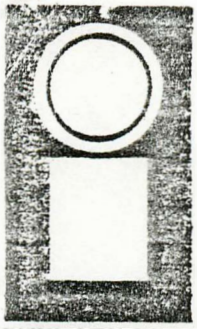
STORE COPY!

L ARNOLD



INTERACTION

NEWSLETTER
DETROIT
INTERACT
GROUP

DATE MARCH, 1980 VOLUME NO. I, no.1

WELCOME TO INTERACTION, the Interact owner's newsletter. It is published as a cooperative effort of the Detroit Interact Group and you, hopefully! The contents of the newsletter will depend on what the readers help contribute to our efforts. If you have a special interest or need, write us about it or better yet send us an article on it and Interaction will try to provide the resource to develop your interest or need.

This is the only free issue you will receive. We are not interested in making a profit, but we cannot continually spend our own money to keep the newsletter going. The newsletter subscription for 1980 is ten dollars. For that ten dollars you will receive all the newsletters published this year. We will have a minimum of six issues per year. We plan to have more issues per year, but the number will depend on reader contributions and staff time.

Interaction cannot pay for articles but we will try to print any articles readers send us that others might like to read. This includes items on computers and computing in general, information on other related fields of interest (e.g., a number of members are ham radio operators), and we will even print personal opinions. If we can locate the proper people we would like to print a history of Interact's development and downfall.

Interaction will have and would like to have articles on software development (hidden commands and functions!), programming tutorials, graphics and animation, software bugs, hardware bugs, hardware development (such as external control), feasibility of memory expansion (outboard S-100 bus ?), machine improvements (new keyboard? with real keys.), and useful and fun programs. We will also have a software exchange column and a program development column. This is for when you're having trouble writing or running a program, write us and maybe someone can help you de-bug your program.

We will not sell software but we will supply information on who will exchange programs. If you want to sell your software, hardware, or accessories there will be an advertisement charge.

The ad rates are fifty cents per typed line for individual subscribers and ten dollars per inch for commercial ads. please send typed copy or photo ready copy on standard sheet of paper. Interaction cannot guarantee any adverstisement but we will not publish any ad known to be false or fraudulent.

SUBSCRIPTION INFORMATION DETAILS ON PAGE 7

POKEing AROUND

POKE is a command used to place a value in a memory location. POKE 19215, 25 would place the number 25 in memory location 19215. The Interact Owner's Manual made no reference to this command and many owners were told by Interact that it could not be done. An attempt to perform a POKE would result in a SN error (syntax) or BS error (subscript out of range). What Interact would not tell us is that an initializing POKE must first be given. This is POKE 19215, 25 . After this POKE almost any portion of memory can be manipulated directly.

A POKE can only be used in RAM memory, not in ROM memory. ROM stands for Read Only Memory which means it can be read but not written or manipulated. Caution must also be used when POKEing RAM as Basic is stored in RAM memory and an incorrect POKE can destroy part or all of your operating, taking your program with it. Basic must then be reloaded.

The complement of POKE is PEEK. PEEK allows us to read data directly from RAM memory. PEEK is a function and is used in the form A=PEEK(X) , where X is a valid and allowable memory location. A will be returned as an integer between 0 and 255. This is also the maximum value that can used in a POKE command. This is because each memory location is a "byte". A byte is made up of 8 bits of binary information and the largest number in binary using 8 bits is 11111111_2 which equals 255_{10} in decimal.

SOFTWARE BUG

A useful graphics symbol is CHR$(1). When used in the form OUTPUT CHR$(1),x,y,c , it will place a 5x5 spot anywhere on the screen. But it doesn't always work! Closer examination of the problem shows it fails to appear upon initial loading of Basic and after a reset. There is a solution, simply hit the backspace key before typing RUN for a program which uses the CHR$(n) function. You can also provide a line early in your program which produces the same effect. This would be for example,
5 PRINT CHR$(8) . This makes an automatic backspace and eliminates any further problems.

HARDWARE BUG

If you look inside the tape compartment of the Interact tape unit you can see 2 tape heads. One of them is an erase head but it doesn't erase, it is nonfunctional! If you open the machine you can see there are no connections to the head. So how does an Interact erase previously recorded tapes? It doesn't. In some cases this may cause problems such as extra data and failure to load. This may be particularly bad when recorded on one machine and played on another. So rather than losing a valuable program because the tape wasn't blank, erase a tape if you suspect something might be recorded on it. This can be done on a bulk eraser or a regular cassette recorder with the microphone disconnected. A future project of Interaction will the addition of an erase circuit to your Interact.

RING A BELL

Typing a Control/G causes the bell to sound. This is the same as CHR$(7) . Typing Control/G in a program does not work during a RUN but [illegible] during a LIST to call attention to a line.

POKEing THE SCREEN

Graphics can be accomplished quicker by directly POKEing screen memory. After initializing with a POKE 19215, 25 , a POKE between location 16384 and 18655 will display a 4 pixel bar on the screen.

POKE 16384, n is the top left corner
POKE 16405, n is the top right corner
POKE 18624, n is the bottom left corner
POKE 18655, n is the bottom right corner

Increasing the location value by 1 will create a horizontal line.
Increasing the location value by 32 will create a vertical line.

The color values of the 4 pixels will depend on the value POKEed into memory. Using color position numbers (0,1,2,3), you can pick colors for each of the 4 pixels.

Where C_1 = first pixel color
C_2 = second pixel color
C_3 = third pixel color
C_4 = fourth pixel color

for POKE XXXXX,N $N = C_1+4*C_2+16*C_3+64*C_4$
Thus for all pixels being color 1 $N=1+4*1+16*1+64*1 = 85$

Use the chart on the next page to get any possible combination

DON'T PANIC

Interaction wants to warn all Interact owners not to panic. Although Interact is out of business, through legitimate sources all software and accessories are still available. Most items are available at original cost or less. Do not pay higher prices than Interact charged. Upgrades to 16K are not available yet but should be forthcoming. Upgrades to printer interface will take our independent development and Basic In ROM will most likely never be available. Interaction will be providing information in the future on ROM listings and assembly language ROM's.

Software tapes, controllers, new key caps, additional program books, schematics and service information, and additional Interacts and parts are currently available from 2 sources. Write or call them for the availibility of specific items.

MICRO VIDEO
~~275 E. Liberty Plaza~~ P.O.Box7357
Ann Arbor, MI 48107
(313) 996-0626

COMPUMART / NCE
1250 N. Main St.
Ann Arbor, MI 48107
(313) 994-3200

SCREEN POKE VALUES

Value 0 makes all 4 pixels equal to color position 0 that is the background color

0 = 0000
1 = 1000
2 = 2000
3 = 3000
4 = 0100
5 = 1100
6 = 2100
7 = 3100
8 = 0200
9 = 1200
10 = 2200
11 = 3200
12 = 0300
13 = 1300
14 = 2300
15 = 3300
16 = 0010
17 = 1010
18 = 2010
19 = 3010
20 = 0110
21 = 1110
22 = 2110
23 = 3110
24 = 0210
25 = 1210
26 = 2210
27 = 3210
28 = 0310
29 = 1310
30 = 2310
31 = 3310
32 = 0020
33 = 1020
34 = 2020
35 = 3020
36 = 0120
37 = 1120
38 = 2120
39 = 3120
40 = 0220
41 = 1220
42 = 2220
43 = 3220
44 = 0320
45 = 1320
46 = 2320
47 = 3320
48 = 0030
49 = 1030
50 = 2030
51 = 3030
52 = 0130
53 = 1130
54 = 2130
55 = 3130
56 = 0230
57 = 1230
58 = 2230
59 = 3230
60 = 0330
61 = 1330
62 = 2330
63 = 3330
64 = 0001
65 = 1001
66 = 2001
67 = 3001
68 = 0101
69 = 1101
70 = 2101
71 = 3101
72 = 0201
73 = 1201
74 = 2201
75 = 3201
76 = 0301
77 = 1301
78 = 2301
79 = 3301
80 = 0011
81 = 1011
82 = 2011
83 = 3011
84 = 0111
85 = 1111
86 = 2111
87 = 3111
88 = 0211
89 = 1211
90 = 2211
91 = 3211
92 = 0311
93 = 1311
94 = 2311
95 = 3311
96 = 0021
97 = 1021
98 = 2021
99 = 3021
100 = 0121
101 = 1121
102 = 2121
103 = 3121
104 = 0221
105 = 1221
106 = 2221
107 = 3221
108 = 0321
109 = 1321
110 = 2321
111 = 3321
112 = 0031
113 = 1031
114 = 2031
115 = 3031
116 = 0131
117 = 1131
118 = 2131
119 = 3131
120 = 0231
121 = 1231
122 = 2231
123 = 3231
124 = 0331
125 = 1331
126 = 2331
127 = 3331
128 = 0002
129 = 1002
130 = 2002
131 = 3002
132 = 0102
133 = 1102
134 = 2102
135 = 3102
136 = 0202
137 = 1202
138 = 2202
139 = 3202
140 = 0302
141 = 1302
142 = 2302
143 = 3302
144 = 0012
145 = 1012
146 = 2012
147 = 3012
148 = 0112
149 = 1112
150 = 2112
151 = 3112
152 = 0212
153 = 1212
154 = 2212
155 = 3212
156 = 0312
157 = 1312
158 = 2312
159 = 3312
160 = 0022
161 = 1022
162 = 2022
163 = 3022
164 = 0122
165 = 1122
166 = 2122
167 = 3122
168 = 0222
169 = 1222
170 = 2222
171 = 3222
172 = 0322
173 = 1322
174 = 2322
175 = 3322
176 = 0032
177 = 1032
178 = 2032
179 = 3032
180 = 0132
181 = 1132
182 = 2132
183 = 3132
184 = 0232
185 = 1232
186 = 2232
187 = 3232
188 = 0332
189 = 1332
190 = 2332
191 = 3332
192 = 0003
193 = 1003
194 = 2003
195 = 3003
196 = 0103
197 = 1103
198 = 2103
199 = 3103
200 = 0203
201 = 1203
202 = 2203
203 = 3203
204 = 0303
205 = 1303
206 = 2303
207 = 3303
208 = 0013
209 = 1013
210 = 2013
211 = 3013
212 = 0113
213 = 1113
214 = 2113
215 = 3113
216 = 0213
217 = 1213
218 = 2213
219 = 3213
220 = 0313
221 = 1313
222 = 2313
223 = 3313
224 = 0023
225 = 1023
226 = 2023
227 = 3023
228 = 0123
229 = 1123
230 = 2123
231 = 3123
232 = 0223
233 = 1223
234 = 2223
235 = 3223
236 = 0323
237 = 1323
238 = 2323
239 = 3323
240 = 0033
241 = 1033
242 = 2033
243 = 3033
244 = 0133
245 = 1133
246 = 2133
247 = 3133
248 = 0233
249 = 1233
250 = 2233
251 = 3233
252 = 0333
253 = 1333
254 = 2333
255 = 3333

SIMON

A pattern matching game. Move the joystick and repeat the sequence the computer randomly generates. The sequence increases from 1 to 10 and if done correctly plays the Star Wars theme. the program uses a machine language subroutine to draw the display. USR subroutines will explored and discussed in future issues.

```
 1  REM SIMON INTERACT LEVEL II BASIC
 2  REM WRITTEN BY MARV LONG AND RUSS PATTEN -  2/80
 5  COLOR 2,3,4,1
10  CLS
20  POKE 19215,25
30  FOR X = 23808 TO 23813
40  READ Y : POKE X,Y
50  NEXT X
60  DATA 1, 6, 93, 195, 162, 5
70  POKE 19473, 0 : POKE 19474, 93
75  FOR A = 1 TO 3
80  FOR X = 23814 TO 23818
90  READ Y : POKE X,Y
100 NEXT X
105 B= USR(0)
110 NEXT A
120 DATA 39, 56, 1, 0, 28
130 DATA 39, 56, 2, 39, 28
140 DATA 78, 31, 3, 0, 84
150 COLOR 0,0,0,0
160 FOR X = 1 TO 10
170 FOR Y = 1 TO 10
180 LET A(Y) = INT( 4*RND(1)+1) : NEXT Y
190 FOR F = 1 TO 10
195 FOR D = 1 TO 200 : NEXT D
200 FOR Y = 1 TO F
210 ON A(Y) GOSUB 1000, 1010, 1020, 1030
220 NEXT Y
230 FOR Y = 1 TO F
240 LET C = JOY(0)
250 IF C = 1 GOTO 2000
260 IF C = 4 GOTO 2100
270 IF C = 8 GOTO 2200
280 IF C = 2 GOTO 2300
```

SIMON CONT.

```
 290  GOTO 240
 300  NEXT Y
 310  GOTO 450
 400  COLOR 1, 2, 3, 4 : SOUND 3,48
 410  FOR Q = 1 TO 500 : NEXT Q
 420  SOUND 7,4096
 430  NEXT X
 435  IF X > 10 GOTO 920
 450  NEXT F
 460  GOSUB 2500
 910  NEXT X
 920       END
1000  COLOR 1,0,0,0 : TONE 168,150 : GOTO 1040
1010  COLOR 0,2,0,0 : TONE 131,192 : GOTO 1040
1020  COLOR 0,0,3,0 : TONE 110,229 : GOTO 1040
1030  COLOR 0,0,0,4 : TONE 80,315
1040  COLOR 0,0,0,0 : RETURN
2000  GOSUB 1000
2010  IF A(Y) = 1 GOTO 300
2020  GOTO 400
2100  GOSUB 1010
2110  IF A(Y) = 2 GOTO 300
2120  GOTO 400
2200  GOSUB 1020
2210  IF A(Y) = 3 GOTO300
2220  GOTO 400
2300  GOSUB 1030
2320  GOTO 400
2500  COLOR 2,3,4,1
2510  FOR E = 1 TO 16
2520  READ G : READ H
2525  TONE G,H
2530  NEXT E
2535  RESTORE
2540  FOR S = 1 TO 21 : READ S : NEXT S
2545  RETURN
2550  DATA 168,75,110,115,124,51,131,48,148,43,80
2560  DATA 156,110,115,124,51,131,48,148,43,80,156
2570  DATA 110,115,124,51,131,48,124,51,148,170
2580  RETURN
```

See Vol. I No. 2

SUBSCRIPTION INFORMATION

To subscribe to Interaction for 1980, please send a check for $10.00 to

INTERACTION
Stephen Cook
15356 Prevost
Detroit, MI 48227

Please make check PAYABLE TO : STEPHEN COOK
as we do not have a newsletter bank account yet.

Also any articles or letters should be sent to the same address.

Complete as much of this questionaire as you wish and please return whether you decide to subscribe or not. This information will help us produce a better newsletter and locate other Interact owners.

NAME _____

ADDRESS _____

CITY _____ STATE _____ ZIP _____

PHONE _____

HOW MANY INTERACTS DO YOU OWN ? 16K units ____ 8K units _____

Serial Nos. _____ _____

What computer topics interest you most?

What would you like to see in Interaction?

What article was most helpful?

What article was least interesting?

What other computer applications of the Interact would you like to see developed?

DAVE YOUNG

INTERACTION

NEWSLETTER
DETROIT
INTERACT
GROUP

DATE MAY, 1980 VOLUME NO. I, no2

HI! Interaction number 2 has arrived. From the returned questionaires I've received, I see that the newsletters are arriving in good condition. If anyone has problems with the condition of your newsletters when they arrive, write me and we'll try to get them to you in better shape. Also if you receive a mutilated newsletter write and we'll send you a new copy.

On writing and calling, I did not print any subscribers' addresses in this issue but I would like to include addresses to encourage correspondence. If anyone has objections to giving out or printing your address, please let me know. I will not give out phone numbers but will leave that up to individual decision.

If someone wants to contact me you can call me (Steve Cook) at (313) 272-7594, evenings. I am out quite often so a better phone number might be Marv Long's, who has a phone answering machine. His number is (313) 546-6444. Neither one of us can afford to return long distance calls but if you leave a message or question, we will write back to you.

All of us involved in the newsletter are new at this and things are developing so quickly following Interact's collapse we are having trouble keeping organized. We have a lot of information that is not very useful or too voluminous to print. We cannot print, in a practical sense, ROM listings that would simply look like pages and pages of numbers that most of us would be unable to interpret.

We have yet to organize a software exchange for Basic programs on tape. Among other problems, no one wants to handle all the copying on their machine, wearing out difficult to replace parts. Also how do we cover the costs (tapes, mailing, etc.) involved? Do we require submission of a program before we give one in exchange? What kind of programs should we exchange? If anyone has any suggestions or comments please write or call.

EDU-BASIC BUG

Early production Edu-Basic tapes had a line numbering defect. You could not enter any line number that is a multiple of 256 (such as 256, 512, 768, 1024 and so on). If you entered a line number with one of these numbers it was actually entered as a line number 256 less than what you typed in. Thus if you typed in the following program:

```
256   PRINT "1"
512   PRINT "2"
768   PRINT "3"
```

and then asked for a LIST what you would get is:

```
  0   PRINT "1"
256   PRINT "2"
512   PRINT "3"
```

When I complained to Interact about this they acknowledged the error

EDU-BASIC BUG cont.,

and told me to mail back the tape and they would exchange it, which they did. That alternative is no longer available since Interact's closing. So you will either have to purchase a new tape or work around the defect, if you have the need to use Edu-Basic. If you have an older Interact or if it sat in the dealer's stock for a long time (I only purchased my Interact last November and it had the bad tape.), check your Edu-Basic before you use it to avoid this bug.

COMPUTE-A-COLOR BUG?

Bill Adams of Palm Bay, FL wrote me telling me about a problem with Compute-A-Color. He says the Full Screen and Erase Screen keys do not perform their functions, but the keys immediately to the right of these do perform the respective functions. My unit and the several other units in the Detroit area I've checked do work correctly. If anyone else has this problem please write so we can determine if the defect is in the program tape, keyboard overlay or just a fluke in Bill's Compute-A-Color.

PRODUCT RELEASE BULLETIN

RE: CHM-100 RS232C Port
DATE: April 25, 1980

JRA Systems announces a new support product for the INTERACT Personal Computer. The model number CHM-100 RS232C port which is fully compatible with INTERACT's popular 'RS232 BASIC' and hardware. This product will be available in late May or early June 1980. The product when combined with 'RS232 BASIC' and a printer will provide the user with an 'LPRINT' command for directing program output to a RS232 serial printer; and an 'LLIST' command for obtaining hard copy listings of BASIC programs. The unit is a dual port with one port having full handshaking and the other having a three wire send, receive setup.

With the appropriate software the port could also be used to connect to a MODEM for telephone communications between the INTERACT and other computers, etc.

For the experimenters there will be a wire wrap area on board for implementing your own ideas. The port will be available in assembled and tested or kit forms. The price will be around the $100.00 mark although pricing is not completed at this time.

For further inquiries write to :

J R A Systems
Box 555
St. Clair Shores, MI 48080

SIMON CORRECTION

The Simon program in the first issue was missing a line. Thanks to Alan Bishop of Grand Rapids, MI for pointing it out to me. In the future I'll try to be more careful when copying listings.

ADD THIS LINE: 2310 IF A (Y) = 4 GOTO 300

Also Line 2545 is extranous and can be deleted.

THE INTERACTORY

By Marv Long

In our first issue we ran a game, Simon, using a USR command to draw the color blocks very quickly. We promised at that time to start explaining the use of this command and that's exactly what we're going to do.

USR A command to allow a jump to a machine language routine and a way back to Basic.

First a trip down memory lane - in Hexidecimal:

4000 - 7FFF	RAM
4000 - 49FF	SCREEN
4A00 - 4C00	SPARE
4C00 - 5F7F	USEABLE RAM
5F80 - 5FFF	STACK & VARIABLES

Level II Basic starts writing at 4C00 approximately and ends at 5F7F. So when we write a machine language subroutine, we'll put it up towards 5F7F so our Basic program won't write over it. POKEing it after 5F7F will clobber Basic. If you've read this far and are lost, stick with it. You'll be able to draw the blocks even if you don't understand how. While you have your Interact busy drawing go pick up a copy of UNDERSTANDING DIGITAL COMPUTERS - from Radio Shack @ 3.95. Or pick up some other book that explains binary and hexidecimal number systems.

We're going to work in hex convention, but we have to POKE in decimal. We've included a program at the end of this column to make conversion easier.

