The Boston Computer Society

SINCLAIR-TIMEX USER GROUP NEWSLETTER

Volume 2, Issue 4 April 1983

This newsletter is produced to inform group members of the agenda and logistics for future meetings, as well as to recap and amplify the information provided at the last meeting. It also provides a forum for members and interested parties to communicate what they have learned or developed relating to Sinclair and Timex computer products. Meetings are open to the public; however, attendees are encouraged to join the Boston Computer Society (BCS). This newsletter is free to members. Back issues are one dollar each.

USER GROUP MEETING

Date: Wednesday, April 13, 1983
Time: 7:00 p.m.
Place: Large Science Auditorium
UMass, Harbor Campus
(Directions on last page)

In April, Dave Wood will present his printer interface with the Sinclair-Timex computer. The printer is the Okidata Microline 92, a full-width dot matrix printer which expects a Centronics-type interface. He uses the CAI/O interface board from CAI Instruments. Dave will describe the hardware and software required for the interface. In addition, Dave will describe how to get listings of BASIC programs with printers that can't utilize the LLIST capability of the computer.

Sue Mahoney will demonstrate "Ator the ABC Gator," a new software package from Timex. Ator was designed to teach ABCs to preschoolers. It introduces a new technique of combining the audio from the tape recorder with the operation of the computer.

Following the presentations, we plan to break into special interest groups to discuss topics of interest. Tentatively, an advanced and a beginner's group are planned.

At the May meeting which will be held on the third Wednesday of the month (May 18), Dan Roy will unveil an add-on color capability for the TS-1000 he developed. The product will be called ColorSin 81. It provides color, sound, and joy stick with no modification to the Sinclair-Timex computer. Also, Dan will demonstrate the CAI stringy floppy mass storage device, which he says was essential for the development of ColorSin 81.

If you have items to discuss at a future meeting or suggestions for presentations, contact Sue or Cliff.

HIGHLIGHTS OF THE MARCH MEETING

The first item at the last meeting was Bob Masters' review of Henry April's keyboard. The keyboard, sold by Henry's company, E-Z Key, sells for $84.95 plus shipping and handling. (BCS members can save shipping and handling charges by ordering the keyboard from Henry and picking it up at the meeting.) Bob gave a brief review of Henry's development process, one which Henry shared with us over the past year. Henry first designed the keyboard features, and then attempted to get a keyboard with a style and feel which he was confortable with. Bob likes the keyboard and the personalized service Henry has afforded BCS members. See a review article by Bob in the March issue of Syntax.

Next, Will Stackman reviewed MCODER, an integer compiler (see page 3). Will explained a compiler is a computer program which translates programs into machine language. The Sinclair-Timex BASIC is an interpreter. Compiled programs run much faster that interpreted programs. Also, Peter Nichols gave a brief demonstration of another integer compiler sold by Bob Berch, 19 Daques Street, Rochester, New York 14620. Both compilers appeared relatively easy to use. The compiled code was impressively fast. It was stated that Non-Trival Solutions, P.O. Box 2941, Amarillo, Texas 79105 also sells an integer compiler.

Larry Becker gave a very interesting demonstration of an intelligent laboratory station terminal he has developed. Larry is on sabbatical leave from Hiriam College in Northeast Ohio. He developed the system while at the Technical Education Research Centers (TERC) in Cambridge. His terminal consists of the Timex TS-1000 computer plus an additional card of electronics. The system allows up to 16 TS-1000 terminals to interface with a single central Apple computer. The terminal has parallel input/output and an analog-to-digital interface. A digital-to-analog interface is also planned. Data can be collected and displayed by the TS-1000 terminal. In addition, data can be automatically relayed to the central computer. At the central computer, the data can be stored on floppy disk or displayed on high-resolution color monitors. The central computer allows student's data to be processed, averaged, graded, and so on.

In Larry's demonstration, data on the motion of a pendulum was observed. He mounted a potentiometer with a pendulum attached on the side of a table. After setting parameters for voltage levels and sampling time (this was easy with the menu driven program he wrote), the rotational motion of the pendulum versus time was sampled. A beautiful low-resolution graph of a sine wave was plotted by the TS-1000 computer. Larry then changed a few parameters and the 500 sample points were almost instantaneously transferred to and displayed on the Apple computer.

The board Larry designed to interface the Timex computer uses approximately ten chips. The main chips are an 8255 parallel input/output chip, a 0809 analog-to-digital converter, and a 75117 tristate differential serial buffer/transceiver. The interface at the Apple was just another 75117 wired to the paddle port! Larry currently operates the interface at 20 kilobaud, over a reversible 3-wire line. With multiple terminals, a polling protocol is envisioned. Machine language programs in both the Apple and Timex are employed to operate the interface. Larry says the target price for the interface board is 50 to 100 dollars. The capability to download programs may be added.

