2 2 2 1 0
F  0 1 1 1 0 1 0
G  1 1 1 1 1 0 0

  2) AFTER THE SAND SLIDES:

A  0 0 1 1 1 1 1
B  0 0 1 1 1 1 0
C  1 1 1 1 1 1 0
D  0 0 1 1 1 1 0
E  1 1 1 1 1 1 0
F  0 1 1 1 1 1 0
G  1 1 1 1 1 0 0


*** YOU'VE LEVELED THE PILE IN 4 MOVES!
*** AND YOU DIDN'T LOSE ANY SAND!!

TYPE  RUN  TO PLAY AGAIN...
```

SANDPILE–PROGRAM LISTING

```
100 ' NAME:  SANDPILE***
110 '
120 ' BY:  MAC OGLESBY ON 06/29/76
130 '
140 ' DESCRIPTION:  WITH A BULLDOZER AND A PILE OF SAND, DRIVE THE
150 '    DOZER BACK AND FORTH AND TRY TO LEVEL THE PILE WITHOUT LOSING
160 '    ANY OF IT.
170 '
180 ' INSTRUCTIONS:  TYPE "RUN" FOR COMPLETE INSTRUCTIONS.
190 '
200 ' CATEGORY:  TEENCAT***
210 '
220 ' LANGUAGE:  BASIC
230 '
240 ' INDEX LINE:
250 '    CAN YOU !LEVEL THE PILE OF !SAND WITH THE BULLDOZER?
260 '
270 '
1000 RANDOMIZE
1010 FOR J=2 TO 6
1020    FOR K=2 TO 6
1030       READ D(J,K)                         'NUMBERS WHICH SHOW SAND DEPTH
1040    NEXT K
1050 NEXT J
1060    DATA 1,1,1,1,1,1,2,2,2,1,1,2,3,2,1,1,2,2,2,1,1,1,1,1,1
1070
1080 FOR J=1 TO 8                             'CHANGES NEEDED TO ROW, COL. TO
1090    READ R(J),C(J)                        '  EXAMINE THE 8 NEARBY SQUARES
1100 NEXT J
1110    DATA -1,-1,-1,0,-1,1,0,-1,0,1,1,-1,1,0,1,1
1120
1130 PRINT
1140 PRINT "WELCOME TO THE SANDPILE!  ";
1150 PRINT "YOU HAVE A BULLDOZER AND A PILE OF SAND."
1160 PRINT
1170 PRINT
1180 FOR J=1 TO 5
1190    LET T$=T$&CHR$(34)
1200 NEXT J
1210 LET C1$=CHR$(13)&CHR$(13)
1220 PRINT TAB(5);"!";TAB(16);"*"       'PRINT A DECORATION
1230 PRINT "&L MMM";C1$;"'L WWW";TAB(14);"* * *"
1240 PRINT "0.....0";C1$;"*";T$;"*=[";TAB(12);"* * * * *"
1250
1260 PRINT
1270 PRINT "TOP VIEW AT START:"
1280 GOSUB 2550
1290 PRINT "THE NUMBERS TELL SAND DEPTH AND THE LETTERS IDENTIFY ROWS."
1300 PRINT
1310 PRINT "WANT COMPLETE INSTRUCTIONS";
1320 LINPUT A$
1330 LET A$=SEG$(A$,1,1)
1340 CHANGE A$ TO A
1350 IF (121-A(1))*(89-A(1))<>0 THEN 1390  'ALLOW FOR LOWERCASE Y
1360    GOSUB 2670
1370
1380    'GET PLAYER'S MOVE
1390 PRINT
1400 PRINT "ROW:  A - G  ";
1410 INPUT A$
1420 LET A$=SEG$(A$,1,1)
1430 CHANGE A$ TO A
1440 IF A(1)<96 THEN 1460
1450    LET A(1)=A(1)-32                     'CHANGE LOWERCASE TO UPPER
1460 IF (71-A(1))*(A(1)-65)=>0 THEN 1490
1470    PRINT "PLEASE TYPE A LETTER FROM A TO G  ";
1480    GOTO 1410
1490 PRINT "DIRECTION:   > OR <  ";
1500 INPUT A$
1510 IF A$=">" THEN 1570
1520    IF A$="<" THEN 1550
1530       PRINT "PLEASE TYPE > OR <  ";
1540       GOTO 1500
1550    LET D=-1
1560    GOTO 1580
1570 LET D=1
1580 PRINT "BLADE ANGLE:   1=/  2=\  ";
1590 INPUT B
1600 IF B=1 THEN 1660
1610    IF B=2 THEN 1640
1620       PRINT "PLEASE TYPE 1 OR 2  ";
1630       GOTO 1590
1640    LET B=-1
1650
1660 LET T=T+1
1670
1680 'THE DOZER PUSHES SAND
1690 LET J=A(1)-64                         'IDENTIFY THE ROW
1700 FOR K=1 TO 7
1710    IF D(J,K)=0 THEN 1740
1720       LET D(J,K)=D(J,K)-1              'MOVE SOME SAND
1730       LET D(J+B*D,K+D)=D(J+B*D,K+D)+1
1740 NEXT K
1750 PRINT
1760 PRINT "TOP VIEWS AFTER MOVE";T
1770 PRINT "  1) AFTER THE DOZER PUSHES SAND:"
1780 GOSUB 2550
1790
1800 'SAND SLIDES
1810 LET S=0                               'RESET SLIDE FLAGS
1820 LET Q=0
1830 FOR J=1 TO 7                          'CHECK 49 SQUARES IN TURN
1840    FOR K=1 TO 7
1850       IF D(J,K)<2 THEN 2020           'SEE IF ENOUGH SAND FOR A SLIDE
1860          FOR L=8 TO 1 STEP -1
1870             LET M=1+INT(RND*L)        'PICK NEARBY SQUARE AT RANDOM
1880             'LOOK FOR DEPTH DIFFERENCE OF 2 OR MORE
1890             IF ABS(D(J,K)-D(J+R(M),K+C(M)))<2 THEN 1940
1900                LET Q=S=1
1910                LET D(J,K)=D(J,K)-1  'SHIFT SOME SAND
1920                LET D(J+R(M),K+C(M))=D(J+R(M),K+C(M))+1
1930                GOTO 2020              'GO CHECK NEXT SQUARE
1940             IF M=L THEN 2010          'REARRANGING NOT NECESSARY
1950                LET A9=R(L)            'ELSE SWAP VALUES JUST TRIED
1960                LET R(L)=R(M)          '  WITH THOSE AT BOTTOM OF LIST
1970                LET R(M)=A9
1980                LET A9=C(L)
1990                LET C(L)=C(M)
2000                LET C(M)=A9
2010          NEXT L                       'AND SHORTEN THE LIST
2020    NEXT K
2030 NEXT J
2040 IF Q=0 THEN 2060
2050    GOTO 1820                          'SOME SAND SLID; CHECK PILE AGAIN
2060 IF S=1 THEN 2090
2070    PRINT "THERE WERE NO SAND SLIDES!"
2080    GOTO 2130
2090 PRINT "  2) AFTER THE SAND SLIDES:"
2100 GOSUB 2550
2110
2120 'DUMP THE SAND WHICH LEFT THE BOARD
2130 LET S=0
2140 FOR J=0 TO 8
2150    FOR K=0 TO 8
2160       IF (7-J)*(J-1)<0 THEN 2190
2170       IF (7-K)*(K-1)<0 THEN 2190
2180          GOTO 2220
2190       IF D(J,K)=0 THEN 2220
2200          LET S=S+1                     'TALLY SAND LOST THIS MOVE
2210          LET D(J,K)=0                  'DUMP LOST SAND
2220    NEXT K
2230 NEXT J
2240 IF S=0 THEN 2310
2250    PRINT "THE SAND FROM";S;"SQUARE";
2260    IF S=1 THEN 2280
2270       PRINT "S";
2280    PRINT " WENT OFF THE BOARD!"
2290
2300    'HAS PILE BEEN LEVELED?
2310 FOR J=1 TO 7
2320    FOR K=1 TO 7
2330       IF D(J,K)<2 THEN 2350
2340          GOTO 1390                     'CONTINUE GAME...
2350       LET Q=Q+D(J,K)                   'TALLY SAND LEFT
2360    NEXT K
2370 NEXT J
2380
2390 'END-OF-GAME MESSAGES
2400 PRINT
2410 PRINT "*** YOU'VE LEVELED THE PILE IN";T;"MOVES!"
2420 LET Q=35-Q
2430 IF Q=0 THEN 2490
2440    PRINT "  BUT YOU LOST THE SAND FROM";Q;"SQUARE";
2450    IF Q=1 THEN 2470
2460       PRINT "S";
2470    PRINT "!"
2480    GOTO 2500
2490 PRINT "*** AND YOU DIDN'T LOSE ANY SAND!!"
2500 PRINT
2510 PRINT "TYPE  RUN  TO PLAY AGAIN..."
2520 STOP
2530
2540 'PRINT THE BOARD
2550 PRINT
2560 FOR J=1 TO 7
2570    PRINT CHR$(J+64);"  ";
2580    FOR K=1 TO 7
2590       PRINT " ";CHR$(D(J,K)+48);
2600    NEXT K
2610    PRINT
2620 NEXT J
2630 PRINT
2640 RETURN
2650
2660 'INSTRUCTIONS
2670 PRINT
2680 PRINT "THE SANDPILE IS CENTERED ON A BOARD OF 49 SQUARES (7 BY 7)."
2690 PRINT "YOU START WITH ENOUGH SAND TO COVER 35 SQUARES TO DEPTH 1."
2700 PRINT "YOUR JOB IS TO LEVEL THE ENTIRE PILE TO DEPTH 1 WITHOUT"
2710 PRINT "LOSING ANY OF IT.  SAND WHICH IS PUSHED OFF OR WHICH SLIDES"
2720 PRINT "OFF THE BOARD IS CONSIDERED LOST."
2730 PRINT
2740 PRINT "AT EACH TURN YOU DRIVE THE DOZER ALONG A ROW OF YOUR CHOICE."
2750 PRINT "YOU ALSO CHOOSE DIRECTION (> OR <) AND BLADE ANGLE (\ OR /)."
2760 PRINT "AT EACH PASS THE TOPMOST LAYER OF SAND IS PUSHED DIAGONALLY"
2770 PRINT "AHEAD AND ONTO THE NEARBY SQUARES."
2780 PRINT
2790 PRINT "*** SAND SLIDES OVER ONTO NEARBY SQUARES IF THE DIFFERENCE"
2800 PRINT "IN DEPTH IS 2 OR MORE."
2810 PRINT
2820 RETURN
2830
2840 END
```



REVIEW



CALCULATORS, COMPUTERS, AND ELEMENTARY EDUCATION

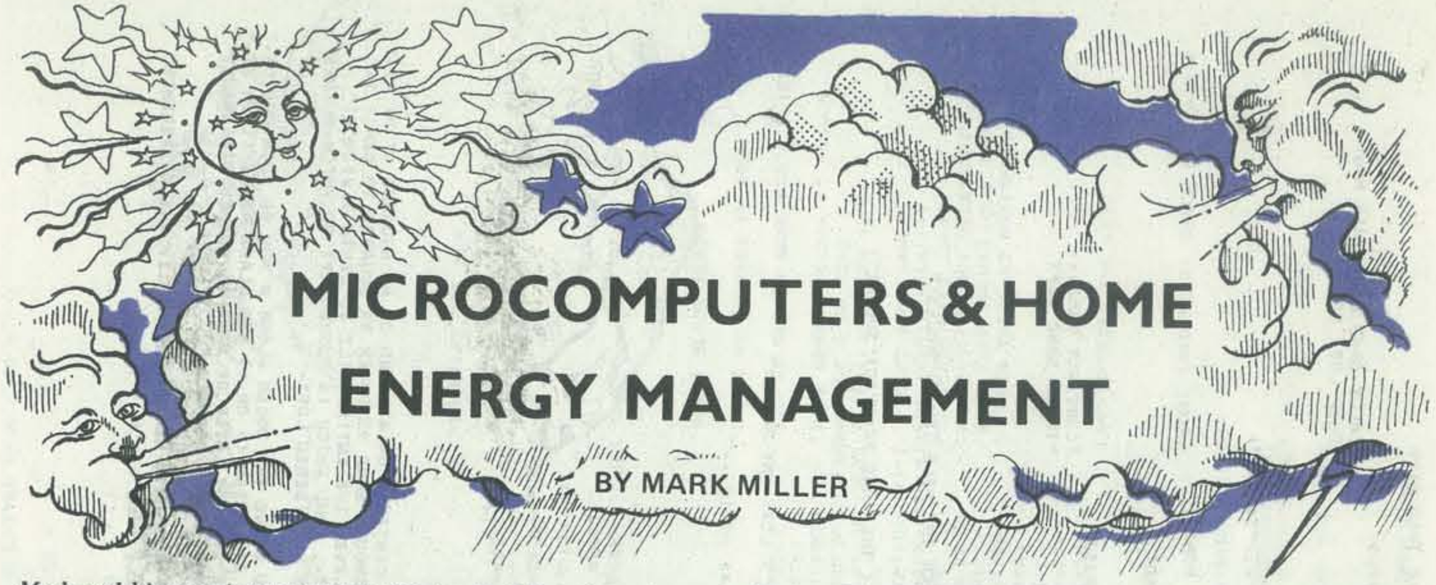
by David Moursund
From the Math Learning Center, 325 13th St NE, Salem, OR 97301; 1977, 160 pp, $7.00