LET'S DRAW A BLOCK

```
10  CLS          Clear screen
20  POKE 19215, 25          Initialize to allow pokes
```

We are going to write our routine at 5D00 hex. This location has to be poked into Basic so when we call USR it knows where to jump to. It is poked in two 8 bit words. Least significant first (00) then most significant (5D). 00 is 0 in decimal - 5D is 93 in decimal. 19473 and 19474 are the locations in Basic that make USR work.

```
30  POKE 19473, 0 : POKE 19474, 93
```

Our program is 6 bytes long, so we'll set our locations in memory. 5D00 = 23808 decimal.

```
40  FOR X = 23808 TO 23813
50  READ Y          Will read data we're poking in.
60  POKE X, Y       X is address from line 40.  Y is data from 80
70  NEXT X          Loop for incrementing X
```

Interactory cont..

We have two commands to give our machine.
01 LOAD IMMEDIATELY and 195 (C3 hex) JUMP
Also the address of of our routine + the length
(6 bytes) and we're calling a routine already
written in the Interact's ROM to draw blocks.
5A2 - 162,05 in decimal written least significant
and most significant bits as before at line 30.

```
 80  DATA 01, 06, 93, 195, 162, 05
```

The ROM routine 5A2 = 23814 requires 5 bytes
of information for the block parameters, so
we'll give X these locations.

```
 90  FOR X = 23814 TO 23818
100  READ Y          To read our new data table
110  POKE X, Y       X address from line 90   Y data from line 130
120  NEXT X          Loop
```

When we draw in machine language our Y axis is
different from Basic. 0 is the UPPER left hand
point on the screen and 77 is the LOWER left.
Our ROM routine wants the following:

1 - Vertical height in pixels
2 - Horizontal width in pixels
3 - Number for color
4 - Vertical position
5 - Horizontal position

```
130  DATA 40, 40, 3, 18, 36
140  B = USR(0)          This calls our routine.
```

That's it. If you don't fully understand it try it anyway.
Play with line 130. By changing the data, you can increase
the size, change the shape, color, and position of the block.

If you have any questions, comments, or suggestions, write me or INTERACTIC

MARV LONG
1661 COLLEGE ST.
FERNDALE, MI
48220

Writing in machine language is more difficult perhaps, than Basic
but it allows us tools to do more and faster things with the Interact.
In future issues we'll try drawing pictures and creating sounds not
possible in Basic.

NUMBER BASE CONVERSION

This program will convert one number system to another. Enter the base you're using, then the number you want converted. Then type in the new base and hit return. The computer will convert and print the number in the new base. Once you have setup a conversion, it is not necessary to re-enter the two bases until you wish to change them. Just hit a return. Happy counting.

```
100   REM BASE CONVERSION PROGRAM
105   REM ADAPTED FOR INTERACT
110   REM BY RUSS PATTEN
120   DIM W(20)
140   CLS
150   PRINT
200   INPUT "INPUT BASE"; B1
205   IF B1 = -1 THEN END
210   IF B1 < 2 OR B1 > 16 THEN 600
220   PRINT "INPUT NUMBER" : INPUT N1$
300   INPUT "OUTPUT BASE" ; B2
310   IF B2 < 2 OR B2 > 16 THEN 600
400   GOSUB 1000
410   PRINT "OUTPUT NUMBER =" : PRINT N2$
500   GOTO 150
600   PRINT "BASE NOT IN RANGE"
610   PRINT "OF 2 TO 16"
700   GOTO 150
1000  REM BASE ROUTINE ROUTINE
1010  REM DECIMAL
1020  LET L = LEN (N1$)
1030  LET D = 0
1040  LET P = 0
1050  FOR J = L TO 1 STEP -1
1060  K = ASC (MID$ (N1$, J, 1 ))
1070  IF K > 64 THEN K = K - 7
1080  K = K - 48
1090  IF K < B1 AND K > -1 THEN 1140
1100  PRINT "INVALID INPUT FOR BASE"
1110  PRINT B1
1120  N2$ = "???"
1130  RETURN
```

B1=10
N1$="A"
B2=16
GOSUB 1000

NUMBER BASE CONVERSION cont.,

```
1140   D = D + K * B1 ^ P
1150   P = P + 1
1160   NEXT J
1170   REM DECIMAL TO BASE
1175   LET W(1) = B2 : FOR U = 2 TO 20 : LET W(U) = W(U - 1) * B2 : NE
1180   H$ = "0123456789ABCDEF"
1190   N2$ = "  "
1200   P = LOG (D) / LOG (B2)
1205   LET P = INT (P)
1210   FOR J = P TO 0 STEP -1
1220   LET X = W(J)
1225   IF J = 0 THEN X = 1
1230   C = D / X
1235   LET C = INT (C)
1240   N2$ = N2$ + MID$ (H$, C + 1, 1)
1250   LET U = INT (D - C * X + .5)
1255   D = U
1260   NEXT
1270   RETURN
```

SAMPLE RUN

```
INPUT BASE?   16
INPUT NUMBER?   5D00
OUTPUT BASE?   10
OUTPUT NUMBER =   23808
```

To end program, type in a -1 for INPUT BASE.

REAL TIME FOR LEVEL II

by Vic Volkman
1131 Lakepointe
Grosse Pointe park, MI 48230

One of Interact's best kept secrets has been its real time clock, which may be one of the machine's most valuable features. The Model One has two clock registers that can be accessed through the PEEK and POKE commands; 24559, the one second clock and 24560, the one minute clock. The two timers open up new possibilities in time keeping and real time games. The one second clock increments up one every 1/60th of a second, while the one minute clock increments four seconds at a time. These timers, like stopwatches, are most effective if they are initialized before they are used (which means setting them to zero). After the value of the one second timer has reached 59, one whole second has passed. The timer should immediately be set back to zero, and after that a counter should be incremented to keep track of the elapsed time. The one second timer could also conceivably be stopped at 119 if the user desired to keep time in two second increments. The one minute timer is much more useful when longer time periods are needed because it does not require as much attention as the one second timer. Although it is only accurate to four seconds it can be left alone for 15 minutes before it has to be checked on. This would make it a good move timer for a checkers game or something like that. The following program demonstrates a one second timer.

```
10   POKE 19215, 25 : REM INITIALIZE PEEK AND POKE
20   CLS : REM CLEAR THE SCREEN
30   S = 0 : REM SET SECONDS = 0
40   POKE 24559, 0 : REM INITIALIZE CLOCK
50   IF PEEK (24559)>58 GOTO 70 : REM HAS ONE SECOND PASSED?
60   GOTO 50 : REM NO TIME CHANGE
70   OUTPUT S, 56, 38, 0 : REM CLEAR OLD TIME
75   S = S + 1 : REM INCREMENT SECONDS
80   OUTPUT S, 56,38, 3 : REM DISPLAY NEW TIME
90   GOTO 40 : REM START A NEW COUNT
```

A real time digital clock using the clock register is in the Basic Examples Booklet sold by Micro Video.

DADDY'S 'PUTER

One of the reasons I bought a computer (other than that I like expensive toys) is as an educational aid for my 3 year old daughter. I plan to write learning exercises and games for her as she grows. I am currently working on some number and letter recognition programs. When finished they may be in future issues.

Even now she likes to use Compute-A-Color and wants to type M's (her name is Margaret) when I'm using Basic. Since she likes to see the letters on the screen, I type in this short program whenever she wants to play with daddy's 'puter, as she calls it.

DADDY'S 'PUTER cont.,

```
10   CLS
20   A$ = INSTR$ (1)
30   IF A$ = "   " THEN 10
40   PRINT SPC (2) ; A$ ;
50   GOTO 20
```

Pressing the space bar erases the screen which she thinks is a great trick. If anyone is writing educational programs, please write and send a listing or tape copy (I'll return the tape with another program on it if you like) of what you're doing and we'll print the program or ideas. And of course we'll give you credit for anything we print.

DUMMY

We have already told you about PEEK and POKE and are continuing an explanation of USR. There is one more command Interact did not release. This is the last hidden function in Level II Basic. It was discovered by one member of the group (Jim Coon of Trenton, MI) even before someone had disassembled Basic. The function is DUMMY but we are not sure what it was written for or what exactly we can do with it. The function causes a machine reset and can be used either as a direct command or in a program. The format is PRINT DUMMY (N$, n) where N$ is any string variable or constant and n is a numerical variable or constant. A possible use of DUMMY is in place of an END statement, as in the Gravity game where I use it to end a game. This might be handy in a adventure type game where the last losing statement would be e.g., "YOU'RE DEAD!". Normally with an END statement the output on the screen would be

```
YOU'RE DEAD
OK
```

Using DUMMY you could first say YOU'RE DEAD, then after a pause loop to read that, PRINT DUMMY ("RESET", 0) would end the program much nicer. If you find any other uses for DUMMY please let us know and we'll print them in the newsletter.

GRAVITY

A 'real' time skill game using principles of gravity, acceleration, and inertia. The game is for one player using a joystick. Playing instructions are in the program. A convention I have adopted is that after a page of text is printed, the computer waits for a key input to continue displaying text. In Gravity, this is the I$=INSTR$(1) lines in among the instruction text. I think this is probably the best way to overcome the limited screen text display of the Interact. Also word and blank spacing is not screen accurate as it would be too difficult to show you where and how many blanks to insert in a program listing.

HOPE YOU ENJOY THE GAME.

```
  1  REM GRAVITY
  2  REM INTERACT BASIC
  3  REM BY S. COOK  -  3/80
 10  CLS : GOSUB 3000
 20  PRINT "WOULD YOU LIKE INSTRUCTIONS?"
 25  PRINT "(Y-N)"
 30  I$ = INSTR$ (1)
 35  IF I$ = "Y" THEN GOSUB 5000
 40  PRINT
 50  PRINT "MAXIMUM SCORE (0 - 9)?"
 60  M$ = INSTR$ (1)
 70  W = ASC (M$) - 48
 80  SKILL = 1
 90  CLS : GOSUB 1000
 95  IF JOY (0) = 0 THEN 95
 99  A = 0 : Y = 70
100  FOR X = 5 TO 110
120  PLOT X, Y, 1
125  SPEED = 2 * SKILL
130  FOR P = SPEED TO 20 : NEXT
135  PRINT CHR$ (7);
140  IF FIRE (0) = 0 THEN A = A - 2
160  A = A + 1
180  Y = Y - (A + SKILL)
200  IF Y < 15 THEN 2000
210  IF Y > 70 THEN 2000
500  NEXT
550  IF W = SKILL - 1 THEN 900
600  SKILL = SKILL + 1
700  GOTO 90
```

GRAVITY cont.,

```
 900   CLS
 910   PRINT : PRINT "HOPE YOU ENJOYED GRAVITY!"
 920   PRINT : PRINT "PLAY AGAIN (Y - N)?"
 930   I$ = INSTR$ (1)
 940   IF I$ = "Y" GOTO 10
 950   PRINT : PRINT "COME BACK AGAIN SOON!"
 960   FOR P = 1 TO 500 : NEXT
 990   PRINT DUMMY ("RESET", 0)
1000   REM SETUP
1010   FOR X = 0 TO 113
1020   PLOT X, 70, 2
1030   PLOT X, 15, 2
1040   NEXT
1050   S = SKILL - 1
1060   OUTPUT S, 80, 72, 3
1090   RETURN
2000   REM LOSE
2010   FOR F = 1 TO 4
2020   COLOR 3, 4, 7, 0
2030   SOUND 3, 3336
2040   COLOR 8, 2, 6, 3
2050   NEXT
2060   SOUND 7, 4096
2090   IF JOY (0) = 0 THEN 2090
2100   GOTO 90
3000   REM TITLE
3005   SOUND 0, 24844 : G = 0
3010   FOR Y = 72 TO 6 STEP -6
3020   G = NOT G
3025   X = 3 * G + 30
3030   OUTPUT "GRAVITY", X, Y, 1
3040   OUTPUT "GRAVITY", X, Y, 3
3050   OUTPUT "GRAVITY", X, Y, 0
3060   NEXT
3090   SOUND 7, 4096
3100   CLS : RETURN
5000   REM INSTRUCTIONS
5005   PRINT
```

pause to read above statement, the reset

Screen lines

score in upper right corner

LOOSE SEQUENCE

HEADER

GRAVITY cont.,

```
5010   PRINT "THE OBJECT OF GRAVITY IS TO KEEP THE DOT BE-"
5020   PRINT "TWEEN THE BLACK LINES. PRESS THE BUTTON TO STOP"
5030   PRINT "THE FALL, BUT IF YOU HOLD IT TOO LONG IT WILL HIT"
5040   PRINT "THE TOP LINE."
5045   I$ = INSTR$ (1) : PRINT
5050   PRINT "THE NUMBER IS THE SKILL LEVEL. THE LARGER IT IS, THE"
5060   PRINT "FASTER THE DOT WILL FALL. ENTER THE MAXIMUM SKILL"
5070   PRINT "YOU WISH TO TRY WHEN THE GAME BEGINS."
5075   PRINT : I$ = INSTR$ (1)
5080   PRINT "MOVE THE JOYSTICK TO START THE GAME AND TO RESET"
5090   PRINT "THE DOT IF YOU HIT A LINE." : PRINT "GOOD LUCK!"
5095   PRINT : I$ = INSTR$ (1)
5100   RETURN
```

INTERACTION

NEWSLETTER
DETROIT
INTERACT
GROUP

ATE JUNE - JULY, 1980 VOLUME NO. I, no. 3

INTERACT RS232 PORT IS HERE!

THIS WAS PRINTED FROM AN INTERACT USING ONE OF THE NEW RS232 SERIAL PORTS. WE USED AN ANADEX PRINTER BUT ANY RS232 PRINTER OR TERMINAL WILL WORK. WE ALSO COULD HAVE USED A MODEM TO ACCESS ANOTHER INTERACT- [illegible] -OR THE SOURCE OVER THE PHONE LINES. READ THIS ISSUE FOR FURTHER DETAILS

```
5 DIM A$(100)
10 LPRINTCHR$(14)
20 LPRINT"        INTERACT RS232 PORT IS HERE!"
30 LPRINT
35 LPRINTCHR$(15)
40 INPUTA$
50 LPRINTA$
60 GOTO 40
```

As you read above, RS232 is here. And not one RS232 unit but two different RS232 units. The first is the one mentioned in the last newsletter built by JRA Systems. It will be sold through Micro Video, see their ad for details. It is designed for use with RS232 Basic which has the commands LLIST and LPRINT. A prototype unit was used by Marv Long with a borrowed Anadex printer to produce the banner above. The pointers in RS232 Basic are different from LEVEL II Basic and LEVEL II programs must be run through RS232 EZEDIT to run in RS232 Basic.

The second unit does not require RS232 Basic but was designed for use with LEVEL II Basic. The Universial 80 Port is developed, built and sold by Slagh Systems Services. See their ad in this issue for more details. At press time we do not yet have a Universial 80 Port, but at the May meeting of the Detroit Interact Group, Mark Slagh demonstrated the modem capabilities of the Universial 80 when used with a terminal emulator program. He connected to several time sharing systems in the area without difficulty.

In the future, we will be testing both units and publishing the results of our findings. As we see it there are advantages and disadvantages to both units, but more on that later. Right now, no one in the Detroit Interact Group owns a printer or modem and until some of us obtain such we cannot do much testing of any RS232 Port. If you obtain either RS232 Port, write and tell us about your experiences.

NO, WE DON'T HAVE ALL THE ANSWERS!

Most Interact owners who have contacted us have had their units only a short time. Even Marv Long and I have had our Interacts only since last fall. The Detroit Interact Group started just in time to watch Interact fold. Most long time owners have sold their Interacts or stuck them in a closet in disgust because Interact did not release information on the machine before they collapsed. When Interact was in business, use of PEEK and POKE was unknown and even the sparse documentation that is now available did not exist then. When Interact closed, they didn't deliver all their records to us. All the information at Interact disappeared as suddenly as the company did. We have been able to obtain only bits and pieces from ex-employees. As far as we have been able to find out, very little documentation actually existed. So we don't have all the answers yet and we are not skilled enough to find and develop all the capabilities of the Interact. That's the purpose of the newsletter, For ALL Interact owners to explore and exchange information as we and you find out things about the Interact.

Everyone wants all the secrets of the ROM and wants to know how to copy machine language tapes. We're sorry but the secrets of the ROM are not that great and as we explore them and learn to use ROM subroutines we will be explaining them. Micro Video plans to publish an explanation or listing of ROM information later this year. Patience please. As for copying tapes, there is little incentive to produce new programs if the programmer can sell only one copy that will be reproduced ad infinitum. If you want to copy tapes the information can be found in the machine and software but we will not publish or distribute or accept ads for copy routines. We're sorry but we have some old-fashioned principles that you will have to accept. A programmer is an author/artist who we believe is entitled to reasonable reward for his or her work. If you think the prices are unreasonable don't buy but first compare prices with equivalent TRS-80 and other computers' software and utilities, the prices are very similar.

YOU CAN PEEK ROM

By using these initializing POKEs you can remove the limits on PEEK. Be sure that you use the correct values or you may lose BASIC.

```
POKE 30462, 195
POKE 30463, 25
POKE 30464, 119
```

After POKEing these locations, you can PEEK ROM, i.e., locations below 2049 and BASIC, locations above 26128

KALEIDOSCOPE

by Vic Volkman Grosse Pointe Park, MI

This program runs over 20 times faster than conventional programs of this type because it employs the block graphics character CHR$(1) to plot a total of 100 pixels or about 8% of the screen in a single loop. The plotting alternates between color #0 and a random color #1-3 to produce a dazzling four color display. The plotting algorithm is so quick that it has an almost hypnotic quality.

```
  5   REM KALEIDOSCOPE - 200 BYTES
 10   CLS : O$ = CHR$(1)
 20   X = INT (RND (1) * 16.6667) * 6
 30   Y = INT (RND (1) * 11) * 6 + 17
 50   OUTPUT O$, X, Y, C
 60   OUTPUT O$, 112 - X, Y, C
 70   OUTPUT O$, X, 77 - Y, C
 80   OUTPUT O$, 112 - X, 77 - Y, C
 90   IF C = 0 THEN C = INT (RND (1) * 3) + 1 : GOTO 20
100   C = 0 : GOTO 20
```

ROSES

by Geoff Hall San Jose, CA

As a modest first contribution to Interaction, I offer the program "ROSES". It is short and simple and produces rather pretty flower or 'petit-point' patterns. I suggest running for a few hours to observe how the flower 'blossoms'.

```
 10   REM "ROSES" BY GEOFF HALL / OCT '79
 20   REM INTERACT LEVEL II
 30   COLOR 0, 4, 1, 3 : C = 57.3 : N = 0 : M = M + 1 : IF M<2 THEN M=2
 40   CLS : FOR A = 0 TO 359 : SI = SIN (M * A / C)
 50   IF N<1 THEN R = 35 * SI : GOTO 70
 60   R = 35 * SQR ( ABS (SI)) : IF SI<0 THEN R = -R
 70   PLOT 56 + R * COS (A /C), 38 + R * SIN (A / C), 3 : NEXT
 80   FOR T = 0 TO 5000 : NEXT
 90   IF N<1 THEN N = 1 : GOTO 40
100   GOTO 30
```

PRODUCT RELEASE BULLETIN

SLAGH SYSTEM SERVICES ANNOUNCES THE RELEASE OF THE U80 LINE OF RS232 COMPATIBLE PORTS. THE U80D AND THE U80M ARE BOTH DESIGNED TO WORK WITH THE INTERACT MODEL ONE. THESE PORTS MAKE IT POSSIBLE TO CONNECT BI-DIRECTIONAL SERIAL RS232 DEVICES TO THE INTERACT, INCLUDING MODEMS, PRINTERS, AND OTHER INTERACTS(WITH U80 PORTS). THE U80 LINE FEATURES SOLDERLESS INSTALLATION IN THE INTERACT, AND IS SOFTWARE PROGRAMMABLE. MORE INFORMATION CAN BE OBTAINED FROM:

SLAGH SYSTEM SERVICES
6980 CARPENTER ROAD
YPSILANTI,MI 48197
(313) 434-0033

DRAWING A CIRCLE

The following are two methods of drawing a circle in Level II Basic. The first program is a cartesian plotting and the second a polar coordinate plotting routine. In both programs the circle size is dependent on R, XM, and YM. If XM ≠ YM then the circle becomes an ellipse. It can be either vertically or horizontally elongated depending whether YM or XM is larger. The step size determines the number of points in your figure; use a smaller value for a better defined figure. Personally, I prefer the polar method because the figure takes a definite shape much quicker.

```
  1   REM DRAWING A CIRCLE
  2   REM CARTESIAN PLOTTING
  3   REM BY S. COOK - 5/80
 10   CLS : WINDOW 12
 20   INPUT "ENTER RADIUS" ; R
 30   INPUT "ENTER SIZE OF STEP" ; S
 40   INPUT "X MULTIPLIER" ; XM
 50   INPUT "Y MULTIPLIER" ; YM
 60   CLS
 70   FOR X = -R TO R STEP S
 80   Y = SQR ( R * R - X*X )
 90   PLOT XM * X + 57 , YM * Y + 46 , 3
100   NEXT X
110   FOR X = R TO -R STEP -S
120   Y = -SQR ( R * R - X * X )
130   PLOT XM * X + 57 , YM * Y + 46 ,3
140   NEXT X
150   GOTO 20
```

```
  1   REM DRAWING A CIRCLE
  2   REM POLAR COORDINATES
  3   REM BY S. COOK - 5/80
 10   CLS : WINDOW 12
 20   INPUT "ENTER RADIUS" ; R
 30   INPUT "ENTER SIZE OF STEP" ; S
 40   INPUT "X MULTIPLIER" ; XM
 50   INPUT "Y MULTIPLIER" ; YM
 60   CLS
 70   FOR A = 0 TO 100 STEP S
 80   X = XM * R * SIN ( A ) + 57
 90   Y = YM * R * COS ( A ) + 45
100   PLOT X, Y, 3
110   NEXT A
120   GOTO 20
```

MEMORY DUMP

by Stephen Woodbridge Bedford, MA

This Level II Basic program will allow the user to dump the contents of any memory location. The program is set up to dump in hex or ascii which is useful in locating command tables and text in a program. This routine uses the USR(0) function to access and move the memory byte to be PEEKed at into the allowable range of addresses for Basic PEEKs.