REVIEW OF MCODER, AN INTEGER COMPILER by Will Stackman

MCODER is an integer compiler program for the Sinclair-Timex computer. It runs on machines with the 8 K ROM and at least 16 K RAM. MCODER is sold by Personal Software Services, Coventry, England. It is available locally from Intercomputer, Inc., 179 Newbury Street, Boston, Massachusetts 02116 for $18.95 including shipping.

The MCODER program occupies a little over 2 K of RAM. As a compiler, it takes a program you write in a modified BASIC language and converts it to an efficient machine language subroutine. This machine language subroutine is then used in an ordinary BASIC program. The machine language program is invoked by executing a USR call to location 18823. The 2 K MCODER program must remain in memory, as it contains subroutines used by the compiled program.

The MCODER BASIC, the modified BASIC in which you must write programs to be compiled, allows only simple arithmetic (+,-,*, and / with no SQR, PI, LN, or trig functions), using only integer (whole) numbers. Boolean operators (AND, OR, NOT) are not available. Only 26 single letter numeric variables are allowed. There is only one array. And there are no string ($) variables at all. Input is through the use of CODE INKEY, which decodes single keystrokes.

The MCODER program is easy to use. However, programs tend to be hard to debug because of the differences between the MCODER BASIC and the one we know. For example, the RND function returns an integer value between zero and 32767, instead of a floating point number between zero and one. Another example is that a FOR NEXT loop from 1 TO 3000 is required to allow enough time to get your finger off an input key. There is an error flag, but no error codes which would make debugging straightforward.

MCODER comes with "migi" (that's smaller than "micro," which is to say, minimal) documentation, which is by far its weakest feature. Commands are explained alphabetically, and some information must be inferred. For example, although not mentioned, you can recompile routines. Each time you do, however, a new REM is added which can't easily be removed. Also, the one example in the instructions is wrong. The screen-filling PLOT example must use 63 for width and 40 for height, not 100 and 50, and the divisor for PAUSE is 60, not 50, as in the United Kingdom.

All in all, as the cheapest and shortest compiler available, MCODER is an interesting piece of software. Its immediate uses seem limited. The most interesting feature, an elaborate SCROLL, might be useful for a long series of complex results. But the integer math makes formulating such results (for example, statistical tables) quite impractical.

DID YOU KNOW OR CARE?

You can discover some interesting (?) facts if you do some PEEKing into the ROM. For example, zero appears more times than any other byte (single byte numbers must be between 0 and 255). It occurs 321 times, with 169 of these occurrences (over half) in the part of the ROM which contains the pixel representations of characters. The next most popular number is 205 (CD hex) with 304 occurrences. This number corresponds to the CALL instruction, the machine language equivalent of GOSUB. The number 153 is the only one which does not appear in the ROM. The number 29 occurs 29 times.

PROGRAM by Jack Hodgson

The program below does nothing very useful, but watching it amuses the hell out of its author. The display looks best when run in fast mode. Once you've figured out how it does what it does (which isn't too tough), enter other values for A$.

```
 10 LET D=0
 20 LET C=0
100 LET A$="13512812800413512813000012800700100012
12912900512813000400012813112900500212812800100500500500 5"
190 LET E=1
200 FOR B=1 TO 8
300 PRINT AT D,C+B-1;CHR$ VAL A$(E TO E+2)
350 LET E=E+3
400 NEXT B
403 IF E>94 THEN GOTO 450
404 LET D=D+1
405 GOTO 200
450 LET C=C+5
460 IF C>24 THEN LET C=0
500 PAUSE 60
698 LET D=0
699 CLS
700 GOTO 190
```

MORE FRESH SQUEEZED DATA 16 K RAM

Last month we featured an article entitled "Fresh Squeezed Computer Data." In the article, two subroutines were given. The first subroutine, beginning at line 200, compressed input data I$ into A$; the second, at line 800, expanded compressed data in I$ into A$. In this article, we illustrate how these routines may be used.

Consider a program to store names and telephone numbers. Let's allocate 20 characters for each name and 10 digits for each telephone number. With 30 bytes for each entry, only 34 names and numbers could be stored per 1 K RAM storage. Using the data compression scheme, we could compress the data into 20 bytes per entry. This would increase the number of entries by 50 percent. The only change necessary to the program published last month is to line 2 to LET B=20.

It is possible to get even more data compression by using more information about names and telephone numbers. We know that names require only 27 different symbols (the alphabet and a blank); and telephone numbers only require eleven different symbols (the numbers 0,1,...,9, and a blank). Using the formula -INT (-N*(LN M/LN 2)/8), where N is the length of the string and M is the length of the alphabet (LEN M$), it is possible to calculate the number of bytes required. The 20-character name (N=20, M=27) requires 12 bytes and the 10 digit telephone number (N=10, M=11) requires 5 bytes, or a total of 17 bytes per entry. Thus, we get 60 entries per 1 K of storage or a 75 percent increase over the original capacity.