David Moursund's book is designed to introduce calculators and computers to elementary school educators, and to make them more computer literate. It is quite well done. I read the book in two sittings, thereby getting a more concentrated dosage than an ordinary reader would have. I was impressed with how thoroughly he covered the topics (Problem solving, Calculators, Calculators in education, Computers, Computer programming, Computer science, and Computers in education).

My two favorite parts were the problems (end of chapter exercises – you know how bad they *can* be) and the discussion about school curriculum. Moursund has created a varied group of problems. There are thought-provoking 'things to do' as well as 'applications' and 'exercises'. Each category is slightly different, and all the problems are interesting to consider. As for the curriculum suggestions – it was refreshing to read incisive and practical suggestions for change by someone who obviously knows math teaching.

It's hard to judge how this book would strike a lay reader. People who really don't like math would have a hard time, but maybe they wouldn't buy it. Read it. It doesn't look like a seven-dollar book because it's not professionally published, and the graphics detract, but the content is *very good*.

Reviewed by Joanne Verplank.



MICROCOMPUTERS & HOME ENERGY MANAGEMENT

BY MARK MILLER

Mark and his associates are currently involved in using microcomputers for energy management in both commercial and residential buildings. This paper is adapted from one presented at the First West Coast Computer Faire; the original paper appears in the Faire Proceedings.

As efficient use of natural resources becomes more important and electronic costs go down, microcomputers will become more common in our daily lives. An area of immediate application is energy conservation and collection. In some climates utility bills for heating and cooling may be eliminated by the use of appropriate building design and low cost microprocessor control systems.

The energy losses in most buildings generally occur in the following order of importance:

1) Excess air infiltration and ventilation.
2) Conduction through the roof and walls.
3) Conduction and radiation through windows.

Most of the energy escaping through the roof and walls may be stopped by high grade insulation. The net energy flow through the windows may be reversed by prudent placement and active control techniques, described later. The energy lost through air change may be reduced by an order of magnitude or turned into a net gain by providing optimum ventilation continuously. For buildings in mild winter climates, the required heat input may come entirely from the active windows and incidental sources like appliances and the metabolic heat of occupants. In areas of severe climatic conditions, fuel requirements may be reduced by over 83 percent, making solar energy systems economical for almost everyone.

The target insulation 'R' value of 30 (double insulation) is not difficult to implement in new construction. Existing homes may even be upgraded to this level with plastic insulation.

The 'active windows' consist of double pane glass on the south facing walls. Insulation panels or shutters close automatically when solar heating or night-sky cooling systems are not operating. These panels have a metallic coating that reflects additional solar energy into the building when they are open. This technique is also suitable for retrofit applications for existing buildings.

A microcomputer energy management system may include the following functions:

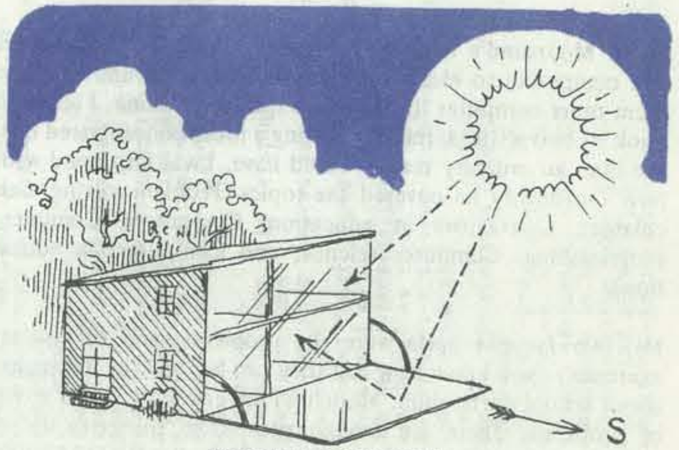
1) Monitoring air quality and human activity to provide appropriate ventilation control. Most buildings ventilate at about ten times the recommended rate when occupied. Reducing air change from 250 to 50 cubic feet per minute at an outside temperature differential of 17 degrees celsius saves the equivalent of 6,400 b.t.u. per hour. This equals about $1.80 per day with electric heating. Reducing the air change rate further when the building is unoccupied enhances savings.

2) The ventilation system can be used in the evening to cool the thermal masses inside the building, eliminating air conditioning costs during summer.

3) In some climates, indoor humidity control may be desirable. In all climates, relative humidity is an important variable for ventilation control programs.

4) Controlling of the active windows, solar water heaters, and so on, with data from the solar and thermal sensors.

5) Reporting the status of building environmental maintenance systems.



Movable reflective panel

6) Providing control for auxiliary functions (wind electric generating systems, backup heaters, lighting circuits, irrigation systems, intrusion alarms, fire sprinklers, and so on).

In the event of hardware addition or change, the control program may be changed by the supplier or the user. The software consists of modular algorithms which can be selected to suit the individual user's needs.

The computer hardware utilizes C-MOS technology to attain extremely low power consumption and high reliability. Our experience with S-100 bus type machines has shown that while they can make good controllers, single board systems with simple analog converters are superior. The simpler single board controllers consume less power and have fewer mechanical connections to fail. Most S-100 compatible Analog-Digital converters are expensive and fast. As conversion speed is not critical here, a simple homebrew design is appropriate. The input sensors may include thermistors, humistors, gas sensors, motion detectors, solar sensors and mechanical switches. Typical output devices are window panel drives, vent fans and register louvers, auxiliary power circuits and the like. Some of these devices equal the cost of the microprocessor controller, however these costs are going down.

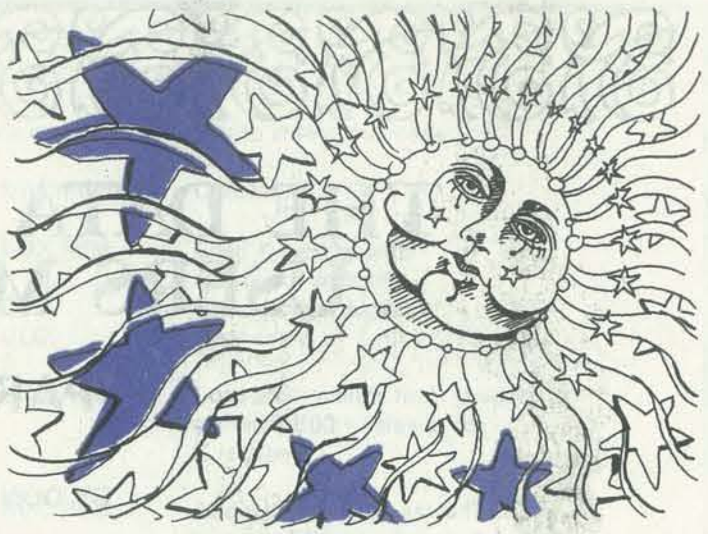
We are currently applying microcomputer controls to a commercial building and a modest foothill cabin. Both have 'passive' solar heating and cooling systems.

The prototype systems cost thousands of dollars. Even at these costs, the energy saved can pay for the system in about 6 years. As production and standardization evolve, some systems will sell for a few hundred dollars.

Some problems associated with systems of this kind are:

1) Lack of local maintenance facilities, especially in remote areas. This problem is minimized somewhat by use of a two card system with removable program storage units. These components may be changed and returned through the mail by the user. About 20 minutes each month is required for cleaning of the sensors and lubrication of the actuator drive mechanisms.