The program has three entry points:

RUN - Set up and initialize PEEK address

RUN 200 - Dump in HEX bytes

RUN 300 - Dump in ASCII

```
  1  REM MEMORY DUMP - STEVE WOODBRIDGE
  2  CLEAR (100)
 10  CLS: POKE 19215, 25
 20  POKE 19473, 0                          } Initialize USR address
 30  POKE 19474, 93                         }
 40  INPUT "START LOC"; I
 50  POKE 23824, I - INT(I/256) * 256       } POKE 16 bit starting address
 60  POKE 23825, INT(I/256)                 }
 70  FOR I = 23808 TO 23823                 } POKE USR assembly code
 80  READ J : POKE I, J                     }
 90  NEXT I                                 }
100  DATA 245, 229, 42, 16, 93, 126, 50, 18
110  DATA 93, 35, 34, 16, 93, 225, 241, 201
120  END

200  X$ = "0123456789ABCDEF"
210  I = USR(0) : H = PEEK(23826)           } HEX Dump Loop
220  GO SUB 900 : PRINT H$ ; " " ;          }
230  GO TO 210

300  A$ = ""
310  FOR J = 0 TO 15
320  I = USR(0)                             } ASCII Dump Loop
330  A$ = A$ + CHR$(PEEK(23826))            }
340  NEXT J                                 }
350  PRINT A$ : GO TO 300
```

MEMORY DUMP (cont.)

```
900   H1 = INT(H/16) + 1
910   H$ = MID$(X$, H1, 1)
920   H1 = INT(H/16) * 16
930   H$ = H$ + MID$(X$, H - H1 + 1, 1)
940   RETURN
```

8 Bit DECIMAL
to HEX Conversion

This program was not put together with much sophistication to make it easier to see how it works. The USR routine might be sensitive to its regisyer being modified so I "bullet-proofed" the 8080 code it used.

ADDRESS	CODE	HEX	DECIMAL
23808	PUSH PSW	F5	245
	PUSH H	E5	229
	LHLD ADR	2A _ _	42, 16, 93
	MOV A,M	7E	126
	STAX TEMP	32 _ _	50, 18, 93
	INX H	23	35
	SHLD ADR	22 _ _	34, 16, 93
	POP H	E1	225
	POP PSW	F1	241
23823	RET	C9	201
23824	*ADR	Temp. Storage for address of the next	
23825		byte to be accessed	
23826	*TEMP	Byte to be PEEKed at	

This routine could probably be made more efficient if data (like the address) could be passed to the routine and the byte to be returned could be passed back to the Basic program. I plan to check into these possibilities in the future.

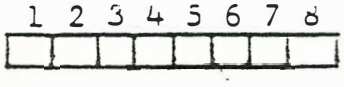
With this routine you can get a listing in HEX of the ROM or locate the command tables in Basic. Good Luck.

INTERACTORY FACTORY
By MARV LONG

Last issue we learned how to call a Rom routine to draw blocks. Let's explore another Rom routine using USR command. Try the following program and then we'll explain it. CHEERS!

```
10 CLS

20 POKE 19215,25

30 POKE 19473,0 : POKE 19474,93

40 FOR X=23808 TO 23813

50 READ Y

60 POKE X,Y

70 NEXT X

80 DATA 01, 06, 93, 195, 151, 04

90 FOR X= 23814 TO 23820

100 READ Y

110 POKE X,Y

120 NEXT X

130 DATA 16, 93, 07, 07, 01, 33, 33

140 FOR X= 23824 TO 23830

150 READ Y

160 POKE X,Y

170 NEXT X

180 DATA 254, 124, 56, 16, 16, 16, 124

190 B= USR(0)
```

We're calling another Rom routine to draw graphics. It draws very quickly and could find a home in your next program, but to use it we have to learn to draw in bytes. A byte on the screen (or anywhere else for that matter) is made up of eight bits



Each bit in this routine represents 1 pixel. The bits talk in "computer"and will accept a 1 or a Ø. The routine will "turn on"

the bits with a one.

Let's draw a dotted line with bits.

1	0	1	0	1	0	1	0

Look familiar? It should, that's the binary number 170. To draw a picture we make a chart. Convert it to binary bytes and then convert to a more usable decimal number or to hex if we're writing in machine language.

BITS	1	2	3	4	5	6	7	8	
BYTES 1	1	1	1	1	1	1	1	0	= 254 DECIMAL
2	0	1	1	1	1	1	0	0	= 124
3	0	0	1	1	1	0	0	0	= 56
4	0	0	0	1	0	0	0	0	= 16
5	0	0	0	1	0	0	0	0	= 16
6	0	0	0	1	0	0	0	0	= 16
7	0	1	1	1	1	1	0	0	= 124

How do we convert? Use last months Base conversion program. Chart it out (larger than eight bits require another byte even if you need only one more bit). Enter your binary number(base 2) and ask for decimal (base 10). It's easy! Now the program:

Lines 10 through 70 are just like last months program.
Line 80 has some new data.
- 01 Load immediatly
- 06,93 is the starting address of our routine
- C3 Jump to
- 151,04 Our Rom routine 0497 Hex

Line 90 Room for our parameters. Note 1 longer than R Fill
Line 130 Routines required information
- 16,93 This is where we'll put our table (picture) 4D10 Hex
- 07 Height in pixels
- 07 Width in pixels
- 01 Color
- 33 Horizontal position in pixels
- 33 Vertical position in pixels

Line 140 - 23824 is 4D10 hex our table- We have 7 bytes so we go to 23830
Line 180 Our picture. See the chart
Line 190 Call USR

Plot your own picture. Keep them simple at first. Line 130 must show size, and allows you to change color and screen position. Make sure you make line 140 the proper length for your picture.

Those of you with a monitor can use this routine directly in machine language. Compare the last article on RFILL with the monitor write up on same for ideas on how to do it.

What would you like to read in the factory? Write and let me know.

MARV LONG
1661 College
Ferndale, Mi. 48220

So now you've got your monitor. You've filled the screen with color, displayed the rom on the tube and drawn colored blocks. Now what. Well how about a short machine language program that really does something. In Dr. Dobbs Journal Feb. 1976 there was a program written by Steve Dompier. It plays music, not through your speaker but through a nearby A.M. radio.

MUSIC OF A SORT

```
4E00- LXI H - 21
  01-         1C  >  ADDRESS OF FIRST DATA (SONG) ENTRY
  02-         4E  >
  03- MOV A,M-7E
  04- CPI    -FE
  05-         FF  -  START OVER DATA
  06- JZ     -CA
  07          00
  08          4E
  09-MVI D   -16
  0A          25  -  TEMPO- CHANGE TO SUIT
  0B-DCR B   -05
  0C-JNZ     -C2
  0D          10
  0E          4E
  0F-MOV B,M -46
4E10-DCR C   -0D
  11-JNZ     -C2
  12          0B
  13          4E
  14-DCR D   -15
  15-JNZ     -C2
  16          0B
  17          4E
  18-INR L   -2C
  19-JMP     -C3
  1A          03
  1B          4E
```

Program written by Steve Dompier
Relocated by Marv Long

Merry Oldsmobile transcribed by

Marv Long

Now type in music data, starting at address 4E1C as follows:

39,40,45,45,72,45,45,72,45,45,45,45,40,45,4C,4C,66,4C,4C,66,4C,4C,

4C,4C,02,4C,55,5A,5A,39,39,3D,39,33,33,39,45,4C,55,4C,4C,50,4C,4C,

45,55,55,55,55,FF You should end at address 4E4D

When you have this typed in with your monitor type G4E00 then take a small transitor radio and tune to somewhere around 900 AM and lay the radio on the keyboard near the left end. You should hear the song. Play around with the tuning on the radio and move it around. The position and setting of the radio makes a major difference in results. A full scale can be found in the Dr. Dobbs reprint book #1 as well as an explanation of the program.

DATA SONGS

We've had quite a few requests for music that could be used in existing programs, so we're starting data songs. As space permits we'll run a song or two in data statements only. The program form we will use follows. We will not repeat the program.

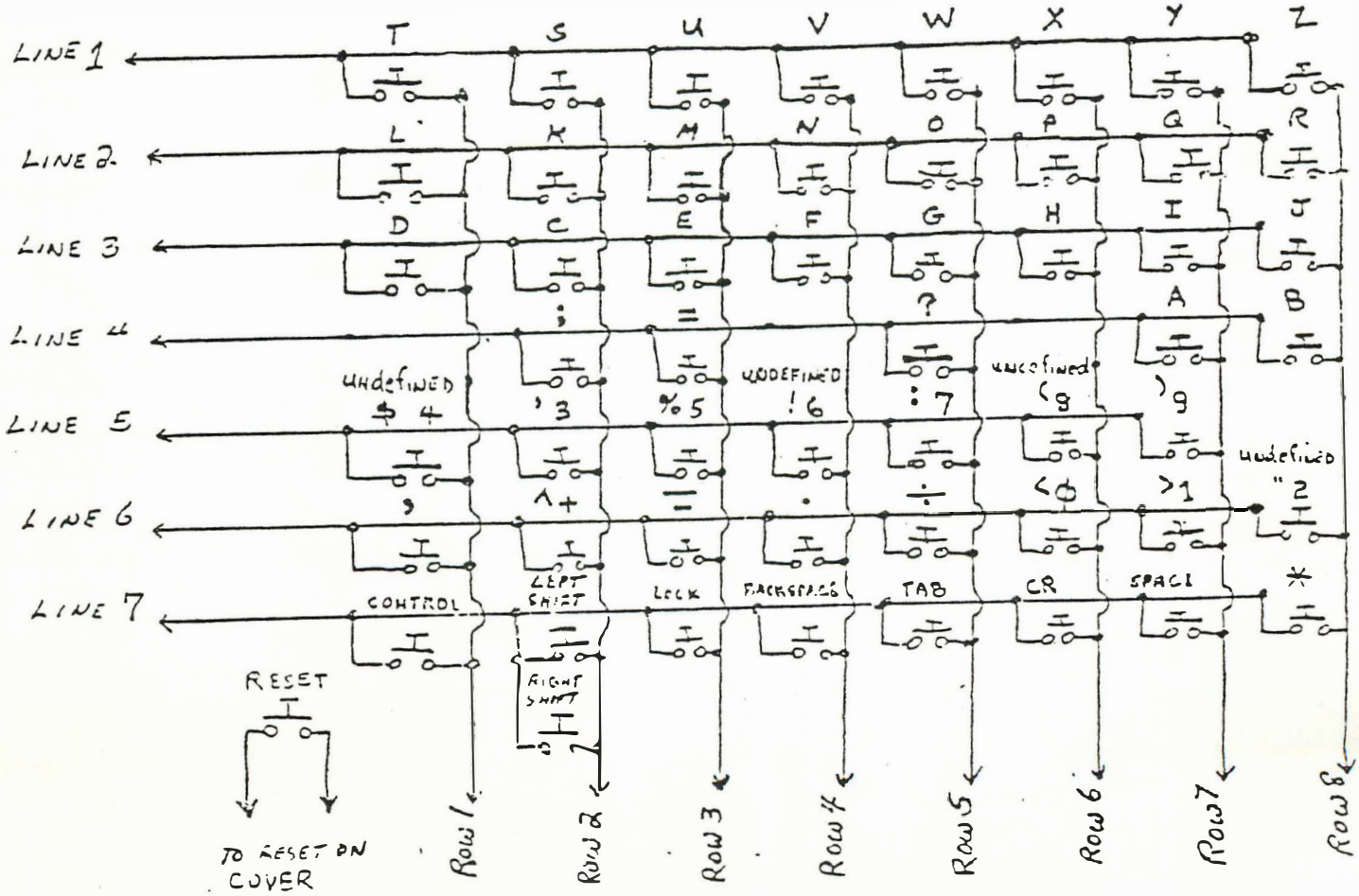
```
01 DATA SONGS
02 MUSIC TRANSCRIBED BY MARV LONG
10 FOR M=1 TO 25
```

You'll have to change line 10 to ½ the number of data In this case there's 50 items so 25 is ½ that amount.

```
20 READ X:READ Y
30 TONE X,Y
40 NEXT M
50 DATA 168,150,80,156,91,277,97,130,117,215,110,115,71,355

60 DATA 131,192,63,200,71,355,80,158,85,296,80,158,59,427

70 DATA 51,494,59,214,63,400,71,177,80,316,91,138,97,260

80 DATA 110, 115,71,370,63,420,59,480
```

For all you space nuts. When you wish upon a star.

AUXILIARY KEYBOARD MATRIX SCHEMATIC



AUXILIARY KEYBOARD

by Anthony Panetta Sterling Heights, MI

The keyboard that Interact chose to use with its computer is of a very inexpensive design. The keys actuate switches that are nothing more than metal pads that short between points on an etched board. This is similar to the technology that brought down the price of hand held calculators, but should never have been used on an item that retailed for over $500. The raised keypads found on some of the later models are not much better. It would be difficult to rebuild the existing keyboard, but it is a simple matter to run a separate keyboard in parallel.

Since the keyboard is only a series of contact switches arranged in a 7 by 8 matrix, one only has to duplicate this matrix and wire it directly to the keyboard inputs of the computer. This might sound complicated, but I was able to wire the complete matrix and connect it to the computer in one evening.

To begin, a keyboard must be purchased. Keep in mind the key travel, feel, and keycap designs while shopping. It will be a separate addition should have its own case. Watch out for exotic snap action switch designs. The Interact will not respond to switch closures of less than a couple hundred milliseconds. A few of the keycaps will have to be changed to correspond to Interact's design, but these can be painted and relabeled with press on figures.

To prepare the keyboard for wiring, the switches have to be isolated from each other. If they are mounted on an etched copper clad board, the copper pattern must be cut leaving at least 54 separate normally open switches. At this stage, I would strongly recommend purchasing a wire wrap tool of any brand to aid in wiring. A wire will be connected to one side of each switch in each line. The same is done with each row. How far the keyboard is to be located from the main computer will determine the appropriate length of cable used. At least 17 conducters are needed if a reset line is to be included. I used two 4 ft. lengths of 10 conducter ribbon cable.

To connect the auxiliary keyboard to the computer, loosen the 6 screws underneath and raise the top cover slightly. The cable connections should be visible and easy to unplug. With the cover removed, the auxiliary keyboard can be connected to either end of the cables from the Interact's keyboard. The connecter plugs have small numbers stamped on them that should correspond to the lines 1 through 7 and rows 1 through 8.

Ed. note-

A thank you to Dennis Choinski of Racine, WI , who also installed a new keyboard. His letter to me collaborates with the information in Tony's article. Both have not wired switches to the undefined matrix points. I'm curious if these contacts might produce additional characters. If someone has tried this, please write with your results.

We have also found that a chassis to chassis ground should be added to reduce RF 'hash' interference such as you get when you plug in the joysticks.

ADDITIONAL KEYBOARD INFORMATION

by Harry Holloway Ann Arbor, MI

Here are the keyboard connections, as found by probing with an ohmmeter

CONNECTION	8	9	10	11	12	13	14*	15*
PIN ON 81LS97	2	4	6	8	12	14	18	16

CONNEC-TION	PIN ON** 81LS97							
1	15	*	SPC	CR	TAB	BACK	LOCK	SHIFT	CTRL
2	14	2	1	Ø	↑	←	-	+	,
3	13		9	8	7	6	5	3	4
4	12	B	A		?		=	;	
5	11	J	I	H	G	F	E	C	D
6	10	R	Q	P	O	N	M	K	L
7	9	Z	Y	X	W	V	U	S	T

* My schematic shows the switch in these leads occurs before connectio to the joystick inputs. In fact, it occurs afterward, below the main board.

** Pin 7 strobes the joysticks

xxxxxxxxxxxxxxxxxx

This arrangement doesn't agree with the schematic, but it does tie in with Microvideo's memory map except for the position of ; , which may be a typo.

	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
3800_H	SHIFT	CTRL	LOCK	BACK	TAB	CR	SPC	*
3801	+	,	-	.	↑	Ø	1	2
3802	3	4	5	6	7	8	9	
3803	;		=		?		A	B
3804	C	D	E	F	G	H	I	J
3805	K	L	M	N	O	P	Q	R
3806	S	T	U	V	W	X	Y	Z

PRODUCT REVIEWS

MICRO VIDEO MONITOR by Micro Video Ann Arbor, MI

This is a machine language monitor program to be used instead of Interact's in-house Monitor ROM (which went in the extra ROM socket in the machine). The Micro Video Monitor is actually a better program than the M-ROM which was written by Intel. The Micro Video Monitor has the following commands:

D	Display memory	(lists contents of memory locations)
F	Fill memory	(fill memory blocks with a constant)
M	Move memory	(move memory blocks around)
P	Partial screen	(like window in Basic)
X	Examine and modify registers	
H	Hexidecimal sums and differences	
W	Write to tape	
L	Load program from tape	
R	Rewind tape	
S	Substitute	(allows individual address values to be changed)
G	Program execute (Go)	

With the Micro Video Monitor, you can write machine language subroutines easily, write machine language programs, and examine machine language programs. This program will help you understand how your Interact works. But if you do not know some machine language already, buy a book on machine and assembly language programming for 8080's first, so that you realize machine language is not at all like programming in Basic.

Reviewed by S, Cook

HAMMURABI by Micro Video Ann Arbor, MI

A machine language educational game. In this economic simulation game, you are Hammurabi, king of ancient Sumeria , and are required to determine whether to buy or sell land, how many acres of crops to plant, and how much food to distribute to the populace. All of this in order to maintain your population without starving to death too many people.

This game is quite like the Basic version from Creative Computing and published in their Basic Computer Games book. While the game can be entered on the Interact in Basic it does reach the memory limit. The Micro Video version offers a few more options of play and better explanations during play. The bottom line decision is then whether you want to spend your time writing and debugging a program in Basic or pay $8.95 for a completed smoothly running program. I've played both versions and it's a toss up which is better.

Reviewed by S. Cook

PRODUCT REVIEWS, cont.

There is a bug in INTERACT MICROCHESS: it cheats. It appears that under certain circumstances, the computer doesn't know when it is in check, and responds by checking you back.

I played four games against the computer last night (skill level 3, I played white) and found it to be an oponent of ability about equal to mine. I found the chessboard display and joystick piece selection to be extremely natural and easy to use. I won two of the games, fair and square, after long and hard-fought battle. The computer won two, but in each of these it cheated.