YET MORE ...

Below is the code for the data compression technique described on the last page. If you start at line 5, the data is compressed into the string D$. Starting at line 500 the Dth data entry, where D is the input parameter, is decompressed.

```
  5 REM  D$=SAVE DATA
 10 LET D$=""
 15 LET T=256
 25 PRINT "ENTER NAME"
 30 INPUT I$
 32 IF  LEN I$=0 THEN  GOTO 25
 35 GOSUB 100
 40 GOSUB 200
 45 LET D$=D$+A$( TO B)
 55 PRINT "ENTER PHONE NUMBER"
 60 INPUT I$
 62 IF  LEN I$=0 THEN  GOTO 55
 65 GOSUB 150
 70 GOSUB 200
 75 LET D$=D$+A$( TO B)
 80 GOTO 25

100 REM NAME SETUP
110 LET M$="ABCDEFGHIJKLMN
OPQRSTUVWXYZ "
125 LET N=20
130 GOTO 600
150 REM NUMBER SETUP
160 LET M$="0123456789 "
175 LET N=10
180 GOTO 600

600 REM COMPLETE SETUP
610 LET M=LEN M$
620 LET B=-INT (-N*(LN M/LN 2)/8)
630 DIM K(B)
640 RETURN

500 REM D=ENTRY  N$=NAME  P$=PHONE
505 PRINT "INPUT ENTRY NUMBER"
510 INPUT D
515 LET I$=D$(D*17-16 TO D*17-5)
520 GOSUB 100
525 GOSUB 800
530 LET N$=A$
535 LET I$=D$(D*17-4 TO D*17)
540 GOSUB 150
545 GOSUB 800
550 LET P$=A$
555 PRINT N$;" ";P$
560 GOTO 505
```

The example illustrates how one uses a data compression technique. Even further data compression could be achieved if variable length records were allowed.

USER GROUP LIBRARY

In February, Chuck Durang on behalf of Reston Publishing Company donated five books to the user group library. They are:

Mastering Machine Code on Your ZX-81 by Toni Baker

Making the Most of Your ZX-81 by Tim Hartnell

49 Explosive Games for the ZX-81 by Tim Hartnell

The ZX-81 Pocket Book by Trevor Toms

Fifty 1 K / 2 K Games for the ZX-81 and TS-1000 by Allistair Gourlay, James Walsh, and Paul Holmes

In addition, we have Byteing Deeper into Your TS-1000 by Mark Harrison, which was donated by Tom Bell of Addison-Wesley Publishing Company. To borrow these books see a member of the library committee at the next meeting.

NEW BOOKS

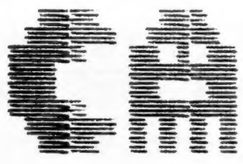
Updating the list of books available in the Boston area (see the February newsletter), R. F. Valle reports having found the following:

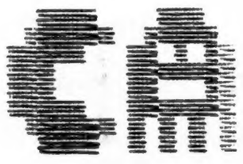
Charlton, Mark. The Gateway Guide to the ZX-81 and ZX-80, More Than 70 Programs. Morris Plains, New Jersey: Creative Computing Press, 1981.

Hartnell, Tim. Getting Acquainted With Your ZX-81 (Third Edition), More Than 80 Programs. Morris Plains, New Jersey: Creative computing Press, 1982.

Page, Edward. 37 Timex 1000 Sinclair ZX-81 Programs for Home, School, Office. Woodsboro, Maryland: ARCsoft Publishers, 1983.

Page, Edward. 101 Timex 1000/Sinclair ZX-81 Programming Tips and Tricks. Woodsboro, Maryland: ARCsoft Publishers, 1983.



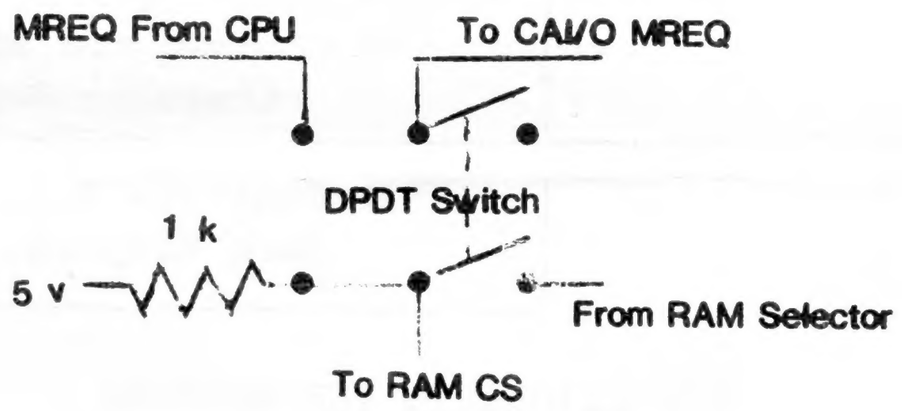


TIMEX NEWS

Sue Mahoney informed the group that the Timex printer is being shipped. She also said the Timex´ computer newsletter is being prepared. It will be called Ramblings.