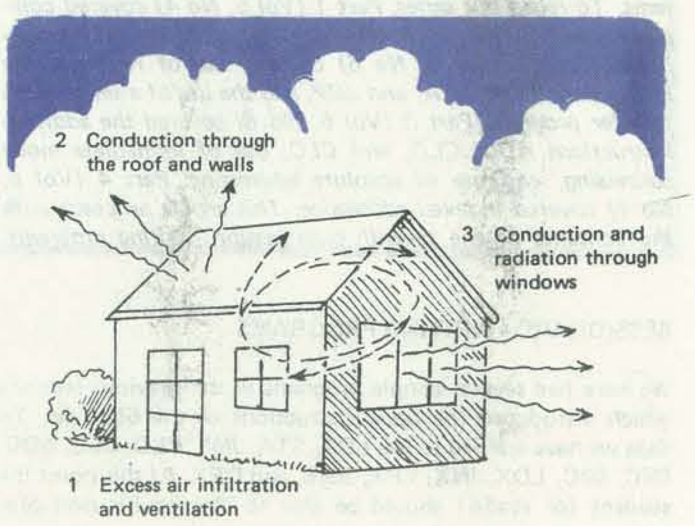
2) Nonstandard installations. Since building and climatic situations vary greatly, a standard software design may not be practical in some cases.



3) Low end ventilation. Air infiltration rates low enough for periods of low ventilation requirements may be hard to obtain. This is especially true in older buildings and at windy locations. Double door vestibule type entries and vapor barriers in all exterior walls and roof are required. Some manufacturers are claiming good results with sealant and paint type coatings. Special attention must also be given to window seals to reduce infiltration around the edges.

In summary: By structurally integrating the building and the energy collector-storage system, and utilizing intelligent control techniques, great thermal efficiency and economy can be realized. The increase in system efficiency made possible by high technology controls is essential to the simplification of the overall system. This allows substantial reduction of solar hardware. For many buildings, recurring fuel costs for heating and cooling can be practically eliminated without a gross investment in hardware.

Energy system control is an exciting application for home microcomputers. Others are doing work in this area. Correspondence is encouraged. I may be reached at: 821 Walnut St, Chico, CA 94926, Phone (916) 342-6102 or (916) 891-1300.


2 Conduction through the roof and walls
3 Conduction and radiation through windows
1 Excess air infiltration and ventilation

THE DATA HANDLER USERS MANUAL:

PART 5

BY DON INMAN

Don Inman is a former teacher, now editor of Calculators/ Computers, *who's been working with teachers in the San Jose School District. Under Don's guidance, the teachers have built Data Handlers, complete microcomputer systems based on the 6502 microprocessor, and are now learning to use them.*

This user's manual is designed to serve both as a self-teaching guide and as an outline for a course at the beginning level of computer science. While it deals specifically with the Data Handler, it can easily be adapted to other microcomputers using the MOS Technology 6502.

The first semester course consists of nine two-hour class sessions, the first two of which were spent constructing the systems. To recap our series, Part 1 (Vol 5, No 4) covered computer specification, computer notation and use of the keyboard. Part 2 (Vol 5, No 5) covered use of registers, the instructions LDA, STA, and JMP, and the use of a simple data transfer program. Part 3 (Vol 5, No 6) covered the addition instructions ADC, CLD, and CLC, use of immediate mode addressing, and use of absolute addressing. Part 4 (Vol 6, No 1) covered indexed addressing. This article proceeds with the contents of the seventh class session, writing programs.

SESSION VII – WRITING PROGRAMS

We have had several sample programs in the previous sections which introduced the basic instructions of the 6502 set. To date we have learned to use LDA, STA, JMP, CLD, CLC, ADC, SEC, SBC, LDX, INX, CPX, BNE, and DEX. At this point the student (or reader) should be able to develop his own programs.

Quoting from *Your Home Computer* (by James White, $6 from the PCC Bookstore):

> Get a few sample programs, preferably of moderate complexity. Learn what each instruction step does. Then write a very simple program. Test and correct it until it does what you want. When something doesn't work the way you expected, try to discover why. Then write a harder program adding the use of features you haven't become familiar with.
>
> Programming is best learned by doing it. With your own computer, you are excellently equipped to work—to learn as quickly as you want. Build your skills into the capability to handle complex programs, and you're well on your way to becoming a competent programmer.

New instructions can be introduced one or two at a time as Mr. White suggests. We will continue to introduce basic examples and techniques in the sections which follow, but students are encouraged to modify, add to, and develop their own programs. Programs can always be refined and improved so that they will be more efficient or more clear.

Each program should be documented so that others can use it with no difficulty. Branches and jumps should be clearly marked. A brief description of what the program accomplishes and how it does it should also be included.

A typical example of a student designed program is shown on the next page. This program adds 8-bit numbers located in successive locations starting with FE01. The number of addends to be used is entered at location FE00. (If two numbers are to be added, load 02 in FE00. For ten numbers load 0A, etc.).

INDEXED ADDRESSING ADDITION PROGRAM

Under the mnemonic column, 'ABS' refers to absolute mode; 'IMM' refers to immediate mode.

LABEL	ADDRESS	INST/DATA	MNEMONIC	COMMENTS
START	FC00	A2	LDX IMM	Load X register with number of addends.
	FC01	〈_〉		
	FC02	D8	CLD	Clear decimal mode.
	FC03	18	CLC	Clear carry.
	FC04	BD	LDA ABS	Load accumulator from memory address FE00 + value in the X register.
	FC05	00		
	FC06	FE		
LOOP1	FC07	7D	ADC IMM	Add with carry the value from memory FE01 + value in the X register.
	FC08	01		
	FC09	FE		
	FC0A	8D	STA ABS	Store the result into memory FD50.
	FC0B	50		
	FC0C	FD		
	FC0D	E8	INX	Increment X register for next number.
	FC0E	EC	CPX, FE00	Compare X register value with the value in memory FE00.
	FC0F	00		
	FC10	FE		
	FC11	D0	BNE	Branch if result is not equal to zero to LOOP1.
	FC12	F4*		
LOOP2	FC13	4C	JMP	Jump to location FC13.
	FC14	13		
	FC15	FC		

You are finished; the answer is in FD50.

*To determine this number, count from FC12 (inclusive) back to FC07: that's 12 instructions. We're counting backwards, so we need the negative binary equivalent to decimal 12. Here's how:

Express 12 in binary: $12 = 0000\ \ 1100$

Find the one's complement, $\overline{12}$ by changing 1's to 0's and 0's to 1's: $\overline{12} = 1111\ \ 0011$

Add 1 to the one's complement $+1$

to get the two's complement: $-12 = 1111\ \ 0100$

Now express the result in hex: $-12 = 1111\ \ 0100 = \text{F4}$

DOUBLE PRECISION ADDITION

If we limit ourselves to addition of 8-bit numbers, we limit ourselves to sums of less than 256. For practical problems, this will not be sufficient. By going to double precision it is possible to represent binary numbers on the order of 65,000. This should be sufficient for our present needs. The method of extension to higher order precision is the same as that for double precision. Therefore, further extension would be straightforward.

To create a 16-bit double precision number, we merely break our number into two 8-bit bytes. One part we call the low-order byte; the other is the high-order byte.

Consider the sum of 258 and 4112. Writing each in binary notation, we have:

```
     high order       low order
 0000  0001    0000  0010
       256   +        2        = 258

 0001  0000    0001  0000
  4096   +        16           = 4112
```

We first add the low order bytes of each number after clearing the carry bit.

```
              0000  0010
              0001  0000
low order sum 0001  0010 with no carry
```

We then add the high order bytes with any carry from the low-order sum.

```
               0000  0001
               0001  0000
                        0 carry
high order sum 0001  0001
```

Our result is:

0001 0001 0001 0010 = 4096 + 256 + 16 + 2 = 4370

Since the Data Handler is an 8-bit computer, the 16-bit result obtained in double precision addition must be stored in two different memory locations. We will now show one method of double precision addition using three numbers. Perhaps you will be able to write a shorter program to accomplish the same results.

When we add numbers, we ordinarily place the numbers in columns and perform a column addition as in the 'Hand Calculation' example. However, addition is really a binary operation performed on only two numbers at a time. This forms a partial sum to which the next number is added. Our computer is programmed in this manner as shown.

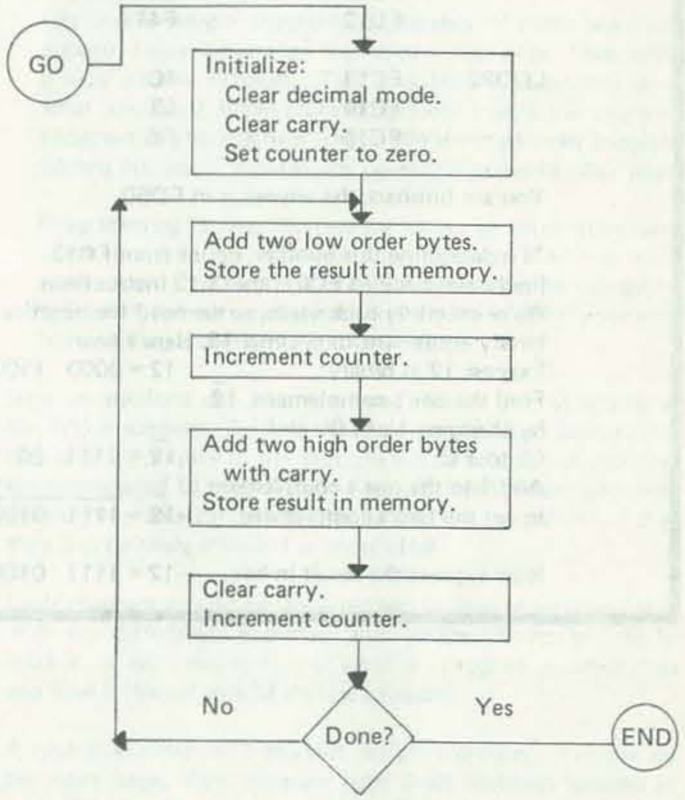
HAND CALCULATION

```
236   addition
 19   performed
216   by columns
 39   from right
196   to left with
 59   carry
---
765
```

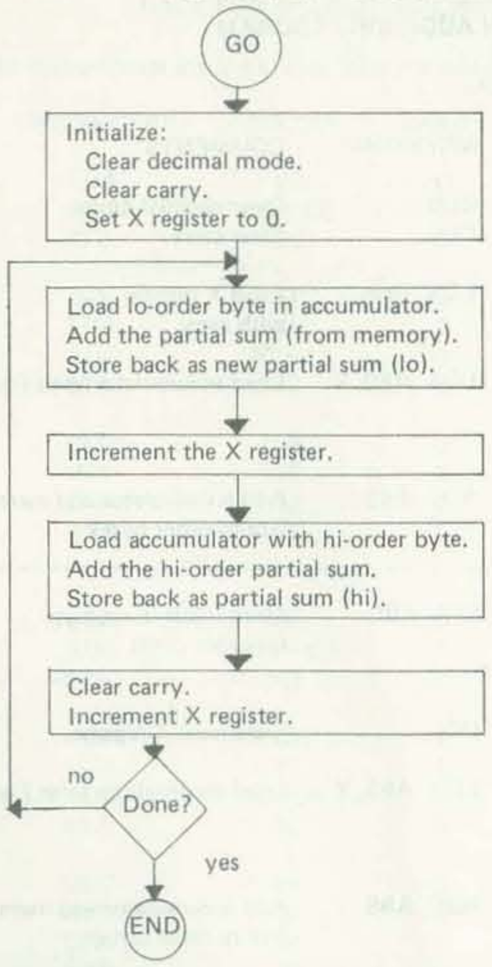
COMPUTER METHOD

```
two numbers    236
added to      + 19
form a         255    1st partial sum
partial       +216
sum to which   471    2nd partial sum
the next      + 39
number is      510    3rd partial sum
added         +196
               706    4th partial sum
              + 59
               765    final answer
```

The program to be developed will make use of a loop to build up the partial sum until all numbers have been added. We will develop the program through the use of some rather crude flow charts.



FLOW CHART FOR DOUBLE PRECISION ADDITION


GO
Initialize:
Clear decimal mode.
Clear carry.
Set X register to 0.
Load lo-order byte in accumulator.
Add the partial sum (from memory).
Store back as new partial sum (lo).
Increment the X register.
Load accumulator with hi-order byte.
Add the hi-order partial sum.
Store back as partial sum (hi).
Clear carry.
Increment X register.
no
Done?
yes
END

COMMENTS

Initialize the problem for hex addition.

Load lo-order byte of nth number, add in the previous partial sum, and store in same location.

Increment the counter for new number.

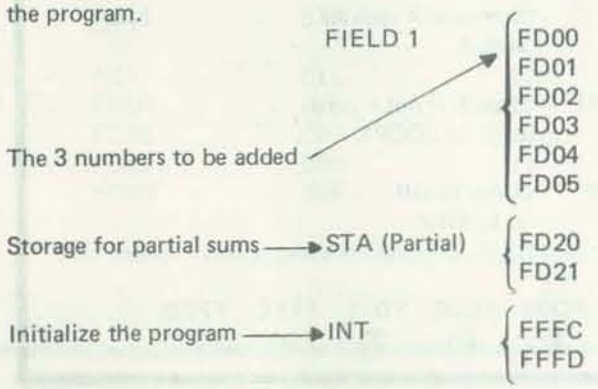
Load hi-order byte of nth number, add in the previous hi-order partial sum, and store back in in same location.

Clear the carry and increment the X register again.

Check to see if finished (counter will be incremented by two as each pass is made).

SECOND TRY

We now have second thoughts and draw a flow chart as shown above which will better fit our machine language instructions. We may now decide on the memory locations to be used in the program.



We are now ready to write mnemonic code to perform the desired additions. We present one version, followed by the machine language program and a trace of the program.

DOUBLE PRECISION ADDITION

```
LABEL   MNEMONIC        NUMBER
        CODE            OF BYTES

START   CLD             1
        CLC             1
        LDX, 00         2

LOOP1   LDA  ABS, X     3
        ADC  ABS        3
        STA  ABS        3
        INX             1
        LDA  ABS, X     3
        ADC  ABS        3
        STA  ABS        3
        CLC             1
        INX             1
        CPX, 06         2
        BNE, LOOP1      2