In one, I had a king, rook, and pawn, the computer had a king and two rooks. I checked its king with my pawn. it responded by checking my king with its rook. I had no way of telling it that its king was still in check, and that its move was thus illegal. It wouldn't let me "capture" its king with my pawn or do anything else other than move my king out of check, which I did. So I helplessly watched as it maneuvered my king into checkmate with its two rooks and declared "I WIN", still oblivious to the fact that its king had been in check with my pawn all along.

The other game was similar except that its king was in check with my bishop.

In each case, the machine apparantly felt safe in ignoring that it was in check, because it knew it could continue checking me until checkmate, thus never giving me the chance to "capture" its king. The computer has the edge here because it "owns" the chessboard.

Reviewed by Rich Pasco
Mountain View, CA

BASIC Bug: INPUT Prompts

by Rich Pasco Mountain View, CA

There is a bug in INTERACT BASIC LEVEL II, or perhaps in its documentation.

The syntax of the INPUT statement is documented as:
INPUT [<string expression>;] <variable list>
The string expression, if present, becomes a prompt.

The problem is that general string expressions do not work, and only string literals seem to work.

For example, if NAME$ has the value "SAM", then
INPUT NAME$+"'S SCORE"; S
would be expected to prompt
SAM'S SCORE?
and to set S to the numerical value typed.

But what really happens is that it prompts,
?
whatever is typed is assigned as a string to NAME$, and then it complains
SN ERROR
(syntax error) about the rest of the statement.
Parenthesizing the string expression doesn't seem to help.

The simplest fix is to change the documentation to read:
INPUT [<string literal>;] <variable list>
and to change the code to read:
PRINT NAME$+"'S SCORE";
INPUT S

EASY METHOD FOR COLORED TEXT IN LEVEL II

by Vic Volkman Grosse Pointe Park, MI

Ever wonder how different sections of text are given in different colors in such games as Star Track? As complex as this seems, it can be done with great ease by simply resetting the Interact's text color register. This register, located at 24558, can be adjusted to give any one of 255 color combinations with one POKE command. The following is a list of bytes to POKE in and their result:

POKE 24558, 0	Changes text to COLOR position #0
POKE 24558, 85	" " " COLOR position #1
POKE 24558, 170	" " " COLOR position #2
POKE 24558, 255	" " " COLOR position #3
POKE 24558, 85	" " " Two color mix
POKE 24558, 127	" " " Two color mix

There are 16 two color combinations and 4 one color possibilities for a total of 20 possibilities for the clever user. Once a character color is POKEd in, it will remain until a RESET or a backspace is hit during user input. Example program:

```
10   POKE 19215, 25 : REM INITIALIZE
20   POKE 24558, 85 : PRINT "THIS IS COLOR POSITION 1"
30   POKE 24558, 87 : PRINT "NOW STRIPED CHARACTERS!"
40   POKE 24558, 255 : PRINT "TEXT IS NOW RESET TO COLOR 3 (NORMAL)"
```

INDIVIDUALIZED INSTRUCTION AND THE INTERACT

The guest speaker for the May meeting of the Detroit Interact Group was Dr. Emanuel Hertzler, associate dean of academic affairs, University of Michigan - Dearborn. Dr. Hertzler uses Interacts for individualized Computer Assisted Instruction (CAI). He constructs a tape incorporating audio lectures with instructional programs and questions and answers. This tape can be used by a student at home in an Interact to allow studying and learning at the student's convenience. Students become involved with the learning process because of the dual sensory (aural and visual) instruction. In addition, at the conclusion of a lesson the student can be given a test which is graded immediately by the computer and provides positive reinforcement or a basis for remedial study.

Dr. Hertzler's talk was extremely interesting and possibly in the future we'll be examining and publishing more about his work. We'll try to provide information on how to create your own individualized Interact instructor programs. If any owner has already done such work, please let us know what you've done.

Announcing the

RS232C Peripheral Interface

for the Interact Computer

$99.95 (interface hardware only)

The RS232C Peripheral Interface from Micro Video offers you the following features:

Easy, in-home installation in approximately 30 minutes with no soldering required -- no need to be without your computer for even a day!

Dual port with handshaking and 3-wire send/receive for use with a printer or modem.

Crystal-controlled baud rate clock, requires no adjustment.

Standard and non-standard baud rates and other I/O parameters totally software selectable.

Wire wrap area for custom modifications.

Advanced design using state-of-the-art LSI circuitry for low-power, trouble-free operation.

Designed for upward compatibility with future hardware and software enhancements.

Interface hardware assembled, thoroughly tested, and burned in.

Equips your computer to drive any RS232C-compatible device.*

Completely compatible with Micro Video's RS232 Microsoft Level II BASIC and RS232 EZEDIT.

Clear installation instructions and documentation included.

Full 90-day parts and labor warranty.

RS232 Level II BASIC -- Floating point BASIC language with LPRINT and LLIST for lineprinter access and the USR function for machine code access..... $24.95

RS232 EZEDIT -- Same commands as standard EZEDIT program (SUBSTITUTE, APPEND, RESEQUENCE, LOCATE, etc.) PLUS the TRANSLATE command for converting your standard Level II programs into RS232 format for listing and execution..... $19.95

RS232 Pack -- The complete interface kit -- fully assembled and tested interface hardware with installation instructions and documentation, RS232 Level II BASIC, RS232 EZEDIT, blank data tape..... $129.95

*With appropriate software (not included)

More questions before you order? Call us at (313) 996-0626, and let's talk. Check, money order or VISA/MC accepted. Please add $2.00 for shipping and handling.

Complete Interact Support from MICRO VIDEO

P.O. Box 7357
204 E. Washington St.
Ann Arbor, MI 48107

More From Micro Video!!

Continuing with software development and support for owners of Interact computers, Micro Video offers:

EARTH OUTPOST I -- The mission of your space station is to defend the world against hostile alien invaders in this electrifying action game. A guaranteed winner for parties! $14.95

ATTRO-LOGIC -- The rules are simple, but sharp strategy determines who comes out ahead in this challenging logic game. Play the computer or a friend..... $ 8.95

DELUXE EQUIPMENT CASE -- Heavy-duty, velvet-lined case for transporting your computer. Three closed compartments for tapes, accessories and documentation..... $89.95

ALIGNMENT SET — Eliminate tape loading problems by aligning your computer's tape drive to the standard used by Interact, NCE/CHC, and Micro Video..... $ 8.95

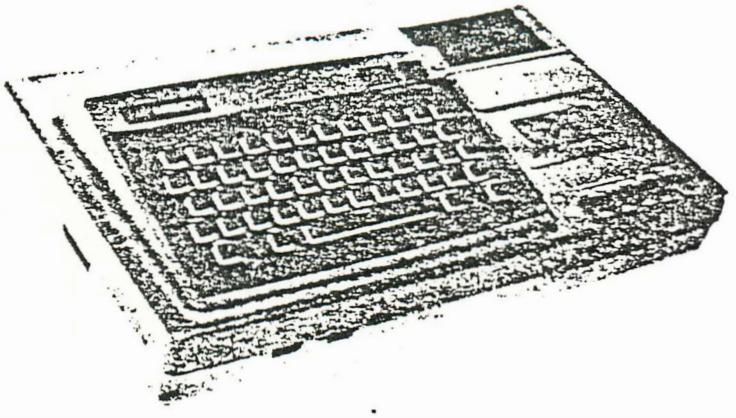
And, of course, the rest of the Micro Video line of software and accessories for Interact computers. To receive a complete product catalog and be added to our mailing list for upcoming announcements, call or write:

MICRO VIDEO

P.O. Box 7357
204 E. Washington St.
Ann Arbor, MI 48107
(313) 996-0626

Check, money order, or VISA/MC accepted. Please add $2.00 for shipping and handling.

INTERACT COMPUTER 16K LEVEL II



A powerful microcomputer with a full standard 53 key typewriter style keyboard and built-in cassette recorder for use in entering and storing programs. Utilizing the versatile Intel 8080A micro processor, the Interact computer has 16K of RAM and 2K of ROM at its disposal. Output is in color on your own TV screen. Sound is also generated through your TV set and can consist of exhilarating computer beeps or two full octaves of music. This is truly a powerful computer capable of developing and handling a variety of programs, and with availabie software provides true educational stimulus (examples: Star Track, Chess, Concentration, etc.). The Interact computer offers the ideal ooportunity to learn BASIC, the most popular computer language for small business and hobbyists, and with the use of the 84 page manual, you learn at your own pace.

INSTALLATION: Attach to antenna terminals of your color TV set and plug into 120 VAC wall outlet. Computer includes built-in RF modulator, FCC approved and UL and CSA listed.

PROGRAMING: Most Micro-computers start you off with an abbreviated 4K BASIC, and then later you have to relearn the more powerful language. Since this computer has 16K of RAM, we start you with Level II Microsoft 4.7 BASIC. BASIC is the most popular Micro-computer language using commands that are words we are ordinarily used to: such as. PRINT, NEW, GOTO (For go to), END, COLOR, JOY, INPUT, etc. To help you learn programing we have included an 84 page instruction manual, plus a 20 page book of program examples. You will immediately be able to enter into the keyboard programs that will:

1. Plot Bar Graphs
2. Draw the American and/or Canadian Flag.
3. Digital Clock, keeps time in hours, minutes and seconds.
4. Generate sounds such as a siren, fog horn, idling jet, idling motor, and more.

With a standard blank cassette tape you can store your programs for future use and even up-date information as you program.

QUESTIONS FREQUENTLY ASKED:

1. How does the Interact compare with other Micro-computers?

Answer: Interact works directly with your color TV set, and will reproduce in color. Also most other Micro-computers start at 8K. some 4K or less. Interact is 16K. In addition Interact comes with two Joysticks (usually this is extra, or not available).

Another advantage is the fact that Interact will generate sounds (music, siren, etc.) on your TV set.

Interact **BASIC** provides commands for color, sound and joystick controls.

2. Can additional peripherals be added?

Answer: No and yes. For most people the answer is no. However many hobbyists who understand electronics have adopted the Interact for printers, modems and even floppy discs. We do not condone this, however, we will supply a service manual for $5.00 that provides complete schematics. We suggest that any electronic enthusiast who decides to open the case and work directly on the computers PC board at least wait 90 days, since opening the case will violate the warranty.

SPECIFICATIONS: Intel 8080A microprocessor. 16K RAM. 2K ROM, built-in cassette recorder, 53 key keyboard (raised keys), built-in RF Modulator, computer-antenna switch, two joysticks with variable potentiometer and fire control. 120 VAC plug-in-the-wall transformer. FCC approved, UL, CSA listed.

MONEY BACK GUARANTEE-WARRANTY: Shipped direct to you from the manufacturer, on a 5 day money back guarantee and a limited 90 day warranty covering parts and labor. Out-of-warranty service available from the factory (minimum labor charge $25.00, maximum charge will not exceed $50.00).

BONUS SOFTWARE: To give you an immediate opportunity to use your Interact Computer, one software tape will be provided at no extra charge. This tape will feature a strategy-type game or educational program, and will provide insight as to the computer's full capabilities.

ORDER FORM

		Price Each	Total
_____	Quantity Interact Computer(s) with Level II BASIC and one software cassette tape.	$299.00	_____
_____	_____ ...	_____	_____
_____	Quantity, Service manual	$ 5.00	_____
	Wisconsin Residents, add 4% Sales Tax	_____	_____
	Freight and Handling	$ 7.50	$ 7.50
			_____
		TOTAL	$_____

Enclosed: ☐ Check. Money Order.

Charge to: ☐ Visa. ☐ Master Charge.

Credit Card No. ☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐ Exp. Date ☐☐☐☐

Make Payable To: MANU-TRONICS, INC.
And Send To Attn.: Ms. Catherine Paupore
9115 - 26th Avenue
Kenosha, Wisconsin 53140

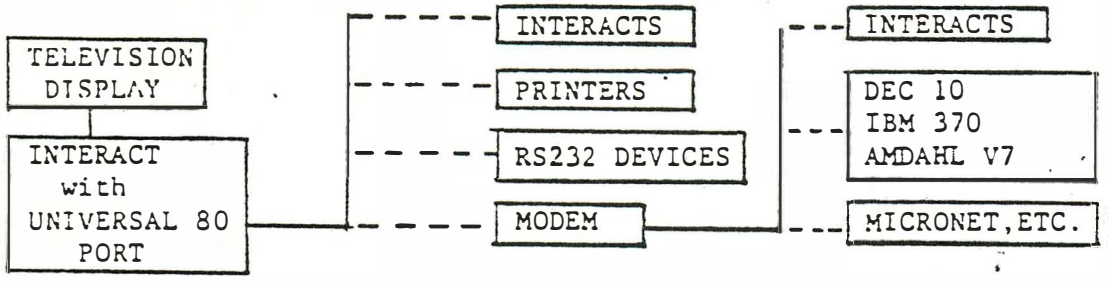
6980 CARPENTER ROAD YPSILANTI, MICHIGAN 48197 PHONE(313) 434-0033

SLAGH SYSTEM SERVICES

INTERACT IS NOW *INTERACTIVE*...AS LOW AS $49.95

SLAGH SYSTEM SERVICES CONTINUES THE UNIVERSAL 80 LINE OF RS232 PORTS WITH THE INTRODUCTION OF THE U80M, OUR NEWEST FULLY SOFTWARE PROGRAMMABLE RS232 PORT. THE U80M ALLOWS INTERACT OWNERS TO CONNECT THEIR COMPUTERS TO RS232 COMPATIBLE DEVICES, INCLUDING MODEMS, PRINTERS, AND OTHER COMPUTERS, WITH SOFTWARE PROGRAMMABLE BAUD RATE CONTROL.

POSSIBLE INTERACT CONFIGURATIONS USING UNIVERSAL 80 PORTS



UNIVERSAL 80 PORTS FEATURE

*SOLDERLESS INSTALLATION
*FULLY BI-DIRECTIONAL SERIAL
*VERY LOW POWER CONSUMPTION
*RELOCATABLE ADDRESSING
*COMPLETE DOCUMENTATION PROVIDED
*ACCESSABLE THROUGH LEVEL II BASIC OR MACHINE CODE PROGRAMMING
*PROGRAMMABLE BAUD RATES
*HANDSHAKING CAPABILITY
*PARALLEL OUTPUT CAPABILITY
*CUSTOMIZING AREA ON BOARD
*AVAILABLE AS KIT OR ASSEMBLED/TESTED

INSTALLATION AVAILABLE

THE UNIVERSAL 80 LINE OF PORTS CAN BE USED IN OTHER 8080 AND Z80 SYSTEMS

INTRODUCTORY PRICING ORDER FORM

SLAGH SYSTEM SERVICES

SPECIAL PRICING AND LIMITED TIME FREE OFFER

	KIT FORM	ASSEMBLED/TESTED	INSTALLED
UNIVERSAL 80 P.C.BOARD & INSTRUCTIONS	$19.95		
U80D 2 PROGRAMMABLE BAUD RATES	$49.95	$69.95	$89.95
U80M MULTI PROGRAMMABLE BAUD RATES	$54.95	$74.95	$94.95
DB25P INTERCONNECT CABLE	$ 8.95	$14.95	
TERMINAL PROGRAM #1 FOR INTERACT	$ 8.95 CASSETTE		

QUANTITY PRICING AVAILABLE

WE WELCOME ALL QUESTIONS AND SUGGESTIONS

***SPECIAL OFFER** ORDERS FOR THE U80M OR U80D RECEIVED BY JULY 15, 1980 WILL RECEIVE THE TERMINAL PROGRAM #1 *FREE*, AND THE DB25P CABLE AT *HALF PRICE*.

QUANTITY	ITEM	PRICE EACH	TOTAL

SUB-TOTAL_____
4% Sales Tax_____
(MI Residents Only)
TOTAL ORDER_____

NAME_____ TELEPHONE_____

ADDRESS_____ CITY_____ STATE_____ ZIP_____

PURCHASE AGREEMENT: *I AGREE NOT TO ALLOW ANY PART OF THIS ORDER TO BE REPRODUCED IN ANY FORM WITHOUT THE EXPRESS WRITTEN CONSENT OF SLAGH SYSTEM SERVICES.*

*DATE*_____ *SIGNED*_____

JCA

INTERACTION

NEWSLETTER
DETROIT
INTERACT
GROUP

DATE August - September, 1980 VOLUME NO. I, no.4

INTERACT COMMUNICATES !

Over the summer both Marv Long and I acquired MicroNET accounts. We have both used MicroNET using our RS232 equipped Interacts and modem. However at $5.00 per hour connect time we don't expect to use MicroNET a great deal. It is one way you can play Adventure on your Interact but can be quite costly if you're not a good player which I am not. I will be using MicroNET occasionally and if you would like to leave a message for me (Stephen Cook) my User ID is 7Ø15Ø,311 . Don't leave any timely or urgent messages as I don't know how often I'll be calling MicroNET. It mainly depends on the size of my VISA bill as MicroNET directly bills your credit card for time used so you don't see your money being spent.

A better system for messages are the numerous Computer Bulletin Board Systems (CBBS) around the country. If you're not familiar with these I suggest you read some of the articles published in a number of issues of Microcomputing magazine earlier this year. Look for articles entitled Dial Up Directory. Most of these systems are open and free to all users and your only cost is a long distance phone call, if the CBBS is not in your local area. Most CBBS's run evenings and many run 24 hours a day. With night phone rates you can send a cross country message for less than $3.00. One of the systems I use is the Michigan Apple Fone 313-357-1422. It is known as an ABBS (Apple BBS) because the hardware used is an Apple computer with disk drives. If you want to leave me a message on this system, first log in, and when you're asked which function type in C;M which loads the mailbox catalog file. Then proceed to leave your message. Be sure and type H for Help first if you don't understand how to operate a Bulletin Board System. Also try a local system first before making a long distance call as you can waste a lot of time and your money trying to figure out a system if you don't know what's happenning.

If you want to talk directly to my Interact call me some evening at 313-272-7594. I have a Novation modem which can run in either Answer or Originate mode up to 300 Baud, full or half duplex. We probably won't accomplish much but if you want to test your RS232 unit and terminal program, feel free to call and we'll try a hookup.

On RS232 units, we have operated both MicroVideo and Slagh units. Both perform their functions as terminals and as outputs to printers but we have not had enough time to do complete comparisions. Hopefully soon in the future we will find the time.

INTERACTORY FACTORY

by Marv Long 1661 College Ferndale, MI 48220

Well summer is over and things are hopping at the factory. Got the auxiliary keyboard on, the RS232 off and running on both printer and modem, wrote some software and received some nice mail from readers across the U.S. and Canada. Thanks for the notes, programs and suggestions, etc.

Well this month's program by Russ Patten shows how to output your own CHR$. This should be self explanatory and it's quite helpful so give it a try.

```
5 CLS
10 POKE19215,25
20 POKE24545,0:POKE24546,94
30 FORX=24064TO24083
40 READY
50 POKEX,Y
60 NEXTX
70 DATA9,9,12,0,12,0,30,0,45,0,76,128,12,0,18,0,18,0,18,0
90 FORX=24084TO24101
100 READY
110 POKEX,Y
120 NEXTX
130 DATA76,128,45,0,30,0,12,0,12,0,12,0,18,0,33,0,64,128
140 OUTPUTCHR$(1),20,20,2
150 FORX=24102TO24119
160 READY
170 POKEX,Y
180 NEXTX
190 DATA64,128,33,0,0,0,0,0,0,0,0,0,0,0,33,0,64,128
200 FORX=24120TO24137
210 READY
220 POKEX,Y
230 NEXTX
240 DATA0,0,0,0,0,0,33,0,64,128,0,0,0,0,18,0,18,0
250 OUTPUTCHR$(1),20,20,2
260 OUTPUTCHR$(3),20,20,0
265 FORX=1TO100:NEXTX
270 OUTPUTCHR$(2),20,20,2
280 OUTPUTCHR$(4),20,20,0
285 FORX=1TO100:NEXTX
290 GOTO250
```

INTERACTORY FACTORY, cont.

This month we'll add a hint on the Micro-Video monitor. If you want to add a front screen (banner) to a machine language program you've written,you'll find even with a partial command you'll get some output command to the screen. To put a banner on without this try the following. First substitute a C9 at 65F4. That will kill the scroll. Next substitute a C9 at 6437. That will kill output to the screen. You are now working blind so be careful. Take a picture you have made with compute color (if you don't have one at this point you're in a lot of troble) Load and then write the program starting at the screen (4000). You will get some color shift when you load in your new program as the color latches are not reset but you will have a banner.