TWO MEMORIES IN ONE SPACE by Dave Wood

Many of the new Timex-Sinclair add-ons make use of the unused address space between the ROM and RAM (memory addresses 8192 through 16383). However, if you buy devices from more than one vendor, this "unused" memory space can soon become over-used. Following is a quick-and-dirty solution I use to share (or multiplex) the address space for a CAI/O interface which drives my printer and my home-made static memory (used to store machine code). The only part required is a simple double pole double pole (DPDT) switchand a small resistor. The switch selects which of the two equipments use of the memory space. Obviously, both cannot be selected at the same time.



The activation of a device in 8-to-16 K memory area depends upon the voltage states of some of the lines which come out on the back edge connector. It is usually a logical zero (zero volts) on one specific line. In the case of the CAI/O interface, the entire interface box is activated by the MREQ line (pin 14 on the top edge connector) going to logical zero. If this signal doesn´t get to the CAI/O, the interface is inoperative -- that is, it´s internal EPROMs are not used. In the case of my memory, the RAM chip select line (CS) goes to logical zero when the proper address is decoded. If the CS is held at logical one (+5 volts), then my memory is inoperative (but it still remembers all that is in it). Above is the wiring of the DPDT switch. I made the MREQ line connections on an interface connector between the computer and the CAI/O. Thus, no modification to the computer or to the CAI/O were required.

ADVERTISING

In the May issue of this newsletter, we plan to begin publishing paid advertising. We are doing this for two reasons. First, there are costs involved. The BCS has been very supportive, actually unquestioning. But if we continue to grow, as we plan, we will soon be more of a burden than it is fair to expect the BCS to bear. The second reason is that advertising will give our readers another forum for examining the products and services that are available to them.

We do, however, want to make some important points. We shouldn´t even need to say that editorial (reviews ...) will be independent of advertisers. This publication in no way endorses products advertised, nor disparages ones that are not. In addition, only a small portion of the newsletter will be devoted to ads. Advertising space will be, within certain public guidelines, assigned on a first-come, first-served basis. We have given much thought toward creating a policy that will showcase advertisers that are of value to our readers and screen out others. These policies will evolve over time. We seriously solicit your comments on them (as well as other aspects of this newsletter and user group). Next month we take the plunge. We hope to use this opportunity to better serve you. It may take us a couple of issues to get the hang of it. Your help is welcome and your forbearance is begged and, oh yes, if you want to buy an ad, please call.

FOR MORE INFORMATION

Sue Mahoney, Director of the Sinclair-Timex User Group
c/o The Boston Computer Society or call (203) 573-5816.

Cliff Danielson, Newsletter Editor
14 Davis Road, Chelmsford, MA 01824, (617) 256-4638.

John Kemeny, Contributing Editor & Correspondent With Other User Groups
284 Great Road, Apt. D5, Acton, MA 01720.

Library Committee: Beth Elloitt, Sean O´Rahilly, and Bob Sanchez.

ADVERTISING INFORMATION
Computer Related Products and Services Only
Open Rate: $40 per Quarter Page
For Complete Rate Card and Discount Information Contact
Jack Hodgson, P.O. Box 526, Cambridge, MA 02238

DIRECTIONS TO THE MEETING

The Sinclair-Timex User Group meets in the Large Science Auditorium (Room 8/2/009) of the University of Massachussets of Boston, Harbor Campus. The Harbor Campus is only 3 miles from downtown Boston and easily accessible by public and private transportation. From the north or west, take the Southeast Expressway to Exit 17. Turn left onto Columbia Road. Enter the rotary and take the first right (Morrissey Boulevard). Bear right on the traffic island, following UMass/Boston sign. Turn left into the Campus. From the south, take Morrissey Boulevard northward to the campus. On the MBTA, take the Red Line (Ashmont Train) to Columbia Station. Transfer to the free University shuttlebus in the T parking lot.

IMPORTANT NOTICE ! If the mailing label on this newsletter is handwritten, then you are not on the mailing list of the Sinclair-Timex User Group. You need to either join the BCS or, if you are a BCS member, contact Mary McCann in the BCS office to be added to the Sinclair-Timex mailing list.

The Boston Computer Society

Three Center Plaza
Boston, MA 02108
617-367-8080

Nonprofit
U.S. Postage
Paid
Permit 1138
Boston, MA

A.F. Stanonis
Circle Chess
PO Box 63
Des Plaines, IL 60017