LOOP2   JMP, LOOP2      3
```

DOUBLE PRECISION ADDITION PROGRAM

The numbers to be added are 13EC, 27D8 and 3BC4.

LABEL	ADDRESS	INST/DATA	MNEMONIC	COMMENTS
START	FC00	D8	CLD	Clear decimal mode.
	FC01	18	CLC	Clear carry.
	FC02	A2	LDX IMM	Load X register with zero.
	FC03	00		
LOOP1	FC04	BD	LDA ABS, X	Load accumulator with Field1 + X.
	FC05	00		
	FC06	FD		
	FC07	6D	ADC ABS	Add accumulator and memory (for lo-order byte).
	FC08	20		
	FC09	FD		
	FC0A	8D	STA ABS	Store result in memory (lo-order byte).
	FC0B	20		
	FC0C	FD		
	FC0D	E8	INX	Increment X register.
	FC0E	BD	LDA ABS, X	Load accumulator with Field1 + X.
	FC0F	00		
	FC10	FD		
	FC11	6D	ADC ABS	Add accumulator and memory (for hi-order byte).
	FC12	21		
	FC13	FD		
	FC14	8D	STA ABS	Store result in memory (hi-order byte).
	FC15	21		
	FC16	FD		
	FC17	18	CLC	Clear carry for next lo-order sum.
	FC18	E8	INX	Increment X register.
	FC19	E0	CPX, 06	Compare X register with 6.
	FC1A	06		
	FC1B	D0	BNE, LOOP1	Branch if not = zero back to LOOP1.
	FC1C	E7		
LOOP2	FC1D	4C	JMP, LOOP2	Jump to self at LOOP2.
	FC1E	1D		
	FC1F	FC		

INITIAL LOADING:	FD00	FD01	FD02	FD03	FD04	FD05	FD20	FD21	FFFC	FFFD
	EC	13	DB	27	C4	3B	00	00	00	FC

TRACE FOR DOUBLE PRECISION ADDITION PROGRAM

This trace follows the computer actions as each instruction is encountered.

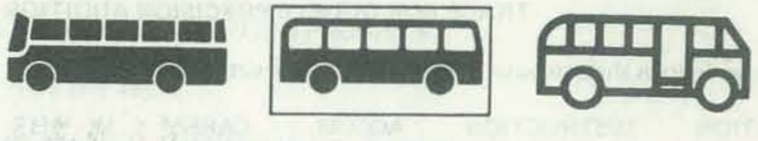
LOCATION	INSTRUCTION	ACCUM.	CARRY	X REG	FD20	FD21	COMPARE X
FC04	LDA	EC	0	0	0	0	
FC07	ADC	00 + EC	0	0	0	0	
FC0A	STA	EC	0	0	EC	0	
FC0D	INX	EC	0	1	EC	0	
FC0E	LDA	13	0	1	EC	0	
FC11	ADC	00 + 13	0	1	EC	0	
FC14	STA	13	0	1	EC	13	
FC17	CLC	13	0	1	EC	13	
FC18	INX	13	0	2	EC	13	
FC19	CPX	13	0	2	EC	13	6 – 2 = 0
FC1B	BNE	13	0	2	EC	13	Branch back.
FC04	LDA	DB	0	2	EC	13	
FC07	ADC	EC + DB	1	2	EC	13	
FC0A	STA	C7	1	2	C7	13	
FC0D	INX	C7	1	3	C7	13	
FC0E	LDA	27	1	3	C7	13	
FC11	ADC	13 + 27 + c	0	3	C7	13	
FC14	STA	3B	0	3	C7	3B	
FC17	CLC	3B	0	3	C7	3B	
FC18	INX	3B	0	4	C7	3B	
FC19	CPX	3B	0	4	C7	3B	6 – 4 = 0
FC1B	BNE	3B	0	4	C7	3B	Branch back.
FC04	LDA	C4	0	4	C7	3B	
FC07	ADC	C4 + C7	1	4	C7	3B	
FC0A	STA	8B	1	4	8B	3B	
FC0D	INX	8B	1	5	8B	3B	
FC0E	LDA	3B	1	5	8B	3B	
FC11	ADC	3B + 3B + c	0	5	8B	3B	
FC14	STA	77	0	5	8B	77	
FC17	CLC	77	0	5	8B	77	
FC18	INX	77	0	6	8B	77	
FC19	CPX	77	0	6	8B	77	6 – 6 = 0
FC1B	BNE	77	0	6	8B	77	No branch.
FC1D	JMP	77	0	6	8B	77	End of program.

The answer 778B appears in memories FD21 and FD20.

The next part of the series on programming the Data Handler will cover multiplication and division – despite the fact that the 6502 instruction set has no provision for these operations.

A BASIC PILOT

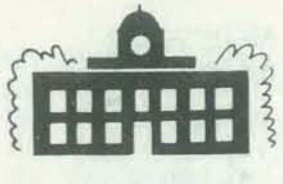
BY CHARLES SHAPIRO



Chuck Shapiro is a junior at Gunn High School in Palo Alto, California; He's been computing 'forever' – ever since he was turned onto it by Bob Albrecht while a student at Peninsula School in Menlo Park. Last year Chuck won the 'best overall' award in an ACM high school programming contest. His entry of a program for scheduling kids in school won him an HP-25 as a prize. For more information on PILOT, see the article 'PILOT CAI' in this issue as well as our back issues.

On this page is what happened when a player 'ran' the PILOT program shown on the next page. Each line that the player types is preceded by a question mark '?'.

```
RUN OF PROGRAM

BEFORE WE GET STARTED, WHAT IS YOUR NAME?
?CHUCK
 OK.. CHUCK , THERE ARE 2 DIFFERENT GAMES.
     1.) GUESS MY NUMBER.
     2.) COUNT THE SQUARES.
 WHICH ONE WOULD YOU LIKE TO PLAY?
?2

    ----------
    !  !  !  !
    !  !  !  !
    ==========
    !  !  !  !
    !  !  !  !
    ----------

HOW MANY 4 SIDED FIGURES CAN YOU FIND?
?9
NOPE, TRY AGAIN.
?6
 THOSE ARE JUST THE EASY ONES !!
?20
NOPE, TRY AGAIN.
?I GIVE UP
BETTER LUCK NEXT TIME!
 WHICH ONE WOULD YOU LIKE TO PLAY?
?1

 I'M THINKING OF A WHOLE NUMBER BETWEEN 0 AND 11.
TRY AND GUESS IT CHUCK .
?4
 TOO LOW !
?8
TOO HIGH !
?6
THATS IT !!!
 WHICH ONE WOULD YOU LIKE TO PLAY?
?2

    ----------
    !  !  !  !
    !  !  !  !
    ==========
    !  !  !  !
    !  !  !  !
    ----------

HOW MANY 4 SIDED FIGURES CAN YOU FIND?
?16
NOPE, TRY AGAIN.
?18
 VERY GOOD, THATS IT !!!
 WHICH ONE WOULD YOU LIKE TO PLAY?
?NONE

IT'S BEEN FUN PLAYING... CHUCK ...BYE..
 DOES ANYBODY ELSE WANT TO PLAY?
?NO
```



Here is a sample PILOT program that involves presenting a choice of two games to the player. One game involves guessing a number; the other involves counting squares. Here is a summary of the PILOT commands available in my PILOT interpreter:

T: Type prints out on the terminal the text following on the same line.

A: Ask causes the computer to pause and wait for user input.

M: Match checks to see if what the user typed in for the last Ask command matches any of the items listed after the M: command. If there is a match, the match flag is turned on; if there is no match, the match flag is not turned on.

J: Jump causes a branch to the label named: that command following the label is executed next.

Y: and N: Yes and No test the match flag. Y or N may be added to the T, A, M, and J commands. 'JY:' means 'jump, if the match flag is yes, or turned on.' 'TN:' means 'type if the match flag is no, or turned off.'

END: End stops execution of that part of the program in which it occurs.