Another point. The monitor's breakpoint routine sets 0's at 4C68 and 4C69. Either don't write to those addresses or you can kill this by substituting 00 at 6061-2-3. This will kill the breakpoint routine but will stop code from creeping into your's.Once you've done this trick you can write a machine language subroutine, load a matching basic program and save the whole works. It will now load with just one L load. Now you can load your combo game just like Micro-Video's Bombs Away.(That's a plug Dave-send my free Computer). Make sure you start the save at 4C06 The first 6 bytes in Basic set the stack and have the jump to the main routine. The Rom looks there at reset so you need it. If you want a banner on your combo routine then load your program first and then basic so the basic program will put in those bytes. Next month a scroll routine.....SIDEWAYS.....

BUS DRIVERS ? WHAT BUS DRIVERS ?

by Thom Linehan 410 Normandy Royal Oak, MI 48073

Most TTL based computers use drivers on the address and data busses. This enchances the noise immunity of the system. Perhaps because Interact chose not to make these busses externally available they also chose not to drive them. (It also made the entire system cheaper and more compact.)

I am presently working out a driver system for the CPU. The stock 1.9 ma of drive is just not enough for serious system expansion. Hopefully by late fall I may have a prototype final board for an S-100 interface. This would allow for numerous sources of expansion boards.

Hopefully, also, it will not require taking the big knife to the main circuit board.

SIG INTERACT

A new Interact club has been formed in the Chicago area. It is part of the Chicago Area Computer Hobbyist Exchange (CACHE). It is a Special Interest Group (SIG) within the club. CACHE meets the third Sunday of every month at 11:00 AM at the DeVry Technical Institute 3300 N. Campbell Chicago (Campbell is 2500 west).

For more information on the Interact SIG contact Jean Barber at (312) 945-4171 . General CACHE meeting information is available on their recorded Hotline (312) 849-1132 .

DIGITAL CLOCK BUG

by Gregg Pittenger 3430 Dresden Columbus, OH 43224

I found that the program 'digital clock' (Interact's Basic Examples Booklet, pages 23-24) changes time from AM to PM (or PM to AM) at 1:00, not at 12:00 as it should. To correct this, retype line 600 and add line 545.

```
600   IF H1Ø = 2 THEN H1Ø = Ø : RETURN
545   IF H = 2 AND H1Ø = 1 GOSUB 630
```

LETTERS

Letters to the Editor
Interaction
15356 Prevost
Detroit, MI 48227

Dear Sir:

The lead article on the RS232 ports in the last issue left a confusing point on the role of Level II BASIC support of the ports. As the licensed supplier of both Level II BASIC and RS232 BASIC interpreters from Microsoft, I'd like to clarify two points.

First, regular Level II BASIC does not support use of either the U80 or the Micro Video port. Level II BASIC was developed at Interact long before the idea of expansion was even considered feasible. Only the later RS232 BASIC supports directly the LLIST and LPRINT facilities which are highly desirable for program listings and printer-based applications such as reports, tables, and word-processing. Our RS232 BASIC is designed and recommended for use only with the Micro Video interface.

Second, Micro Video ROM BASIC (yes, it does exist!) is based on the RS232 BASIC and our interface. While we are still evaluating the economics of offering our ROM BASIC , owners should know that one of our main goals is upward compatibility. Regardless whether owners are interested in expansion through the RS232 port, RAM expansion, or BASIC in ROM expansion, the common denominator of all these expansions is the RS232 BASIC.

David L. Ross
Micro Video

LETTERS, cont.

SLAGH
SYSTEM
SERVICES

Sept 1980

NEWS RELEASE(of sorts)

WE WEREN'T PREPARED.....

.....For the response we got from Interact owners for orders on the U80 Series Ports. The number of orders exceeded our ability to fill them in a prompt manner, and we owe our customers an apology. We had not arranged for adequate supplies of parts from our suppliers, and as a result we were not able to ship on time. This problem has been resolved, and we do not expect it to reoccur.

Our current plans call for shipment of orders within ten days.

In case we have overlooked anyones request for information, please contact us by mail or phone and we will answer your inquiry promptly.

Again, our apologies to anyone inconvenienced by the delays, and thank you for your business.

Sincerely,

Mark Slagh

SCREEN POKES CORRECTIONS

In response to letters from K. Stuempes of Milwaukee, WI and G. Hall of San Jose, CA , yes, there is an error in the screen edge pokes published in Interaction no. 1. The original values were derived from screen observations and the actual limits may not visible. The correct values are as follows :

```
POKE 16384, n  is the upper left corner
POKE 16415, n  is the upper right corner
POKE 18912, n  is the lower left corner
POKE 18943, n  is the lower right corner
```

THANK YOU

A THANK YOU TO DAVID ROSS OF MICRO VIDEO CORPORATION FOR PRINTING SOME OF THE PROGRAMS PUBLISHED IN THIS ISSUE.

CLOCK

by Walt Chmielewski 17303 Broken Back Crosby, TX 77532

This program demonstrates another type of computer clock that uses program execution time as a timing loop. This is how TRS80 owners have to program clock and timer routines. This clock is not very accurate and additional plots will have to be added or deleted periodically to correct errors. Walt suggests adding a line (195 PLOT 54, 7, 3) to slow timekeeping or removing line 190 to speed clock. He wrote this program before he knew about the time keeping registers.

```
0 CLS:PRINT"INSTRUCTIONS:"
1 PRINT:PRINT"IF INSTRUCTIONS  ARE NOT WANTED   TYPE RESET,R,RUN 10":PRINT
2 PRINT"TYPE IN HOUR,    HIT CR           TYPE IN MINUTES"
3 PRINT:PRINT"HIT CR":INPUT
4 CLS:PRINT"HIT CR,          TYPE IN SECONDS  HIT CR, CLOCK    STARTS RUNNING
6 PRINT:PRINT"HIT CR":INPUT:CLS
8 PRINT"IF 0'S OR CR'S   ARE TYPED IN FOR TIME, CLOCK      BECOMES STOPWATCH"
9 PRINT:PRINT"HIT CR":INPUT:CLS:GOTO 200
10 CLS:INPUT"HOUR";X:INPUT"MINUTES";Y:INPUT"SECONDS";Z:CLS
20 Z=Z+1
30 IF Z=60 THEN Y=Y+1:Z=0
40 IF Y=60 THEN X=X+1:Y=0
50 IF X=13 THEN X=1
55 CLS
60 OUTPUT":  :",40,14,7
70 OUTPUTZ,60,14,7
80 OUTPUTY,40,14,7
90 OUTPUTX,20,14,7
102 PLOT18,9,3
104 PLOT18,11,3
106 PLOT18,13,3
108 PLOT18,15,3
110 PLOT18,17,3
112 PLOT18,19,3
116 PLOT88,9,3
118 PLOT88,11,3
120 PLOT88,13,3
122 PLOT88,15,3
124 PLOT88,17,3
126 PLOT88,19,3
128 PLOT20,20,3
129 PLOT20,7,3
130 PLOT86,20,3
131 PLOT86,7,3
132 PLOT22,20,3
134 PLOT84,20,3
136 PLOT24,20,3
137 PLOT24,7,3
138 PLOT82,20,3
139 PLOT82,7,3
140 PLOT26,20,3
142 PLOT80,20,3
144 PLOT28,20,3
145 PLOT28,7,3
146 PLOT78,20,3
147 PLOT78,7,3
148 PLOT30,20,3
150 PLOT76,20,3
152 PLOT32,20,3
153 PLOT32,7,3
154 PLOT74,20,3
155 PLOT74,7,3
156 PLOT34,20,3
158 PLOT72,20,3
160 PLOT36,20,3
161 PLOT36,7,3
162 PLOT70,20,3
163 PLOT70,7,3
164 PLOT38,20,3
166 PLOT68,20,3
168 PLOT40,20,3
169 PLOT40,7,3
170 PLOT66,20,3
171 PLOT66,7,3
```

CLOCK, cont.

```
172 PLOT42,20,3
174 PLOT64,20,3
176 PLOT44,20,3
177 PLOT44,7,3
178 PLOT62,20,3
179 PLOT62,7,3
180 PLOT46,20,3
182 PLOT60,20,3
184 PLOT48,20,3
185 PLOT48,7,3
186 PLOT58,20,3
187 PLOT58,7.3
188 PLOT50,20,3
190 PLOT56,20,3
192 PLOT52,20,3
193 PLOT52,7,3
194 PLOT54,20,3
196 GOTO 20
200 PRINT"COLOR MAY BE     CHANGED PER PAGE 57 OF USERS      MANUAL":PRINT
202 PRINT"HIT CR":INPUT:CLS
204 PRINT"TO CHANGE COLOR  HIT RESET AND R,  CHANGE COLOR PER PG. 57":PRINT
206 PRINT"THEN TYPE RUN"
210 PRINT:PRINT"HIT CR":INPUT
220 GOTO10
```

MEMORY EXPANSION

A flyer is being distributed which tells Interact owners to write me for RS232 and memory expansion plans and kits. There are ads in the newsletter for RS232 units, but I personally do not have any schematics or plans for such. As for memory expansion, while a number of people have written to me that they are working on it, no one has written that they have completed a memory expansion unit or finished an S100 interface.

As soon as anyone finishs either and they tell me about it, I'll be glad to print any information they send me or run their ad if they wish to sell plans or kits. Some people keep saying that memory expansion is easy which may be the case, but not for me, and I have yet to see a demonstration of a working prototype.

ATTENTION HOUSTON

There is already one Interact owners' group in the Houston area that meets informally. Now there is interest in forming another group.

For information on the Houston users' group contact Jack Germaine at 713-334-1028

If you're interested in joining a group meeting in the south Houston area, contact Kevin TenBrook at 713-771-3077

CANADA NEWS

HI "INTERACTERS"

Being owner of an Interact for more than one year now, I studied the internals of the computer by the software. I am now proud to have my own monitor, designed by my-self (Isn't the best first application for machine language.)

As first contribution to Interaction, I present you one of my first Interact program to learn more about the SOUND command.

In the future, I will provide you with a dis-assembler listing of the ROM MONITOR once I receive and debug my RS-232 Interface from Slagh System Services.

The SOUND command, as you know, needs two operands: the first from 0 to 7 (000 to 111 in binary), the second from 0 to 32767 (000 0000 0000 0000 to 111 1111 1111 1111 in binary). Comparing these bits to the pin layout of the SN76477 SOUND GENERATOR IC, I found some relationship. For example:

value 4096 - SYSTEM INHIBIT
2 - DECAY resistors
1 - DECAY
and many others you will have fun to find by your-self.

OPERATION OF THE PROGRAM.

Load EDUBASIC.
Load the SOUND STUDY program.
On the message: ENTER 15 LETTERS REPRESENTING...
enter: + * ; l k j h g f d s a ? . ,
(any string except numerals)
On the message: ENTER THE DECIMAL VALUE...
enter a value from 0 to 7
The program is now running. Try to press different numeric
to get the idea of it.
Now, by pressing one of the 15 letters, you will see its value

added to the second number on the screen.
By pressing again the same letter, you will see its value
substracted from the second number.
You can now experiment a lot. I suggest you try to find the
technical specs of the IC. It helps.

Good luck,

GERARD MICHAUD
449, Pere Rouillard
Rimouski, P. Que.
G5L 5W7
Tel.: 418-723-4313

CANADA NEWS, cont.

```
  1   CLEAR
  8   P."   SOUND STUDY"
 10   P.;P."   ARRAY"
 11   P."INITIALISATION"
 12   P.
 15   FOR  I = 0TO  673
 20   ! (I) = 0; NEXT I
 30   P."ENTER 15 LETTERS"
 32   P." REPRESENTING THE"
 34   P." BIT POSITIONS OF"
 36   P." THE SECOND BINA-"
 37   P." RY NUMBER OF THE"
 38   P." SOUND COMMAND"
 39   P." LOW ORDER BITS
        AT FIRST
 40   B=1
 50   A=INCHR
 52   P. B
 60   !(A) = B
 70   IF B=16384 THEN 100
 80   B =2*B
 90   GOTO 50
100   P.
118   P."ENTER THE DECIMAL"
120   P." VALUE OF THE"
122   P." FIRST NUMBER"
124   P." OF THE SOUND
        COMMAND
125   B=0
130   A=INCHR
134   IF ! (A)<>0 GOTO 140
135   IF A-48<0 GOTO 130
136   IF A-48>9 GOTO 130
137   C=A-48
140   B=B + ! (A)
150   !(A) = - !(A)
160   SOUND (C,B)
170   P.C," AND",B
180   GOTO 130
```

TAPE LOAD PROBLEMS ?

by Thom Linehan 410 Normandy Royal Oak, MI 48073

Interact suggests not placing the computer near the television while in use, for noise reasons. May I also suggest keeping the unit off concrete floors! (Especially basement floors) It appears as though it makes a nice RF trap causing an increase in tape misload occurences. Keep the unit at least a foot from concrete floors.

GUESS THE ANIMAL

Guess The Animal is based on ANIMAL a game published in Creative Computing magazine. When I wrote this program last January, I got another of many disappointments in the Interact when I realized I couldn't save the string array A$ without translating it to numerical data before writing to tape. I have since found out this is common to many systems. If anyone writes a compact string array saving routine, write Interaction and we'll be glad to publish it.

The game is pretty much self explanatory. After the question "ARE YOU THINKING OF AN ANIMAL?" , if you type "LIST" the program will print out a list of animals it knows.

```
5 CLEAR1000
10 CLS
30 PRINT:PRINT:PRINT
40 PRINT"GUESS THE ANIMAL"
45 PRINT
50 PRINT" THINK OF AN AN- IMAL AND THE COM-PUTER WILL TRY TOGUESS IT."
60 PRINT
70 DIMA$(200)
80 FORI=0TO3
90 READA$(I)
100 NEXTI
110 N=VAL(A$(0))
120 REM MAIN CONTROL SECTION
130 INPUT"ARE YOU THINKING OF AN ANIMAL";A$
140 IFA$="LIST"THEN600
150 IFLEFT$(A$,1)<>"Y"THEN120
160 K=1
170 GOSUB390
180 IFLEN(A$(K))=0THEN999
190 IFLEFT$(A$(K),2)="%Q"THEN170
200 PRINT"IS IT A ";RIGHT$(A$(K),LEN(A$(K))-2);
210 INPUTA$
220 A$=LEFT$(A$,1)
230 IFA$="Y"THENPRINT"WHY NOT TRY      ANOTHER ANIMAL?":GOTO120
240 INPUT"THE ANIMAL YOU   WERE THINKING OF WAS A";V$
250 PRINT"PLEASE TYPE IN A QUESTION THAT    WOULD DISTINGUISHA "
260 PRINTV$;" FROM A ";RIGHT$(A$(K),LEN(A$(K))-2)
270 INPUTX$
280 PRINT"FOR A ";V$;" THE ANSWER WOULD BE ";
290 INPUTA$
300 A$=LEFT$(A$,1):IFA$<>"Y"ANDA$<>"N"THEN280
310 IFA$="Y"THENB$="N"
320 IFA$="N"THENB$="Y"
330 Z1=VAL(A$(0))
340 A$(0)=STR$(Z1+2)
350 A$(Z1)=A$(K)
360 A$(Z1+1)="%A"+V$
370 A$(K)="%Q"+X$+"%"+A$+STR$(Z1+1)+"%"+B$+STR$(Z1)+"%"
380 GOTO120
```

GUESS THE ANIMAL cont.,

```
300 REM QUESTION SUBROUTINE
400 Q$=A$(K)
410 FORZ=3TOLEN(Q$)
415 IFMID$(Q$,Z,1)<>"%"THENPRINTMID$(Q$,Z,1);:NEXTZ
420 INPUTC$
430 C$=LEFT$(C$,1)
440 IFC$><"Y"ANDC$><"N"THEN410
450 T$="%"+C$
455 FORX=3TOLEN(Q$)-1
460 IFMID$(Q$,X,2)=T$THEN480
470 NEXTX
480 FORY=X+1TOLEN(Q$)
490 IFMID$(Q$,Y,1)="%"THEN510
500 NEXTY
505 STOP
510 K=VAL(MID$(Q$,X+2,Y-X-2))
520 RETURN
530 DATA"4","%QDOES IT SWIM%Y2%N3%","%AFISH","%ABIRD"
600 PRINT:PRINT"ANIMALS I ALREADYKNOW ARE:"
605 X=0
610 FORI=1TO200
620 IFLEFT$(A$(I),2)<>"%A"THEN650
624 PRINTSPC(1);
630 FORZ=3TOLEN(A$(I))
640 IFMID$(A$(I),Z,1)<>"%"THENPRINTMID$(A$(I),Z,1);:NEXTZ
645 X=X+1:IFX>5THENX=0:PRINT
650 NEXTI
660 PRINT
670 PRINT
680 GOTO120
999 END
```

APOLOGIES DUE FOR POSTAGE DUE

I would like to apoloigize to anyone who received their newsletters with postage due. When I was mailing multiple issues earlier this year, I misinterpeted the postal rate chart and sent out a number of envelopes with insufficient postage. I did not realize my mistake until 6 weeks after one was mailed out it was returned to me for additional stamps.

EZEDIT IDIOSYNCRACY

from Micro Video PO Box 7357 Ann Arbor, MI 48107

The APPEND and RESEQUENCE commands in Micro Video's EZEDIT program function differently from the way they are documented. These commands do not handle multiple line references within a single line during the renumbering operation; only the first line reference encountered is changed to reflect the new line numbers. While not as convenient, this idiosyncracy is not uncommon in language and editing programs. You can circumvent retyping and editing by putting line references on separate lines during initial program entry.

CALENDAR

by John Robinson 725 Berry Lane Lexington, KY 40502

This program will output a monthly calendar for any year since the introduction of the Gregorian calendar. The program gives this date as 1582 but in many countries, it was the late 18th century. England, for example, did not adopt the current system until 1752

```
1 REM CALENDAR BY JOHN ROBINSON
2 REM 725 BERRY LANE, LEXINGTON, KY 40502
3 REM FOR INTERACT COMPUTER LEVEL 2 BASIC
5 COLOR 3,2,1,4
6 CLS
8 D$(1) = "S":D$(2) = "M":D$(3) = "T"
9 D$(4) = "W":D$(5) = "T"
10 D$(6) = "F":D$(7) = "S"
11 D=1
20 PRINT"    CALENDAR"
30 PRINT:PRINT"THIS PROGRAM WILL"
40 PRINT" PRINT A MONTHLY"
42 PRINT"CALENDAR FOR ANY"
44 PRINT" YEAR FROM 1582."
50 PRINT" REMEMBER TO HIT"
55 PRINT" ANY KEY TO SEE"
60 PRINT" THE NEXT MONTH"
65 PRINT
70 INPUT"WHICH YEAR ";Y
71 M=1
72 MM=M
73 IFM>2GOTO100
75 MM=M+12:YY=Y-1
80 REM
100 F=D+2*MM+INT(.6*(MM+1)+YY)+INT(YY/4)-INT(YY/100)+INT(YY/400)+2
160 DAY=F
162 DAY=DAY-7
164 IFDAY>7GOTO162
200 REM
```

CALENDAR, cont.

```
230 IFM=1THENNN=31:M$=" JANUARY"
240 IFM=2 AND Y/4=INT(Y/4)THENNN=29:M$="FEBRUARY"
250 IFM=2 AND Y/4<>INT(Y/4)THENNN=28:M$="FEBRUARY"
260 IFM=3THENNN=31:M$="  MARCH"
270 IFM=4THENNN=30:M$="  APRIL"
280 IFM=5THENNN=31:M$="   MAY"
290 IFM=6THENNN=30:M$="   JUNE"
300 IFM=7THENNN=31:M$="   JULY"
310 IFM=8THENNN=31:M$="  AUGUST"
320 IFM=9THENNN=30:M$="SEPTEMBER"
330 IFM=10THENNN=31:M$="OCTOBER"
340 IFM=11THENNN=30:M$="NOVEMBER"
350 IFM=12THENNN=31:M$="DECEMBER"
360 QQ=DAY
370 IFDAY=0THENQQ=7
380 CLS
390 OUTPUTM$,13,64,3
394 OUTPUTY,73,64,2
400 YY=54:XX=13
405 FORTT=1TO7
410 OUTPUTD$(TT),XX,YY,3
420 XX=XX+14
428 NEXTTT
430 YY=45:XX=7:SS=0
431 GOTO5000
432 FORL=1TONN
435 GOSUB2000
450 IFQQ/7=INT(QQ/7)GOTO4000
460 QQ=QQ+1
470 NEXTL
480 A$=INSTR$(1)
1982 M=M+1
1984 IFM>12THENM=1:Y=Y+1
1986 DAY=DAY+NN
1987 DAY=DAY-7
1988 IFDAY>7GOTO1987
1990 GOTO200
2000 REM
2001 IFSS=1GOTO2005
2002 IFL>9THENSS=1:XX=XX-4
2005 OUTPUTL,XX,YY,1
2010 XX=XX+14
2020 RETURN
4000 YY=YY-7
4010 XX=7
4020 SS=0
4025 QQ=0
4030 GOTO460
5000 REM
5002 IFDAY=1GOTO5100
5003 IFDAY=0THENDAY=7
5005 FORPP=2TODAY
5010 XX=XX+14
5030 NEXTPP
5100 GOTO432
```

MORE TIME ON REAL TIME

by David Schwab 10 Jay Lee Court Ann Arbor, MI 48104

I have some more information on the Interact's internal clock that may be useful. It seems that input from the keyboard and joysticks does not cause a CPU interrupt, but rather the CPU is gated through a dividing circuit that causes a CPU interrupt precisely every 1/60 of a second. The dividing circuit is also used for the video driver. If CPU interrupts are enabled (EI /DI 8080 instructions), the current program is interrupted and control jumps to a ROM routine that, among other things, checks the keyboard to see if any keys are depressed, checks the joysticks to see if they have been moved, and adds 1 to the 16 bits of data at locations 5FEF - $5FF0_H$ with an LHLD, INX H, SHLD sequence of machine language instructions.