```
*START
T:BEFORE WE GET STARTED, WHAT IS YOUR NAME?
A:$NAME
*GAMES
T: OK.. $NAME , THERE ARE 2 DIFFERENT GAMES.
T:     1.) GUESS MY NUMBER.
T:     2.) COUNT THE SQUARES.
*PICK
T: WHICH ONE WOULD YOU LIKE TO PLAY?
A:$GAME
T:
M:1,ONE,NUMBER,
JY:*NUMBER
M:2,TWO,SQUARES,
JY:*SQUARES
M:NONE,STOP,DONE,END,
JY:*END
T:I'M CONFUSED, TYPE 1 OR 2, $NAME ...THANK YOU
J:*PICK
*NUMBER
T: I'M THINKING OF A WHOLE NUMBER BETWEEN 0 AND 11.
T:TRY AND GUESS IT $NAME .
*GUESS
A:
M:1,2,3,4,5,
TY: TOO LOW !
JY:*GUESS
M:7,8,9,10,
TY:TOO HIGH !
JY:*GUESS
M:6,
TY: THATS IT !!!
JY:*PICK
T:THATS NOT A WHOLE NUMBER BETWEEN 0 AND 11.
T:TRY AGAIN.
J:*GUESS
*SQUARES
T:    ----------
T:    !  !  !  !
T:    !  !  !  !
T:    ==========
T:    !  !  !  !
T:    !  !  !  !
T:    ----------
T:
T:HOW MANY 4 SIDED FIGURES CAN YOU FIND?
*FIGUESS
A:
M:STOP,I DONT KNOW,I GIVE UP,
TY:BETTER LUCK NEXT TIME!
JY:*PICK
M:18,
TY: VERY GOOD, THATS IT !!!
JY:*PICK
M:6,
TY: THOSE ARE JUST THE EASY ONES !!
JY:*FIGUESS
T:NOPE, TRY AGAIN.
J:*FIGUESS
*END
T:IT'S BEEN FUN PLAYING... $NAME ...BYE..
T: DOES ANYBODY ELSE WANT TO PLAY?
A:
M:YES,SURE,Y,
JY:*START
END:
```

This is another experimental version of PILOT. I designed this version by using the write-up of PILOT in Volume 5 Number 3 issue as a guideline. This PILOT has a few things missing when compared to the original write-up of Tiny PILOT.

1) Line numbers are only used during the actual writing of the PILOT program. They are not used during execution so all jumps must be made to a label.
2) There is no U: command.
3) At this time there is no Compute statement. I consider this one of the major drawbacks that I will overcome in a future version.

This 'interpreter' is written in HP 3000 BASIC. HP 3000 BASIC has some very useful string manipulation functions. Before I get on to PILOT, here is a brief explanation of some characteristics of this BASIC.

Multiple Declarations:

```
200 H=M=F=17   is the same as
200 H=17, M=17, F=17
```

Multiple Branching:

```
10 GOSUB X OF 80,90,100,110
```

String Manipulation:

```
DIM P$(50)      A string containing
                a maximum of 50
                characters.
LET P$="ABCDEF"
where
  P$(3;1)="C"
  P$(3)="CDEF"
  P$(2;4)="BCDE"
  P$(2,4)="BCD"
DIM A$(5,50)    Five array elements
                each with a length
                of 50 characters.
  LET A$(1)="ABCDE"
  LET A$(2)="FGHIJ"
where
  A$(2,3)="HIJ"
  A$(2,3,4)="HI"
  A$(1,1;3)="ABC"
  A$(1)+A$(2,4)+"BYE"=
                 "ABCDEIJBYE"
```

NUL$ function sets all characters to blanks, (spaces).

I am in the process of writing a file system so PILOT programs may be saved and accessed using the already present operating system.

This interpreter was not written to be implemented on a microcomputer but I suspect it would be possible to translate to one of the available micro/hobbyist BASICs.

Lines 230 - 250
This is a list of possible commands that may be used in a PILOT program. Unknown commands, as long as they are followed by a ':', are ignored during execution.

Lines 280 - 400
Loop to enter PILOT program. The Edit function is tucked in here also.

```
10 REM * * C. SHAPIRO  1977-@GUNN HIGH
20 REM
30 REM * * PILOT INTERPRETER IN HP-3000 BASIC **
40 REM
50 REM * * * VARIABLES * * *
60 REM  P$(*)-PILOT PROGRAM MINUS "(COMMAND)"
70 REM  Q$(*)-COMMAND FOR EACH LINE
80 REM  I$(*)-PILOT INPUT LABELS AND INPUT
90 REM  V$   -VARIABLE WHOSE "VALUE" IS TO BE PRINTED
100 REM  C$(*)-TABLE OF PILOT COMMANDS
110 REM  R$    -OPERATING SYSTEM INPUT VARIABLE
120 REM  Z$    -CONTENTS OF LAST 'A:'
130 REM  A     -FOR-NEXT VARIABLE/PROGRAM COUNTER
140 REM  Z,Z1,H,E,D,C2-FOR-NEXT VARIABLES
150 REM  A1    -# OF LINES IN PILOT PROGRAM
160 REM  C     -STARTING LINE OF EDIT FUNCTION
170 REM  C1    -PILOT INPUT INDEX VARIABLE
180 REM  M     -RESULT OF LAST MATCH; 1=YES,0=NO
190 REM  E1     -MARKS LOCATION OF COMMA IN MATCH FUNCTION
200 REM
210 DIM P$(50,65),Q$(50,65),I$(10,65)
220 DIM V$(65),Z$(65),C$(15,5),R$(5)
230 C$(1)="END",C$(2)="A",C$(3)="T",C$(4)="M",C$(5)="J"
240 C$(6)="JY",C$(7)="JN",C$(8)="TY",C$(9)="TN"
250 C$(10)="MY",C$(11)="MN",C$(12)="AY",C$(13)="AN"
260 MAT P$=NUL$
270 MAT Q$=NUL$
280 REM
290 REM * * * INPUT PILOT PROGRAM
300 REM
310 FOR A=1 TO 100
320   PRINT A;"  ";
330   INPUT P$(A)
340   IF P$(A)="DONE" THEN 410
350   IF P$(A)<>"EDIT" THEN 400
360   REM * * EDIT FUNCTION
370   INPUT "STARTING AT LINE?",C
380   A=C=C-1
390   PRINT C;"  ";P$(C)
400 NEXT A
```

Lines 420 - 540
To make execution of the PILOT program faster the commands are separated from their respective lines of the program. For example, four lines of a program might look like this after being entered:

```
P$(1) -- !THIS IS A REMARK
P$(2) -- J:*OVER
P$(3) -- T:THIS IS JUMPED OVER.
P$(4) -- *OVER
```

After being separated the program and variables would be:

```
P$(1) -- !THIS IS A REMARK
Q$(1) -- (Nul String)
P$(2) -- *OVER
Q$(2) -- J
P$(3) -- THIS IS JUMPED OVER.
Q$(3) -- T
P$(4) -- *OVER
Q$(4) -- (Nul String)
```

No printout would result during execution.

Lines 730 - 870
This is the operating system which takes control after PILOT execution. Options are:

RUN the program again
LIST the PILOT program
Write NEW program (deleting old).

TYPE (T:), Lines 880 - 1100
Each T-line is scanned left to right looking for a $. If a $ is found all characters from $ up to and including the next space are treated as one variable. This variable is then matched against the variable names in I$ (see ASK). I$ is checked from the latest entries back to the first variable. (This avoids Dean Brown's 'straw man' problem of an earlier issue.) If a match is not found, the scan of the original line is continued, looking for another $; if no variables are found the string will be printed as is. If a match does occur, only the first variable in the line will be replaced by the input string it represents.

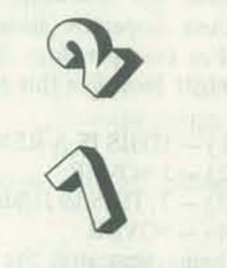
```
410 C1=0,A1=A-1
420 REM
430 REM * * PILOT COMMANDS PLACED IN Q$(*)
440 REM
450 FOR A=1 TO A1
460   FOR B=1 TO 65
470     IF P$(A,B;1)<>"*" AND P$(A,B;1)<>"!" THEN 490
480     GOTO 540
490     IF P$(A,B;1)<>":" THEN 520
500     Q$(A)=P$(A,1,B-1),P$(A)=P$(A,B+1,65)
510     GOTO 540
520   NEXT B
530   PRINT"COMMAND ERROR-LINE";A
540 NEXT A
550 PRINT LIN(1);"RUN OF PROGRAM";LIN(1)
560 REM
570 REM * * EXECUTION OF PILOT PROGRAM; ONE LINE AT A TIME
580 REM
590 FOR A=1 TO A1
600   FOR C2=1 TO 13
610     IF Q$(A)=C$(C2) THEN 640
620   NEXT C2
630   GOTO 690
640   IF C2=1 THEN 700
650   GOSUB C2 OF 210,1340,940,1170,1490,1460,1480,910,930
660   IF C2<=9 THEN 690
670   C2=C2-9
680   GOSUB C2 OF 1140,1160,1300,1320
690 NEXT A
700 PRINT LIN(1);"END OF RUN"
710 MAT I$=NUL$
720 C1=0
730 REM
740 REM * * ENTER OPERATING SYSTEM
750 REM
760 INPUT "REQUEST?",R$
770 IF R$="NEW" THEN 260
780 IF R$="RUN" THEN 550
790 IF R$<>"LIST" THEN 860
800 FOR A=1 TO A1
810   IF Q$(A)<>"" THEN 840
820   PRINT A;P$(A)
830   GOTO 850
840   PRINT A;Q$(A)+":"+P$(A)
850 NEXT A
860 IF R$="STOP" THEN STOP
870 GOTO 760
880 REM
890 REM * * * TYPE
900 REM
910 IF M=1 THEN 940
920 ELSE 1100
930 IF M=1 THEN 1100
940 FOR Z=1 TO 65
950   IF P$(A,Z;1)="$" THEN 980
960 NEXT Z
970 GOTO 1090
980 FOR Z1=Z TO 65
990   IF P$(A,Z1;1)=" " THEN 1020
1000 NEXT Z1
1010 GOTO 1090
1020 V$=P$(A,Z,Z1)
1030 FOR H=(C1-1) TO 1 STEP -2
1040   IF V$(1,15)<>I$(H,1,15) THEN 1070
```

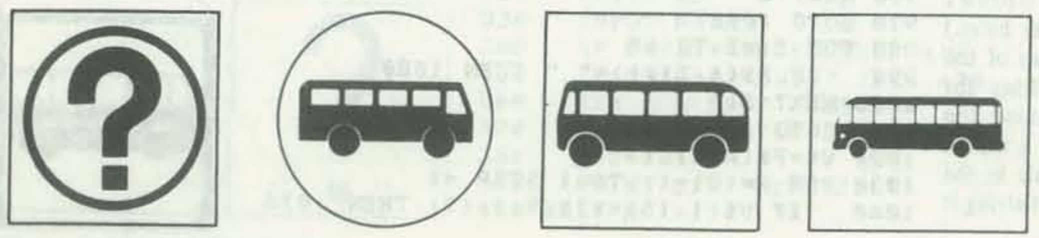
MATCH (M:), Lines 1110 - 1260
A match statement should look like this:
M:Item1,Item2,Item3,
there must be a comma after the last item. This is not a 'window' scan. The last input must match one of the items exactly to be successful. Two or more blanks after a comma signify no more items to be checked.

ASK (A:), Lines 1270 - 1420
If a $ is not found within 10 spaces after the colon it is assumed that the input will not be assigned to a variable. If a variable is present the variable name and the user's input are placed in I$ array for possible use in the future. In either case, the user's input is placed in Z$ for a match statement in case one follows. Z$ is updated with each user input.

JUMP (J:), Lines 1430 - 1570
The whole PILOT program is scanned line by line looking for an instance where the line label matches the jump 'operand'. The program counter is then set to the location after the label and program execution continues from that point. An error message is generated if a jump is attempted but no line label is found.

```
1050   PRINT P$(A,1;Z-1)+I$(H+1)+P$(A,Z1,65)
1060   GOTO 110
1070 NEXT H
1080 GOTO 960
1090 PRINT P$(A)
1100 RETURN
1110 REM
1120 REM * * INPUT MATCH
1130 REM
1140 IF M=1 THEN 1170
1150 ELSE 1260
1160 IF M=1 THEN 1260
1170 M=0,E1=1
1180 FOR E=1 TO 65
1190   IF P$(A,E;1)="," THEN 1210
1200   ELSE 1230
1210   IF P$(A,E1,E-1)=Z$ THEN 1250
1220   E1=E+1
1230   IF P$(A,E;2)="  " THEN 1260
1240 NEXT E
1250 M=1
1260 RETURN
1270 REM
1280 REM * * * ASK
1290 REM
1300 IF M=1 THEN 1340
1310 ELSE 1420
1320 IF M=1 THEN 1420
1330 REM * INPUT LABEL+INPUT ARE IN I$(*)
1340 FOR Z=1 TO 10
1350   IF P$(A,Z;1)<>"$" THEN 1400
1360   C1=C1+2,I$(C1-1)=P$(A)
1370   INPUT"?",I$(C1)
1380   Z$=I$(C1)
1390   GOTO 1420
1400 NEXT Z
1410 INPUT"?",Z$
1420 RETURN
1430 REM
1440 REM * * JUMPING ROUTINE
1450 REM
1460 IF M=1 THEN 1490
1470 ELSE 1530
1480 IF M=1 THEN 1530
1490 FOR D=1 TO A1
1500   IF Q$(D)<>"" THEN 1540
1510   IF P$(A,1,40)<>P$(D,1,40) THEN 1540
1520   A=D
1530   RETURN
1540 NEXT D
1550 PRINT"JUMP TO UNFOUND LABEL"
1560 GOTO 700
1570 END
```





ANNOUNCEMENTS

RADIO SHACK'S MICRO

Radio Shack has introduced its new TRS-80 Microcomputer System. The TRS-80 comes completely wired and tested, ready to plug in and use. A comprehensive owner's manual will be supplied which will cover operation and include instructions on programming.

Pre-recorded cassette programs will include a blackjack/backgammon tape supplied free with the unit, a small payroll package ($19.95), a math education portfolio (3 tapes, $19.95), a kitchen program ($4.95) and a personal finance portfolio (7 cassettes, $14.95). There are also plans for future applications software including general ledger, inventory control and a scientific subroutine package.

The TRS-80 is a Z-80 based system and comes with 4K of ROM and 4K of dynamic RAM internally expandable to 12K ROM and 16K RAM respectively. The 12 inch video display is 16 lines by 64 characters. There will be a computer-controlled cassette interface and an expansion port for additional memory and peripherals. The microcomputer and keyboard is 16½ x 8 x 3½", and the video display monitor is 16½ x 13½ x 12".

Radio Shack's Level I BASIC will reside in ROM and include the standard BASIC statements as well as numeric, array and string variables and video graphics commands.

The Radio Shack TRS-80 Microcomputer System is priced at $599.95, complete with video display monitor and data cassette recorder. The microcomputer alone is available for $399.95.

Deliveries will start September 1. We hope to have a machine to review in time for our November-December issue.

EMULATION FOR ANY PROCESSOR

A new emulation methodology has been developed by M & E ASSOCIATES and is to be manufactured and marketed by DIGITAL SYSTEMS. The Z80 based development system combines the power of a special assembler with a memory emulation module to give the user the first truly 'universal development system.' The assembler allows the insertion of any instruction mnemonic including multibyte instructions like those of the Z80. The user can set the assembler's symbol table for an 8080, F80, 6800, F8 and 6502, etc., and generate the appropriate object code. The object code is loaded into the emulator memory space where it can be executed by the target processor. No more compatibility problems between the emulator's CPU (i.e., I8080) and target system's CPU (NEC 8080). A buffered connector plugs into the target system's 2708 EPROM socket. Connectors for three (3) more 2708 chip selects allow the emulation of 4K bytes with the standard system.

Any target microcomputer system with a standard 2708 can be programmed and debugged using breakpoints. Once a program is debugged the user can program his 2708 EPROM via a RS232 link.

Only one editor, one operating system, one assembler, and one set of utilities need be learned to develop software for the microprocessor of your choice.

Shipments are scheduled to begin in September 1977, with 30 - 60 day delivery.

The development/emulator system, including Z80 CPU, 32K of RAM, 4K emulator RAM, dual floppy disk single density, will sell for $5,995. Write to: M & E ASSOCIATES, 10439 N. Stelling Road, Cupertino, CA 95014.

THREE DIMENSIONAL GRAPHICS

Sublogic Company of Culver City, California announces 3D graphics for microcomputers. The Sublogic 3D micrographics package will allow a user to view two dimensional perspective projections of three dimensional scenes from any location in space. Driving and flying simulations, artistic projections, design projections, engineering analysis, and advanced games are now simple and economical.

Two versions of the graphics package will be offered. A minimal subset BASIC version will be ideal for general purpose, slow speed graphics on any microcomputer system. The 6800 optimized assembly language version with dynamic graphic capabilities is ideal for advanced simulation and complex graphics.

Simple adaptation instructions, program listings, applications, interface, and testing information will be supplied with each package.

The BASIC version will retail for $22. The 6800 package will be priced slightly higher. Contact Sublogic, P.O. Box 3442, Culver City, CA 90230.

WORD PROCESSING SYSTEM

The Electric Pencil is a character oriented word processing system: text is entered as a continuous string of characters and is manipulated as such. Since lines are not delineated, any number of characters, words, lines or paragraphs may be inserted or deleted anywhere in the text. The entirety of the text shifts and opens up or closes as needed in full view of the user. The typing of carriage returns as well as word hyphenation is not required since each line of text is formatted automatically. Whenever text is inserted or deleted, existing text is pushed down or pulled up in a wrap around fashion. Text may be reviewed at will by variable speed scrolling both in the forward and reverse directions.

By using the search or the search and replace function, any string of characters may be located and/or replaced with any other string of characters as desired. Specific sets of characters within encoded strings may also be located and used in creating selective mailing lists. Numerous combinations of line length, page length, line spacing and page spacing allow for any form to be handled. Character spacing and bi-directional printing are included in the Diablo versions. Right justification gives right-hand margins that are even. Pages may be numbered as well as titled.

The Electric Pencil Software is available on Tarbell cassette in four versions: Standard versions are $100 and Diablo versions are $150. For further information and mail orders, contact: Michael Shrayer, 3901 Los Feliz Blvd., #210, Los Angeles, CA 90027.

CHICAGO, OCT 27-29

Another 'Personal Computing' show will be held October 27-29, at the Holiday Inn at Chicago's O'Hare International Airport. One of the many planned door prizes is Processor Technology's newest kit product, Sub System B190 with 16K memory, valued at over $1,000.

In addition to a full array of products, a comprehensive seminar program is being planned. Included is a two-day conference for entrepreneurs and businessmen interested in Personal Computing, called 'Getting a Piece of the Action in Personal Computing'. Another special event will be 'Everyone's Computer School'. The school will teach the basics of programming to beginning computer users.

Admission to the exhibit hall is $10. For further information contact: *Personal Computing* Magazine, 401 Louisiana SE, "G", Albuquerque, NM 87108; telephone (505) 255-8661.

LOS ANGELES, NOV 1-3

INTERFACE WEST is a three-day conference and exposition for computer and communications users and designers to be held November 1, 2 and 3 at the Los Angeles Convention Center; it is cosponsored by *Datamation* Magazine. Sessions will be aimed primarily at designers of micro-based computer peripherals and communications equipment. Other complete programs at INTERFACE WEST will concentrate on Data Communications, Telecommunications, and Small Systems.

For more information, contact Alan R. Kaplan, Director of Program Development, Interface West, 160 Speen St., Framingham, MA 01701, telephone (800) 225-4620 or, from Massachusetts, (617) 879-4502.

HANDS-ON MICRO TUTORIAL