Then control is returned to the interrupted program. This happens so fast that it is usually imperceptible. If nothing happens that disables interrupts, like a tape read or write, the low order byte of the clock (5FEF) is incremented every 1/60 of a second 256 times until it overflows into the high order byte and is reset to zero. At this time the high order byte (5FF0) is incremented. The period of the low order byte is then 256 x 1/60 second or 4 4/15 seconds. The high and low order bytes considered as a single 16 bit number have a period of 65536 x 1/60 seconds or 1092 3/5 seconds (a little more than 18 minutes). Of course the value of either byte can be zeroed by your program after any specified interval of time less than this. Therefore, for precise timing of periods longer than 4 4/15 seconds, the data at locations 5FEF - $5FF0_H$ should be condisered as a single 16 bit number stored in LO, HI format.

I don't know if anyone is ready to use it yet, but there is provision in the ROM interrupt routine for a user-defined routine to be called by the interrupt handler (every 1/60 second). The address of the user-defined routine is simply stored at locations 5FF3 - $5FF4_H$ in LO, HI format. If these bytes are nonzero, the interrupt handler will CALL the routine at that address before it returns control to the interrupted program.

NEWS RELEASE 8/15/80

SLAGH SYSTEM SERVICES

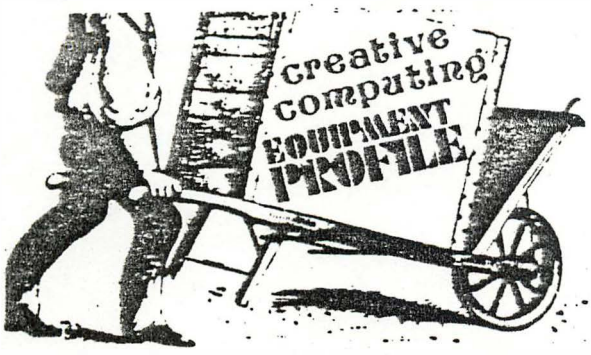
INTERACT CALL IN NIGHT.....THURSDAY

THURSDAY EVENINGS SLAGH SYSTEM SERVICES WILL TAKE CALLS AND ANSWER QUESTIONS FROM INTERACT OWNERS FROM SIX PM UNTIL MIDNIGHT, EASTERN TIME.

SINCE SLAGH SYSTEM SERVICES IS INTERESTED IN SUPPORTING THE THE INTERACT USERS COMMUNITY, WE MAKE OURSELVES AVAILABLE TO ANSWER YOUR QUESTIONS IN THE EVENING, WHEN MORE USERS CAN CALL.

SINCE MOST DAYS I AM NOT IN THE OFFICE, THESE EVENING HOURS WILL MAKE IT EASIER TO GET IN TOUCH WITH ME PERSONALLY. EVENING CALLS ARE WELCOME ANY DAY OF THE WEEK, BUT WE WILL TRY TO BE AVAILABLE EVERY THURSDAY EVENING, AND LOOK FORWARD TO HEARING FROM ANYONE THAT HAS ANY QUESTIONS REGARDING OUR PRODUCTS, SERVICE PROBLEMS, OR GENERAL INTERACT QUESTIONS.

New owners have been asking about - the original price of an Interact. This reveiw is from the Nov., 1979 issue of Creative Computing.



A Low Cost Personal Computer

Interact Computer

Steve North

Although the power and sophistication of personal computers continues to leap forward, there have been few serious attempts to lower price barriers. Even the new Atari and Texas Instruments microcomputers, both solid but unsurprising efforts, are priced at or above the $1000 level. Thus, we're pleased to report on a new, low-cost personal computer, the Interact Computer.

The Interact Computer comes in three configurations: the basic unit with a tape loader in ROM is $499.95. The same computer with an RS-232 printer and communications port is $599.95, and the Level II BASIC-in-ROM unit with the RS-232 interface is $699.96. All the models include 16K RAM, color graphics, joysticks and plug into a TV set. The Interact has a built-in RF modulator for direct connection to the antenna terminals and apparently has FCC approval. A built-in cassette deck provides program and data storage.

The term "Level II" BASIC is an obvious allusion to TRS-80 Level II since Interact does not even have a "Level I" BASIC. It is also a somewhat inaccurate term because Interact Level II is not the same as TRS-80 Level II and some program conversions might be difficult. Both BASICs were written by Microsoft and customized for each particular machine, but Interact BASIC appears to be enhanced 8K BASIC whereas TRS-80 Level II is enhanced Extended 12K BASIC. The Interact does not have IF-THEN-ELSE (only IF-THEN), PRINT USING, EDIT, AUTO, or integer and double precision variables. However, it retains many of the most powerful features of TRS-80 Level II, such as multidimensional arrays, character string arrays, PEEK, POKE, single character input, etc. In addition, Interact BASIC has several statements for high-level control of the display color, text scrolling window and audio output.

The graphics are in 8 colors and 78 x 113 resolution, which is sufficient for most games and applications. By comparison, the Apple has 16 colors in 40 x 40 resolution, or 6 colors in 160 x 280 high-resolution mode. The Interact is at a noticeable disadvantage when it comes to text, though, since characters are displayed as dot matrix patterns on the same grid, yielding very large squarish letters and numbers. As a result the text display is 12 lines of 17 characters, equivalent to 3 or 4 lines of BASIC — a handicap if you write your own programs. However, the color graphics are good for a computer in this price bracket.

The Interact computer has monophonic audio output through the TV speaker, useful for sound effects and simple tunes. The computer can also channel the cassette audio input to the TV speaker so you can hear tapes as they load, a nice feature.

The joysticks are not true analog inputs, but work by activating contact switches to detect the direction in which the stick is pushed. Each joystick has a "fire" button and a control knob mounted at the end of the stick for analog input, such as game paddle positioning. The keyboard layout is like a typewriter but there are spaces between the keys. The keys work like calculator buttons, to be stabbed at rather than gently pressed, which is fine for hunt-and-peck typists but not for touch-typists. We also had a little trouble with bad contacts in the joystick units. This could probably be corrected with some TV Tuner aeresol cleaner or, at the very worst, by taking the joysticks apart and cleaning them.

The Interact Computer is not supplied with software from second sources yet, but Interact has released an adequate line of games and applications on its own. Our favorite games were Breakthrough (a Breakout-style game), Backgammon and Dogfight. This last game commanded the time of our software development staff for almost a half hour a day while the Interact was in our office, revealing at least one latent Red Baron (and also some dissatisfaction with the joystick units). Interact also has over 20 other tapes, including Star Trek, Computer Maze, Hangman, Music Maestro, Compute-a-Color (a color etch-a-sketch program) and two Financial Planning cassettes. Most of the Interact software is very good (but not mind-blowing) and the prices (around $14.95 a program) are likewise OK but not exceptional.

In summary, the good points of the Interact are:

- *Inexpensive
- *Good BASIC
- *Color Graphics

And the bad:

- *Not expandable
- *Limited text displays
- *Keyboard and joystick quality

In interpreting this review, please keep in mind that it's not very significant to compare a $600 computer like the Interact, with a computer costing two or three times as much. For example, the Apple has a better keyboard, expandability and more software, but it costs $1195. If you pay less, you have to expect to give up something. For its price, the Interact is a good performer and it's worth your consideration if you're looking for an under-$1000 personal computer. □

8080 DISASSEMBLER in BASIC - This LEVEL II BASIC program lists addresses, contents, corresponding ASCII character, and 8080 assembler language op codes and registers for any memory locations. It will disassemble ROM, the BASIC interpreter, and, in conjunction with the MICRO VIDEO MONITOR, any INTERACT 8K packaged program or the MONITOR itself. For a MICROSETTE cassette tape and complete instructions including a sample form for listing the output returned by first class mail send $5 00 to:

David J. Schwab
10 Jay Lee Court
Ann Arbor, MI 48104

ORIGINAL INTERACT TAPES $10 each, all six for $50
Backgammon, Breakthrough, Interact Microchess
Star Track, Reversi (Othello), Volleyball
Mickey Kress, 2625 North St. Anthony, Jackson, MI 49203

INTERACT PROGRAMS.....Excellent Condition..... Shipped Postpaid

Calculator, Breakthrough, Regatta, Showdown, Trailblazers -- $5.00 each
Microchess -- $10.00
only one of each available !
Paul Nagrant 31136 Carriage Hill Ct. Farmington Hills, MI 48018

* *DISSASSEMBLER* *TRANSLATOR(ASSEMBLER)* *WRITE 16K TAPES* *

At last. Everything you always wanted in a machine-language monitor and were afraid to ask for. Announcing the

HILO MONITOR

for 16K Interacts. This taped program is an approximately 6K monitor with the user option of location at either high end or the low end of RAM. Its features include:
MEMORY MANIPULATION. The usual dump, fill, substitute, and move plus disassembly listing, mnemonic translator (input programs in 8080 mnemonics instead of looking up op-codes),and an offset references command that may be used to aid relocation of code or to search for and change a specified address.
DEBUGGING AIDS. Execute user code with up to two breakpoints. Examine saved register contents. Load specified register contents.
TAPE OPERATIONS. Read and write with filenames (filenames do not affect load without monitor). Option of writing from output list permits creation of the full range of Interact tape structures. Write without endfile permits chaining of tape writes. Edited tape load permits loading of specified data blocks into user-specified locations. Header record load permits examination of tape structures. Verify command checks data written to tape. Video command permits tape load and write without the commands echoing to the screen (may be used to add professional-quality video banners to programs).
MISCELLANEOUS UTILITIES. Hexadecimal sum and differences. Interconversions between hexadecimal/decimal, op-code/mnemonic, and character/ASCII. Partial scroll.
COMPREHENSIVE DOCUMENTATION - with special emphasis on tape I/O.

Available October 1st, 1980 from Harry Holloway, P.O. Box 2263 Ann Arbor, Mi. 48220--Price $20 postpaid, Mi. residents please add 80¢ sales tax.

SLAGH SYSTEM SERVICES

INTERACT PRODUCTS AND SERVICES

U80	Printed Circuit Board for RS232 interface. Comes with schematics and programming notes.	$29.95
U80M	Multiple software programmable baud rates. Available as kit of parts including printed circuit board, or assembled and tested. The U80M can be operated through standard Level 2 Basic, by writing your own sub-routine or software is available from us, as listed below.	
	Kit, including parts, circuit board, and instructions	$54.95
	Assembled and Tested, ready to install	$69.95
	DB25P Cable for most interfacing	$ 8.95
	Installation of Assembled and Tested U80M	$20.00
T80I2	Terminal Program. Profile your U80M port by setting baud rates, parity, word length under software control. Generate new characters for special operatingconditions. Introductory Price	$ 8.95
P80I1	Printer Output tape overlays standard Level 2 Basic to allow output to a printer through your U80M port without editing your programs or buying a new Basic language. Call us for full details.	$ 8.95
Executive	This program enables you to program in machine code and modify any existing programs, including Level 2 Basic if you wish. All memory locations may be examined and changed, and programs written may be saved to tape.	$ 8.95
C80I1	Available after October 1, this program will allow you to transmit and receive data from your Interact with the U80M port over phone lines, or directly from one computer to another. Price to be announced	
	PLEASE add $3.00 shipping and handling for U80 Products, 50¢ shipping and handling for tapes.	
	Michigan residents add 4% state sales tax	
SERVICE	Call us for FREE service advice Thursday Evenings between 6 pm and Midnight.	
PARTS	SLAGH SYSTEM SERVICES will be putting together a group order to go to Manu-tronics October 15, anyone wishing to order parts may do so at cost plus shipping and handling. Call us for more details.	

SLAGH SYSTEM SERVICES BOX 438 YPSILANTI MI 48197 313-434-0033

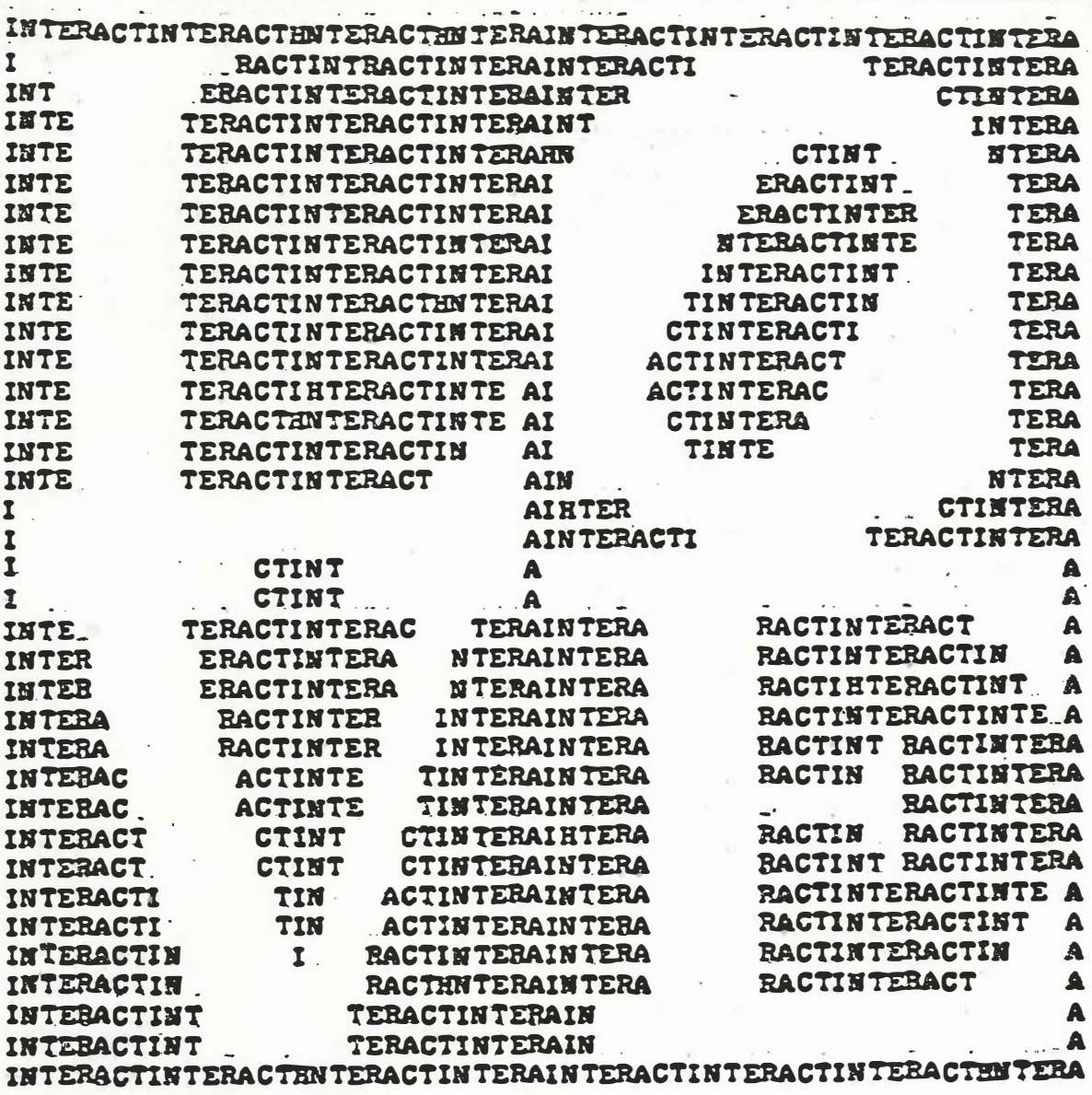
P.S. THE U80M WILL DRIVE 20mA CURRENT LOOP DIRECTLY FOR TELETYPES!!!

CUSTOM CABLE SERVICE AVAILABLE-CALL FOR PRICES

INTERACTION

NEWSLETTER
DETROIT
INTERACT
GROUP

DATE OCTOBER - NOVEMBER, 1980 VOLUME NO. I, no.5



INTERACTION FOR 1981 ?

As the end of the year quickly approaches, I am debating whether I want to continue the newsletter for another year. Running the newsletter has not been a really enjoyable experience and the amount of work I've been doing is hardly worth the small profit the newsletter is finally showing. Many people do not realize how a hobby publication operates. Too many think printing and postage are free and that ten dollars is an unreasonable amount to ask for the newsletter. Some subscribers think that ten dollars buys them a personal programming consultant available 24 hours a day, 7 days a week.

INTERACTION FOR 1981, cont.

The newsletter now has over 300 subscribers and the collating, stapling, labeling, and mailing takes a weeks worth of evenings to do each issue. The mail requiring personal answers take at least one evening's work each week. Combined with my other interests and community commitments, I now have very little free time. I can no longer enjoy turning on my Interact, since I constantly have programs to key in or correct for publication in Interaction.

Now that you've listened to my weeping and whining, let me tell you that I have tentiatively decided to publish Interaction for another year. The subscription will remain at ten dollars despite increased costs. There will be six issues in 1981. I cannot promise more realistically. In 1981 also, any subscriber can run free ads (up to 4 lines and based on space available) in the newsletter.

Again let me emphasize that Interaction is a hobby publication. That is, the newsletter and computers are my hobbies also. I have a full time job and I publish Interaction out of some misguided sense of commitment to the Interact owners who support and cotribute to the newsletter. In 1981, please help me help you by sending your reviews (Is anyone out there dissatified or better yet pleased with the quality of software and documentation on the market?), your programs or interesting subroutines, and any other information of interest to all Interact owners.

My final decision and information on 1981 subscriptions will be in next and final issue for 1980. Do not send renewals yet, but I promise that if I do continue with the newsletter, I will deliver at least as good a product next year as I have this year. The rest, and the decision of what it is worth is up to you.