The Nuclear & Plasma Sciences Society, IEEE, in conjuction with their Annual Symposium are sponsoring a Hands-On Micro-computer Applications Tutorial on Saturday, October 22, 1977 from 9:00 - 5:00 at the Sheraton Palace Hotel, San Francisco. The tutorial is slanted to persons having a computer background and desiring hands-on experience on actual problems. Suitcase trainers based on the 8080 will be used. Eugene Fisher, Lawrence Livermore Laboratory and Michael Maples, M. & E. Associates, microcomputer systems engineers, teachers, and consultants will lead the workshop. Projects include a stop watch timer display and a set-point temperature controller. It is recommended that attendees study in advance materials which will be sent out prior to the tutorial. Class size is limited; fee is $40. A.J. Stripeika, Lawrence Livermore Laboratory, Box 808, Livermore, CA 94550.

COMPUTER & THE ARTS

'The Computer and the Arts,' a mini-course, will be offered at Columbia University this October. In addition to seeing and hearing films, slides, and recordings, students will get some individual experience with computer tools for their own graphic and verbal experimentation. Also, the SIGLASH Conference, 'Computing in the Arts and Humanities,' will be held in New York October 21-23, while the course is in progress. The course meets on 5 Wednesdays, 6 to 8 PM, and costs $100. More information on this and other introductory computer mini-courses is available from Jessica Gordon, Assistant Director of the Center for Computing at (212) 280-2454 and on the Columbia Program for Continuing Education at (212) 280-3331.

MICRO-CHESS TOURNEY

The first annual Micro-chess tourney will be held in Louisville, Kentucky in Augusc of 1978 and we are scurrying around getting ready for the big event. To put on a really fair tournament we are in the process of drawing up the rules and regulations now.

- Competition limited to approved 8 bit micro-processors, no bit slice machines will be allowed. (But other microprocessors will be considered. Send request with SASE to address below.)
- Programs can be in either machine language or a higher level language.
- 16K 8 bit words memory maximum. (9 bits if parity is used.)
- Home-brew machines and commercial machines allowed.
- Top three winners software published through leading magazines.
- Machines may be loaded from any media but after the program is operating the loading device must be detached.
- A panel of judges will rule promptly on program crashes or other unexpected problems.
- Competition will be timed.

For further information write: Louisville Area Computer Club, 3028 Hunsinger Lane, Louisville, Kentucky 49220. Or phone: (812) 283-4128.

STAR JEWELS

Advanced Technology Research Associates (ATRA) is now an authorized distributor of Star Jewels. Yes, science fiction has become science fact! Each Star Jewel pendant has a red light emitting diode in mirrored multifaceted lucite setting. When you switch it on, a patented integrated circuit blinks the LED about three times a second. So, as the pendant swings lightly on its chain (it only weighs an ounce), the jewel seems to dance with inner reflections.

A Star Jewel is powered by two inexpensive hearing aid batteries which last for 2-3 months of use. The electronics and workmanship are unconditionally guaranteed (excluding batteries) for one full year. Star Jewels are available with red, green, amber, blue, silver and smoke golored gems. They come complete with pendant chain, batteries and a velveteen pouch for $31.25 which includes postage and insurance. Take one with you the next time you go to see Star Wars! Send your check or money order payable to ATRA, Box 456, Minneapolis MN 55440. Minnesota residents should add 4% sales tax.

DATASYNC LEADER CHARGED WITH FRAUD

Along with many other publications, we're responding to John Craig's request to spread the word on DataSync, a corporation operating out of Santa Maria, California and advertising in *Interface Age*, *Byte*, and *Kilobaud* in recent months. On July 1 the head of DataSync was arrested and charged with grand theft by false pretenses. The case is pending against Norman Henry Hunt, who used the name David Winthrop while associated with DataSync. Detective Ernie Kapphahn is with the Santa Maria Police Dept., 222 E Cook St., Santa Maria, CA 93454, telephone (805) 925-2831. Det. Kapphahn would appreciate information from anyone who has dealt with Hunt.

John Craig notes that four other officers of DataSync are honest, responsible individuals who are making every effort to fill orders already placed. They are asking for patience on the part of customers, since numerous requests for return of money may force the company to liquidate, in which case customers will receive back only 15¢ on the dollar.

GROUP PURCHASE VICTIMS & OTHER WOES

Kenneth Young has sent us an extensive letter about his group purchase woes, in hopes that knowledge of his experience will help prevent the situation from reoccuring. The letter arrived when our pages were already bursting at the margins, but we decided to make room for these excerpts.

I am one of approximately forty SCCS (Southern California Computer Society) Group Purchase victims. In April and May of 1976, I placed orders for over a thousand dollars worth of IMSAI products. My money was given by the SCCS to Mr. Thomas Hudson, President of Microprocessor Marketing.

It's been *well over* a year now and I have not received the equipment I had ordered and paid for. To the best of my knowledge, the SCCS does not know the present whereabouts of Mr. Thomas Hudson and has not been able to recover the money owed to us for undelivered products.

I feel that the SCCS as a whole has been relatively insensitive to the plight of the Group Purchase victims. No civil or criminal proceedings were ever filed by the SCCS against Microprocessor Marketing. As far as I'm concerned, the SCCS has let Mr. Thomas Hudson go scott free with my money and that of fellow members. Larry Press, the editor of *SCCS Interface*, did try to get this information about non-delivery from Microprocessor Marketing into the magazine but was over-ruled by a majority of the Board of Directors. At *least* SCCS should have told its members of the existence of an unreliable dealer to protect *other* members.

Nevertheless, the Board of Directors did authorize that a 20% refund be made to Group Purchase victims in December 1976 and that a monthly 10% payment be made the following eight months. No interest payments were requested by the victims nor was there any offered by the SCCS. However, the SCCS still owes me a little over $300.00.

Microprocessor Marketing is no longer an authorized IMSAI distributor; IMSAI justifiably (alas) feels they have no responsibility to the Group Purchase victims. Authorized IMSAI dealers are *not* really corporate extensions of IMSAI; it is the consumer's own fault and tough luck of he deals with a bad dealer. I suggest every consumer buy off-the-shelf products or that he order the product he wants C.O.D. *directly* from the manufacturer.

It's my own personal feelings that there will be more consumer rip-offs in the future. It really burns me up to see how understanding the computer hobby publications are to computer hobby manufacturers. There has been hardly any concern for the computer hobby consumer!!! A lot of consumers have been hurt by the 'computer hobby manufacturers and dealers'. Most of the computer hobby manufacturers and computer store dealers are not crooks, but the extremely long delays between product announcement and product delivery (in my opinion) encourage unscrupulous activity. For example, I ordered a Helios II system from Processor Technology in early December 1976 and it still has yet to arrive. A friend of mine ordered a Digital Group printer in early January and he is now being told to expect delivery sometime in September or October. Then there are Peripheral Vision's advertisements for their $750 floppy. The Computer Store in Santa Monica ordered several of these in January and they have not yet arrived. I suspect the Computer Store may have cancelled this order since there is now better stuff available. Of course, I shouldn't leave out IMSAI's advertisement for their color graphics boards which were announced last year. Has anyone received any of those 'phantom' boards? I could go on and on, but I've said enough for now.

Please try to be more consumer oriented and help prospective micro-computer enthusiasts (as well as the old-timers) find the best products to buy on the market and the best people to deal with.

Kenneth Young
3311 West 3rd Street
Apartment 1-319
Los Angeles, California 90020