MOVING AVERAGES

by Ross Campbell 2651 Manchester Rd. Ann Arbor, MI 48104

Here are two programs for calculating moving averages. The first program is for a 4 number moving average and the second is for any number of trails (input) up to 100.

```
  1   REM INTERACT LEVEL II BASIC
  5   PRINT " MOVING AVERAGE " : PRINT
 10   DIM D( 4 )
 20   FOR I = 1 TO 4
 30   INPUT D( I)
 40   NEXT I
 50   FOR I = 1 TO 4
 60   PRINT I ; SPC (2) ; D( I )
 70   NEXT I
 80   FOR I = 1 TO 4
 90   SUM = SUM + D( I )
100   NEXT I
110   PRINT "AVERAGE=" ; SUM / 4
120   INPUT "CHANGES" ; A$
130   IF A$ = "YES" THEN 200
140   END
200   INPUT "SEQ. NO." ; I
210   INPUT "NEW VALUE" ; D(I)
```

MOVING AVERAGE, cont.

```
  5   PRINT "MOVING AVERAGE" : PRINT
 10   DIM D(100)
 20   INPUT "NUMBER OF TRAILS" ; N
 30   FOR I = 1 TO N
 40   INPUT D(I)
 50   NEXT I
 60   FOR I = 1 TO N
 70   PRINT I ; SPC (2) ; D(I)
 80   NEXT I
 90   SUM = 0
100   FOR I = 1 TO N
110   SUM = SUM + D(I)
120   NEXT I
130   PRINT "AVERAGE=" ; SUM / N
140   INPUT "CHANGES" ; A$
150   IF A$ = "YES" THEN 200
160   END
200   INPUT "SEQ. NO." ; I
210   INPUT "NEW VALUE" ; D(I)
220   GOTO 60
```

FASTAD

by Phil Piatkowski

IN THE PADDLEBALL GAME, BREAKTHROUGH, FOUR(4) A/D CHANNELS ARE NEEDED:
- 2 FIRE BUTTONS
- 2 POTS

OF THESE, THE POTS REQUIRE ACCURATE MEASUREMENT AND THE BUTTONS REQUIRE ONLY >0 OR =0 MEASUREMENT.

THE BUTTONS CAN THEREFORE BE SHORT CYCLED THUSLY:
1) A RESET NEED ONLY BE 20 US OR GREATER
2) A CONVERT NEED ONLY BE 20 US OR GREATER

THE FOLLOWING STRATEGY CAN THEREFORE BE USED TO SPEED UP THE PADDLE RESOLUTION.

A) ASSUME AT INTERRUPT TIME A POT VALUE AT A/D CHANNEL N HAS BEEN CONVERTED IN HARDWARE AND THE ROM ROUTINE WILL READ AND STORE THE RESULT. THE ROM WILL THEN SET UP AND START THE CONVERSION OF A/D CHANNEL N-1 WHICH HAPPENS TO BE A FIRE BUTTON.

B) ADDITIONAL INTERRUPT CODE (IN RAM) CAN BE APPENDED TO THE ROM INTERRUPT CODE BY PLACING THE START ADDRESS OF THAT CODE IN LOCATION 5FF3H (USRINT). THE ROM WILL TEST FOR AND JUMP TO THE ADDITIONAL CODE.

C) IF THE A/D CHANNEL IS IMMEDIATELY READ BY THIS ADDITIONAL CODE, ONE OF THE FOLLOWING RESULTS WILL BE OBSERVED:
   1) THE A/D HARDWARE WILL BE STABLE (DONE) AND THE ACCUMULATED COUNT WILL BE VERY CLOSE TO ZERO, WHICH REFLECTS THE OFFSET ERRORS IN THE A/D CONVERTER. THIS INDICATES THE FIRE BUTTON IS DEPRESSED AND THE INPUT VOLTAGE IS ZERO.
   2) THE A/D HARDWARE WILL STILL BE ACTIVE BUT THE ACCUMULATED COUNT WILL ALREADY BE MUCH GREATER THAN ZERO. THIS INDICATES THE FIRE BUTTON IS OPEN AND THE INPUT VOLTAGE IS VREF. IT IS ONLY NECESSARY TO COMPARE THE ACCUMULATED COUNT READ FROM THE A/D HARDWARE WITH SOME MEDIAN VALUE (PROBABLY BETWEEN 16 AND 64) AND, DEPENDING ON A GREATER THAN OR LESS THAN RESULT, STORE A NUMBER OF FFH OR 00H IN THE APPROPRIATE LOCATION IN THE "A/D CONVERSION DATA BLOCK".

FASTAD, cont.

D) NOW THE A/D HARDWARE CAN BE INSTRUCTED TO CONVERT A POT CHANNEL WITH THE SEQUENCE:
   1) RESET CHANNEL (POT)
   2) WAIT 40 US (DO SOME BACKGROUND CODE?)
   3) START CONVERSION OF CHANNEL (POT)
   A COPY OF "3)" CAN BE PUT IN LOCATION 5FC0H (MIOREG) TO ALLOW INTERACT SYSTEM COMPATABILITY. THE ASSUMED CONDITIONS FOR "A)" HAVE NOW BEEN SATISFIED.

E) PERFORM A NORMAL RETURN FROM INTERRUPT.

F) SINCE A PAIR OF A/D CHANNELS ARE CONVERTED PER INTERRUPT, IT IS NECESSARY TO SYNCHRONIZE WHICH PAIR TO CONVERT TO ENSURE EVENNESS OF ACCESSABILITY. HERE, THE CLOCK VALUE AT 5FEFH (TICTIM) WAS TESTED FOR AN ODD/EVEN COUNT.

THERE IS ROOM FOR IMPROVEMENT. THE BASIC LIMITATIONS ARE THE HARDWARE TIMING REQUIREMENTS DESCRIBED AS FOLLOWS:

1) THE RESET FOR THE FIRE BUTTON MUST BE > 20 US. THE ROM A/D ROUTINE ENSURES THIS AND THEREFORE IS NOT SEEN AS A USER CONTROLLABLE OPTION.

2) THE CONVERSION OF THE FIRE BUTTON MUST BE > 20 US. THIS LIMIT IS PASSED BY THE TIME THE USER ROUTINE IS IN CONTROL AND AGAIN IS NOT SEEN AS A USER OPTION.

3) THE RESET FOR THE POT MUST BE > 40 US. HERE THE USER CAN BE DOING SOME BACKGROUND JOBS RATHER THAN WASTING TIME IN A DELAY LOOP.

4) THE CONVERSION OF THE POT MUST BE > 240 US. THIS LIMIT IS PASSED IN THE TIME STARTING FROM THE END OF THIS ROUTINE TILL THE NEXT CLOCK INTERRUPT AND IS AGAIN NOT SEEN AS A USER OPTION.

THE OBJECT IS TO KEEP THE HARDWARE GOING (I.E., GET THE FIRE BUTTON VALUE AS SOON AS POSSIBLE IN THE USER ROUTINE AND IMMEDIATELY GIVE THE HARDWARE A NEW COMMAND) AND DO ALL THE OTHER NECESSARY JOBS WHILE WAITING FOR THE HARDWARE TO RESET.

WHAT CAN YOU DO?

IS THERE A TOTALLY DIFFERENT APPROACH?

IN ANY CASE, EVEN IF YOU USE THE SUGGESTED CODE, HAVE FUN.

PHIL PIATKOWSKI

FASTAD, cont.

```
ASM80 FASTAD.S

ISIS-II 8080/8085 MACRO ASSEMBLER, V2.0         MODULE   PAGE    1

  LOC  OBJ         SOURCE STATEMENT

                   ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
                   ;FASTAD IS A ROUTINE TO SPEED UP THE A/D CONVERSION UPDATES IN THE INTERACT
                   ;  COMPUTER.
                   ; TYPICAL A/D UPDATES WERE SLOW BECAUSE ALL CHANNELS (SOME UNUSED) WERE MEASURED
                   ;  AT A RATE OF 1 CONVERSION PER 1/60 SEC.
                   ; THIS ADD-ON TO THE USER INTERRUPT OPTION SELECTS ONLY THE 4 USED CHANNELS AND
                   ;  CONVERTS 2 (1 FIRE BUTTON & 1 POT. ) PER 1/60 SEC.
                   ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
                   ; THE CLOCK INTERRUPT COUNT 'TICTIM' IS TESTED FOR ODD OR EVEN COUNT TO SET ONE
                   ;  OF THE FOLLOWING MODES;
                   ;  EVEN:  CUT SHORT THE CONVERSION OF CHANNEL 4 (RIGHT FIRE BUTTON) SIGN EXTEND THE
                   ;       RESULT & STORE IN APPROPRIATE MEMORY.
                   ;         RESET A/D & START CONVERSION OF CHANNEL 2 (LEFT POT).
                   ;         RETURN LETTING THE INTERRUPT A/D ROUTINE IN ROM GET CHANNEL 2 & START UP
                   ;       CHANNEL 1 (N-1).
                   ;  ODD:   CUT SHORT THE CONVERSION OF CHANNEL 1 (LEFT FIRE BUTTON), SIGN EXTEND THE
                   ;       RESULT & STORE IN APPROPRIATE MEMORY.
                   ;         RESET A/D & START CONVERSION OF CHANNEL 5 (RIGHT POT).
                   ;         RETURN LETTING THE INTERRUPT A/D ROUTINE IN ROM GET CHANNEL 5 & START UP
                   ;       CHANNEL 4 (N-1).
                   ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
                   ; TO USE, THE CODE IS APPENDED TO A GAME TAPE WITH WHATEVER MEANS AVAILABLE.
                   ;  ALTERNATELY, IT CAN BE LOADED IN AS A SEPARATE TAPE IN ADDITION TO THE GAME TAPE.
                   ;  IT CAN BE CONSIDERED STAND ALONE CODE WHICH EXECUTES INVISIBLY TO THE GAME PROGRAM
                   ;  BUT THERE ARE SOME CAUTIONS:
                   ;       1) THIS CODE DOES TAKE UP A LITTLE EXTRA TIME TO EXECUTE AND THERE IS A CHANCE
                   ;          THAT THE GAME PROGRAM CANNOT AFFORD THIS TIME.
                   ;       2) IF THE GAME PROGRAM ALSO USES THE USER INTERRUPT OPTION 'USRINT',
                   ;          THE RETURN JUMP AT LOCATION 622EH MUST BE CHANGED TO GO TO THE GAME
                   ;          INTERRUPT ROUTINE.
                   ; TO DATE THIS HAS BEEN APPLIED TO THE GAME PROGRAMS "BREAKTHROUGH" AND "VOLLEY BALL".
                   ;  BOTH SHOW PADDLE SMOOTHNESS IMPROVEMENT BUT BREAKTHROUGH HAD IRRATIC BALL MOVEMENT
                   ;  IN THE 'FAST' GAME. (A POSSIBLE RESULT OF FAULT CONDITION 1)
                   ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

  5FF3             USRINT  EQU    5FF3H         ;USER INTERRUPT JUMP TABLE
  5FC0             MIOREG  EQU    5FC0H         ;COPY OF I/O OUTPUT COMMAND
  3000             IOREG   EQU    3000H         ;A/D I/O PORT
  5FF6             ADXBLK  EQU    5FF6H         ;TABLE OF A/D VALUES
  0172             ITRXIT  EQU    0172H         ;DEFAULT JUMP TO FOR NORMAL EXIT
  5FEF             TICTIM  EQU    5FEFH         ;CLOCK INTERRUPT COUNT

  5FF3                     ORG    USRINT        ;SET INTERRUPT JUMP TABLE
  5FF3 0062                DW     START

  6200                     ORG    6200H

  6200 3AEF5F      START:  LDA    TICTIM        ;GET CLOCK LSB
  6203 E601                ANI    1             ;ODD OR EVEN ?
  6205 3AC05F              LDA    MIOREG        ;GET COPY OF LAST I/O COMMAND
  6208 CA3162              JZ     EVEN          ;JUMP IF EVEN
                                          ;ELSE SET UP TO READ CH 1 & START CH 5
```

```
 LOC  OBJ          SEQ          SOURCE STATEMENT

 620B E607          53          ANI     7               ;MASK OFF ALL A/D COMMAND BITS
 620D F668          54          ORI     68H             ; & MASK IN A NEW COMMAND (RESET CH 5)
 620F 47            55          MOV     B,A             ;SAVE TEMPORARILY
 6210 EECO          56          XRI     OCOH            ; GENERATE ANOTHER COMMAND (CONV CH 5)
 6212 11F75F        57          LXI     D,ADXBLK+1      ;POINT TO CH 1 STORAGE CELL
 6215 32C05F        58 COMMON:  STA     MIOREG          ;STORE NEW I/O COMMAND FOR THE ROM A/D CONV.
 6218 4F            59          MOV     C,A             ;SAVE TEMPORARILY
 6219 210030        60          LXI     H,IOREG         ;POINT TO PORT (FOR FAST ACCESS)
 621C 7E            61          MOV     A,M             ;CUT SHORT FIRE BUTTON CONVERSION, GET VALUE
 621D 70            62          MOV     M,B             ;RESET FOR POTS
 621E FE32          63          CPI     50              ;SIGN EXTEND FIRE BUTTON VALUE
 6220 3E00          64          MVI     A,0
 6222 FA2662        65          JM      ALZERO
 6225 2F            66          CMA
 6226 12            67 ALZERO:  STAX    D               ;STORE IN SELECTED CELL FOR FIRE BUTTON
 6227 3E04          68          MVI     A,4             ;TIME DELAY FOR COMPLETE A/D RESET
 6229 3D            69 DEL1:    DCR     A
 622A C22962        70          JNZ     DEL1
 622D 71            71          MOV     M,C             ;START CONVERSION OF POT VALUE
 622E C37201        72          JMP     ITRXIT          ; & RETURN TO EXIT ROUTINE
                    73
                    74
                    75 EVEN:                    ;SET UP TO READ CH 4 & START CH 2
 6231 E607          76          ANI     7               ;MASK OFF ALL A/D COMMAND BITS
 6233 F650          77          ORI     50H             ; & MASK IN NEW COMMAND (RESET CH 2)
 6235 47            78          MOV     B,A             ;SAVE TEMPORARILY
 6236 EECO          79          XRI     OCOH            ;GENERATE ANOTHER COMMAND (CONV CH 2)
 6238 11FA5F        80          LXI     D,ADXBLK+4      ;POINT TO CH 4 STORAGE CELL
 623B C31562        81          JMP     COMMON          ;CONTINUE
                    82
                    83          END

PUBLIC SYMBOLS


EXTERNAL SYMBOLS


USER SYMBOLS
ADXBLK A 5FF6    ALZERO A 6226    COMMON A 6215    DEL1   A 6229    EVEN   A 6231    IOREG  A 3000    ITRXIT A 0172
MIOREG A 5FC0    START  A 6200    TICTIM A 5FEF    USRINT A 5FF3

ASSEMBLY COMPLETE,  NO ERRORS
```

COLOR MASTERMIND

by John Bracey 1468 N. Fourth Columbus, OH 43201

A color graphics version of mastermind. The game uses three colors with the display blocks drawn by a machine language sub routine. The three colors are red, black, and green. Use the corresponding keys for responses. R - red B - black G -green You have 5 guesses to get it right.

```
 1 REM WRITTEN BY JOHN + CLAUDETTE BRACEY
 2 REM COLUMBUS OHIO
 3 DATA "B","R","G"
 5 REM DECLARATIONS
 8 DIM W$(2)
 10 DIM C$(3)
 15 DIM M$(5,2)
 20 DIM G$(5,1)
 30 REM INITIALAZATIONS
 35 FOR I=1 TO 3
 38 READ C$(I)
 43 NEXT I
 45 POKE 19215,25
 48 GOSUB 200
 50 LET U=0
 55 FOR I=0 TO 5
 60 FOR J=0 TO 2
 65 LET M$(I,J)=" "
 70 NEXT J
 75 LET G$(I,0)=0
 80 LET G$(I,1)=0
 85 NEXTI
 90 LET TV=16640
 100 REM PRINT SECTION
 105 PRINT "MASTER MIND"
 110 PRINT "DO YOU WANT RULES"
 112 PRINT "Y OR N"
 114 LET A$=INSTR$(1)
 116 IF A$="N" THEN 1000
 118 PRINT "THE COMPUTER WILL"
 120 PRINT "SELECT A PATTERN"
 122 PRINT "OF 3 COLORS"
 124 PRINT " THE COLORS ARE:"
 125 GOSUB 500
 126 PRINT "     R-RED"
 128 PRINT "     G-GREEN"
 130 PRINT "     B-BLACK"
 139 GOSUB 500
 144 PRINT "AT ANY TIME"
 145 PRINT "YOU MAY "
 148 PRINT "TYPE @ TO QUIT,"
 152 PRINT "H(ELP) FOR RULES,"
 158 GOSUB 500
 160 PRINT "THEN RE-ENTER"
 162 PRINT "YOUR 4-LETER GUESS"
 167 IF U=1 THEN 1510
 170 GOTO 1000
```

COLOR MASTERMIND, cont.

```
200 REM SET UP SUBROUTINE
210 DATA 1,6,93,195,162,5
215 DATA 4,5,2,0,0
220 POKE 19473,0
230 POKE 19474,93
240 FOR L=23808 TO 23818
250 READ M
260 POKE L,M
270 NEXT L
280 RETURN
500 PRINT "ANY KEY TO GO-ON"
505 LET A$=INSTR$(1)
510 RETURN
1000 PRINT
1130 REM BUILD HIDDEN CODE
1140 FOR I=0 TO 2
1150 LET J=INT(3*RND(1))+1
1160 LET M$(0,I)=C$(J)
1165 NEXT I
1167 WINDOW 30
1169 COLOR 7,0,1,2
1171 CLS
1173 OUTPUT "MASTER MIND",30,76,1
1175 REM START GUESSING ROUTINE
1180 LET G=1
1510 PRINT "ENTER GUESS"
1530 FOR I=0 TO 2
1540 LET U=1
1550 LET A$=INSTR$(1)
1570 IF A$="R" THEN 118
1590 IF A$="Q" THEN 3060
1610 REM TEST FOR GOOD GUESS
1620 FOR J=1 TO 3
1630 IF A$=C$(J) THEN 1680
1640 NEXT J
1645 PRINT
1650 PRINT "ILLEGAL COLOR ";A$
1670 GOTO 1510
1680 LET M$(G,I)=A$
1683 LET SC=66
1685 GOSUB 6000
1690 NEXT I
1699 PRINT
1900 REM TEST CORRECT POSITION
1910 FOR I=0 TO 2
1920 IF M$(G,I)<>M$(0,I) THEN 1940
1930 LET G$(G,1)=G$(G,1)+1
1940 NEXT I
2000 REM TEST FOR CORRECT COLOR
2010 FOR I=0 TO 2
2020 LET W$(I)=M$(0,I)
2030 NEXT I
2040 FOR I=0 TO 2
2050 FOR J=0 TO 2
2060 IF M$(G,I)=W$(J) THEN 2090
2070 NEXT J
```

COLOR MASTERMIND, cont.

```
2080 GOTO 2110
2090 LET GS(G,0)=GS(G,0)+1
2100 LET WS(J)="X"
2110 NEXT I
2300 REM PRINT A GUESS
2360 GOSUB 4000
2400 IF GS(G,0)=3 AND GS(G,1)=3 THEN 3010
2410 LET G=G+1
2420 IF G=6 THEN 3060
2430 GOTO 1510
3000 REM PLAYER WIN
3010 PRINT "YOU FIGURED ME OUT"
3013 PRINT "RAT'S !!!"
3015 WINDOW 77
3020 PRINT "WANT TO PLAY AGAIN"
3025 PRINT "Y OR N"
3030 LET AS=INSTRS(1)
3040 IF AS="Y" THEN 50
3050 GOTO 9000
3060 LET G=0
3070 GOSUB 4000
3080 OUTPUT "SOLUTION",40,G1,1
3085 FOR I=1 TO 1000:NEXT I
3090 WINDOW 77
3100 PRINT "I OUTSMARTED YOU"
3110 PRINT "HA HA HA !!!"
3210 GOTO 3020
4000 LET T1=TV
4050 FOR X=1 TO 2
4100 FOR I=0 TO 2
4200 FOR J=1 TO 3
4300 IF MS(G,I)=CS(J) THEN 4500
4400 NEXT J
4500 REM PRINT COLORS
4550 LET CL=(16*J)+(64*J)
4600 POKE T1,CL
4700 LET T1=T1+2
4800 NEXT I
4900 LET T1=TV+32
4950 NEXT X
4955 IF G=0 THEN G=6
5000 LET G1=69-(6*(G-1))
5100 IF G=6 THEN 5500
5200 OUTPUT GS(G,0),51,G1,1
5250 OUTPUT "COL=",34,G1,1
5300 OUTPUT GS(G,1),82,G1,1
5350 OUTPUT "POS=",65,G1,1
5400 LET TV=TV+192
5500 RETURN
6000 REM POKE COLOR AND POSITION
6010 POKE 23816,J
6015 POKE 23817,SC
6020 LET H=10+(I*6)
6030 POKE 23818,H
6040 H=USR(0)
6050 RETURN
9000 WINDOW 77
9100 END
```

On Greater Accuracy for Interact Real Time Clock

by Richard Pasco
235 College Ave.
Mountain View, Calif. 94040

The real time clock mentioned in *The Book of Interact Programs* and in Vic Volkman's article in *Interaction No. 2* is not very accurate, and here's why.

1. The text incorrectly states that the data in location 24559 is incremented at 60 Hz. This is only approximate; actually it is incremented by an interrupt (by examining the schematic) at the NTSC vertical sweep rate, 0.1% slower than 60 Hz. Below are the calculations which give the actual rate:

The color subcarrier is 315/88 = 3.57954545 MHz.
There are 455 cycles of the subcarrier for every 2 horizontal sweeps.
Thus, the horizontal sweep rate is 2250/143 = 15.7342657 kHz.
There are 525 cycles of this sweep for every 2 vertical sweeps.
Thus, the vertical sweep rate is 60000/1001 = 59.940060 Hz.

A clock based on the assumption that the vertical sweep rate were exactly 60 Hz would run slow by a factor of 1000/1001; it would loose 3.6 seconds per hour or more than 10 minutes per week.

2. The text incorrectly claims that the data in location 24560 is incremented every 4 seconds. This approximation is very rough; actually it is incremented by overflow of the data in location 24559, every 256 vertical sweeps, or 4.2709333 seconds.

3. The BASIC code runs asynchronously with respect to the interrupt level code, and can not POKE into the real-time-clock counter without destroying its accuracy. In particular, any code which looks like this won't work:

```
10  CK = 24559
20  IF PEEK(CK) > 58 THEN POKE CK,0: GOTO 40
30  GOTO 20
40  ... your code ...
50  GOTO 20
```

First, suppose that (CK) had just been incremented to 59 when statement 20 were begun. It would be immediately replaced by 0, without waiting for the next interrupt, and upon the interrupt would advance to state 1. Thus, the total amount of time spent in state 59 and state 0 would be one interrupt cycle, and the statement 40 would sometimes be executed after 59 interrupts, not 60 as intended, and the clock would be fast.

Changing the constant in statement 20 from 58 to 59 would not be correct, either. BASIC spends nonzero time interpreting the statement, and sometimes an extra interrupt occurs after the PEEK but before the POKE. When this happens, if the constant were 59, statement 40 would be executed after 61 interrupts. The clock would then be slow.

Which of the two effects above is more severe, and whether 58 or 59 produces more accurate results, depends on the speed of BASIC. But in any event, no better accuracy than one part in 60 could be guaranteed with the code above. A far better technique is to never POKE into CK but only to PEEK into it. The code below illustrates the technique.

```
10  CK = 24559
15  T = 0
20  IF (PEEK(CK)-T AND 255) < 60 THEN 20
30  T = T+60 AND 255
40  ... your code ...
50  GO TO 20
```

GREATER ACCURACY, cont.

This code advances threshold T (circularly, modulo 256) and tests to see if (CK) has passed it. Each time (CK) passes the threshold, a new threshold is set for 60 interrupts later. Since slow BASIC never touches the clock but only watches it, the accuracy of this technique is limited only by the effects discussed in section 1. Even these effects could be corrected (to crystal accuracy) by advancing the clock by 1 second after it has counted 1000 seconds.

Finally, you should be aware that the TONE statement disables interrupts during its execution. (If this were not done, there would be a 60Hz frequency-modulated component in the software-generated sound.) Use of the TONE statement to make the clock "tick" will cause it to miss interrupts and run slow.

MERRY OLDSMOBILE REVISITED

by John Worrall 15 Main Street Avonmore, Ontario, CANADA K0C 1C0

This program is a modification of the Merry Oldsmobile given earlier, which shows how to print information on the screen (one method) and how to produce a machine-language program which will run on the Interact with no other support. Owners who have Computacolor can design their own graphics and Load them into the monitor, or for the masochist, screen memory data are provided for such a picture.

Program notes: 4D30-44 call RFILL as described in the Monitor documentation, to produce a colorful screen wipe. 4D50-68 is a loop which examines address 3805H until a P (FBH) is entered. 4D60-68 calls TXTOUT, which requires the BC pair to be loaded with the address (4D70) where the text info starts. You must define colour, Y coördinate and X coördinate first, then ASCII codes in hex. TXTOUT will display these until it encounters a NUL (ØØ) at which point you can reset colour, Y, & X, or end the subroutine.

The program can now be easily adapted to a free-standing code which will load and run on its own. Draw an appropriate picture with Computacolor, save on tape, then Load into Monitor with a Partial screen command in effect. Then Substitute 4C00 to 4C05 as listed to change the JMP address. Write 4000,5E4D to tape and you have a simple program c/w graphics and prompts.

MERRY OLDSMOBILE, cont.

```
4D30-LXI B     01
  31-          40
  32-          4D
  33-CALL      CD
  34-RFILL     A2
  35-          05
  36-JMP       C3
  37-          60
  38-          4D
   *
  40-          4D
  41-          70
  42-          02
  43-          00
  44-          00
   *
4D50-LXI H     21
  51-          05
  52-          38
  53-MOV A,M   7E
  54-CPI       FE
  55-          FB
  56-JZ        CA
  57-          00
  58-          5E
  59-JMP       C3
  5A-          50
  5B-          4D
   *
4D60-LXI B     01
  61=          70
  62-          4D
  63-CALL      CD
  64-TXTOUT    D3
  65-          01
  66-JMP       C3
  67-          50
  68-          4D
   *
4D70-COLOUR    70
  71-Y COORD.  12
  72-X COORD.  23
  73-"P"ASCII  50
  74-U         55
  75-S         53
  76-H         48
  77-SPC       20
  78-"         22
  79-P         50
  7A-"         22
  7B-RESET     00
  7C-COLOUR    70
  7D-Y COORD.  26
  7E-X COORD.  24
  7F-"T"ASCII  54
4D80-O         4F
```

```
4D81-SPC       20
  82-P         50
  83-L         4C
  84-A         41
  85-Y         59
  86-:         3A
  87-RESET     00
  88-END       FF
   *
5E00-LXI H     21
  01-          1C
  02-          5E
  03-MOVA,M    7E
  04-CPI       FE
  05-          FF
  06-JZ        CA
  07-          04
  08-          4C
  09-MVI D     16
  0A-          25
  0B-DCR B     05
  0C-JNZ       C2
  0D-          10
  0E-          5E
  0F-MOV B,M   46
5E10-DCR C     0D
  11-JNZ       C2
  12-          0B
  13-          5E
  14-DCR D     15
  15-JNZ       C2
  16-          0B
  17-          5E
  18-INR L     2C
  19-JMP       C3
  1A-          03
  1B-          5E          MUSIC DATA
  1C-39,40,45,45,72,45,45,72,45,
     45,45,45,40,45,4C,4C,66,4C,
     4C,66,4C,4C,4C,4C,00,4C,55,
     5A,5A,39,39,3D,39,33,33,39,
     45,4C,55,4C,4C,50,4C,4C,45,
     55,55,55,55,
5E4D-FF
```

```
4C00-LXI SP 31
  01-       80
  02-       7F
  03-JMP    C3
  04-       30
  05-       4D
```

MERRY OLDSMOBILE, cont.

For those without Computacolor, or who just like to punish themselves, here is the screen memory code for a simple picture. Load monitor, set Partial 2, Fill 4ØØØ,49FF,Ø and Substitute as follows (watch the addresses below carefully, vast amounts of Ø's have been omitted):

```
42A8 80 AA AA AA
  AC A8 AA AA 2A
  B0 AA AA 0A 00
42C8 A0 AA AA AA
  CC AA AA AA AA
  D0 AA AA AA 00
42E8 A8 AA AA AA
  EC AA AA AA AA
  F0 AA AA AA 0A
4308 F0 00 00 00
  0C 03 00 00 C0
  10 00 00 00 03
4328 30 03 00 00
  2C 03 00 00 C0
  30 00 00 C0 0C
434C 03 00 00 C0
4346 00 00 30 F6
  50 00 00 3F 30
4368 30 00 3F 00
  6C 03 00 00 C0
  70 00 FC 00 C0
4388 30 00 C0 03
  8C 03 00 00 C0
  90 F0 03 0C 00
4394 03 00 00 00
43A8 B0 00 00 3C
  AC 03 00 00 C8
  B0 0C 00 03 04
  B4 0C 01 00 00
43C8 A0 02 00 E8
  CC 03 00 00 EA
  D0 03 C0 00 95
  D4 BA 1A 00 00
43E8 AA AA AA AA
  EC 6A AA 9A AA
  F0 02 30 A3 95
  F4 EA 18 00 00
4408 AA AA AA AA
  0C 6A AA 99 AA
  10 C2 0C A8 AA
  14 AA 1B 00 00
441C 30 00 30 E6
4424 00 00 00 80
  28 AA AA AA AA
  2C 6A AA 9A AA
  30 EA 00 A8 AA
  34 AA E1 00 00
4444 00 00 00 80
  48 5A 55 A9 AA
  4C 6A AA 9A AA
  50 BA 00 A8 AA
  54 AA 56 01 00
445C 30 DA 6F 04
4464 00 00 00 A0
  68 06 FF A4 AA
  6C 6A AA 9A AA
  70 AE 00 A8 AA
4474 AA FD 00 00
4484 00 00 00 80
  88 C1 00 93 AA
  8C AA 55 A5 AA
  90 AA AA AA AA
  94 5A 00 03 00
44A4 00 00 00 40
  A8 30 00 4C AA
  AC AA AA AA AA
  B0 AA AA AA AA
  B4 35 00 0C 00
44C8 0C 00 30 A9
  CC AA AA AA AA
  D0 AA AA AA 6A
  D4 0D 00 30 00
44E8 0C 3C 30 A4
  EC AA AA AA AA
  F0 AA AA AA 1A
  F4 0C 3C 30 00
4508 0C 3C 30 50
  0C 55 55 55 55
  10 55 55 55 05
  14 0C 3C 30 00
4528 0C 00 30 00
4534 0C 00 30 00
4548 30 00 0C 00
4554 30 00 0C 00
4568 C0
456A 03
4574 C0
4576 03
4589 FF
4595 FF
```

INTERACTION ON TAPE

Because of the calls and letters claiming errors in the Basic programs, although every program is checked before publication and the listings are printouts of the actual programs, I am now making available Interaction programs on tape. Individual issues are available on a data cassette (all the Basic programs only) for $3.00 including postage. Also you can avoid typing programs by just purchasing the tape. Please send check to Stephen Cook 15356 Prevost Detroit, MI 48227 and specify issue number.

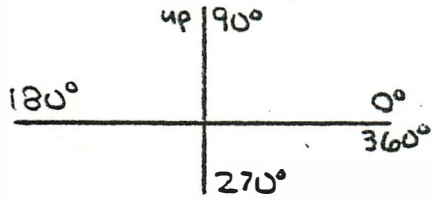
LUNAR LANDER

by Kevin TenBrook 8701 Town Park #3166 Houston, TX 77036

The display of this program shows a scale map (1 pixel = 500Km) of the Earth-Moon system as well as other important information such Earth and Lunar altitude, velocity, direction, minutes into mission, etc.

Caution: This game can take up to 50 minutes to play. You start out in Earth orbit and the goal is to land on the Moon. (This is extremely difficult to do. In the year since I wrote this program, I have never won.) Intermediate goals can be: 1. Get within 4000 Km of the Moon. (The display changes scale at this point to show only the Moon @ 1 pixel = 300 Km. Also your ship no longer blinks on the screen) 2. Acheive Lunar orbit. This is relatively difficult, but many times easier than landing. Some miscellaneous notes about the program.

1. After typing RUN there is a few seconds delay while program calculates
2. Angle



3. Data is given in the following units:
   SPEED: Km / min — ANGLE: degrees
   DISTANCE: Km — FUEL: minutes of burn time
   TIME: min
4. Way to win: Acheive Lunar altitude = Ø Km and velocity ≤ 6 Km / min
5. Ways to lose: Run out of air, leave Earth-Moom system, crash on Earth, crash on Moon, run out of fuel.

```
1 REM ** LUNAR LANDING GAME -BY KEVIN TENBROOK, 1979 **
5 WINDOW12
10 CLS
15 DIMP(45),Q(45)
20 DIME(13),F(13)
30 DATA17,17,17,18,18,18,19,19,19,93,94,94,94,95
40 DATA37,38,39,37,38,39,37,38,39,38,37,38,39,38
41 FORZ=8TO360STEP8
42 P(Z/8)=6*COS(Z*3.14159/180)+56
43 Q(Z/8)=6*SIN(Z*3.14159/180)+38
44 NEXTZ
50 FORI=0TO13
60 READE(I):NEXTI
70 FORJ=0TO13
80 READF(J):NEXTJ
85 FL=100
86 CT=0
87 ALTA=20000:LATL=360000
90 EX=90000:EY=190000:MX=470000:GE=1436400000:GM=17820000
100 X=110000:Y=EY:VX=0:VY=350:T=0:AN=90:BA=90
110 FORI=0TO13
120 PLOTE(I),F(I),1
130 NEXTI
```

LUNAR LANDER, cont.

```
140 ALTA=ALTA-6350
150 LATL=LATL-1720
151 IFALTA<0GOTO2000
152 IFLATL<0GOTO3000
155 IFCT=1GOTO170
160 PLOTX/5000,Y/5000,2
170 OUTPUT"EARTH ALT:",5,70,2
180 OUTPUT"LUNAR ALT:",5,64,2
190 OUTPUTALTA,60,70,1
200 OUTPUTLATL,60,64,1
210 OUTPUT"VELOCITY:",5,58,2
220 OUTPUTSQR(VX^2+VY^2),60,58,1
230 OUTPUT"DIRECTION:",5,24,2
240 OUTPUTAN,60,24,1
250 OUTPUT"TIME:",5,18,2
260 OUTPUTT,30,18,1
270 OUTPUT"FUEL:",60,18,2
280 OUTPUTFL,84,18,1
290 INPUT"FIRE ENGINES";A$
300 FA=0:IFA$="Y"GOTO320
310 GOTO330
320 FA=37:INPUT"ANGLE";BA
330 INPUT"HOW LONG";K
332 IFCT=0THENPLOTX/5000,Y/5000,0
333 IFK<=0GOTO330
340 FORI=1TOK
350 ALTA=SQR((EX-X)^2+(EY-Y)^2):LATL=SQR((MX-X)^2+(EY-Y)^2)
360 AX=(EX-X)*GE/ALTA^3+(MX-X)*GM/LATL^3+FA*COS(BA*3.14159265/180)
370 AY=(EY-Y)*GE/ALTA^3+(EY-Y)*GM/LATL^3+FA*SIN(BA*3.14159265/180)
371 IFLATL<=1720GOTO3000
372 IFALTA<=6350GOTO2000
380 VX=VX+AX:VY=VY+AY:X=X+VX:Y=Y+VY
381 IFX<1GOTO1000
382 IFY<1GOTO1000
383 IFCT=1GOTO386
384 PLOTX/5000,Y/5000,1
385 PLOTX/5000,Y/5000,0
386 SOUND3,600
387 SOUND7,4096
390 NEXTI
400 AN=ATN(VY/VX)*180/3.14159265
410 IFVY*AN<0THENAN=AN+180
420 IFAN<0THENAN=AN+360
430 T=T+K
440 IFFA=37THENFL=FL-K
450 IFX/5000>112GOTO1000
460 IFY/5000>77GOTO1000
470 IFX<0GOTO1000
480 IFY<0GOTO1000
510 IFFL<=0GOTO4000
520 IFT>3000GOTO5000
530 IFLATL<5720GOTO6000
540 CT=0
550 CLS
560 GOTO110
```

LUNAR LANDER, cont.

```
1000 WINDOW77
1010 CLS
1020 PRINT"YOU HAVE LEFT THEVICINITY OF THE  EARTH-MOON SYSTEMAND WILL NEVER
1030 PRINT"RETURN."
1040 GOTO7000
2000 WINDOW77
2010 CLS
2030 IFSQR(VX^2+VY^2)>1GOTO2060
2040 PRINT"YOU HAVE LANDED  IN CENTRAL PARK."
2050 GOTO7000
2060 PRINT"YOU HAVE BURNED  UP IN THE ATMOS- PHERE."
2070 GOTO7000
3000 WINDOW77
3010 CLS
3020 IFSQR(VX^2+VY^2)>6GOTO3050
3030 PRINT"CONGRATULATIONS! YOU HAVE LANDED  ON THE MOON!"
3040 GOTO7000
3050 PRINT"YOU HAVE CRASHED ON THE LUNAR     SURFACE."
3060 GOTO7000
4000 CLS
4010 WINDOW77
4020 PRINT"YOU HAVE RUN OUT OF FUEL AND WILL COAST FOREVER IN SPACE."
4030 GOTO7000
5000 WINDOW77
5010 CLS
5020 PRINT"YOU HAVE USED UP YOUR OXYGEN SUP- PLY AND HAVE     SUFFOCATED"
5030 GOTO7000
6000 CT=1
6010 CLS
6020 FORL=1TO44
6030 PLOTP(L),Q(L),1
6040 NEXTL
6050 PLOT(X-453200)/300,(Y-178600)/300,2
6060 OUTPUT"M",54,40,2
6070 GOTO140
7000 END
```

BOUNCY KEYBOARD ? TRY THIS !

by Thom Linehan 410 Normandy Royal Oak, MI 48073

According to Interact's schematics, capacitors C47-54 exist. According to my Interact's main circuit board, they don't. However, the pads and silkscreen for these eight capacitors (ceramic, .001, 12V) do exist.

Upon inspection of the front edge of the circuit board, one will find 16 unused landings. It appears as though Interact just chose not to stuff these locations. Being a curious person, I stuffed them.

Guess what? A good number of the keys on the keyboard wouldn't function. These additional capacitors (as there are eight more in parallel) slowed down the keying too much. As Interact probably also found out.

However, substitute 47 - 100 pf values for the .001 uf and you'll have a more stable keyboard. You may also find your joysticks less touchy.

TAPE MOTOR CONTROL

by Ken Stuempges 6261 N.Joyce Milwaukee, WI 53225

The following routine may be used to input pre-recorded tape messages to the television speaker, in sync with print statements.

According to Interact's memory map, address 1000_H is the register used for controlling the tape drive motor (along with some color commands). This program utilizes a POKE command to turn on the proper bit to start the tape drive. The FOR - NEXT statement is for a programmable delay to allow the tape message to be played. A few dummy PRINT statements prior to the actual PRINT "_____" command allow the tape to get started. Some experimentation will be required to get the timing right.

Load the program, push the 'read' button, and RUN.

```
10   POKE 19215, 25
20   POKE 4096, 68
22   PRINT
23   PRINT
25   PRINT "message "
30   FOR M = 1 TO 1500
40   NEXT M
50   POKE 4096, 00
```

Statement 20 starts the tape. 4096 is the register we must POKE. The value should be 68 or greater. (Changing the value will also affect the colors.) Statements 22 &23 allow the tape to start. (Use more PRINTS as required.) Statement 30 and 40 is the delay to allow the tape to run and play the message. Statement 50 stops the tape.

BIG LETTERS

By Bob Draganski 14301 Harrison Livonia, MI 48154

This short program can be used as a sub-routine in pre school educational programs. It displays large letters (25 x 25 pixels) in Interact format.

```
5 PRINT:PRINTCHR$(8);
10 CLS
20 COLOR4,4,0,7
100 A$=INSTR$(1)
110 CLS
120 OUTPUTA$,0,5,1
200 FORY=6TO1STEP-1
210 FORX=5TO0STEP-1
220 IFPOINT(X,Y)=1THENGOSUB1000
230 NEXT:NEXT
900 GOTO100
1000 REM BIG LETTERS
1020 XB=5*X+40
1040 YB=5*Y+30
1060 OUTPUTCHR$(1),XB,YB,3
1900 RETURN
```

FAST FOURIER TRANSFORM

by William Adams 411 S.W. Bradway Palm Bay, FL 32905

A Fourier transform yields an analysis of a complex waveform resulting in a set of simple additive sinusoidal components. For more information and for an introduction to FFT see the July, 1980 issue of Creative Computing.

```
5 REM-EVALUATES THE FFT OF A PULSE TRAIN WITH USER-SPECIFIED DUTY CYCLE
6 REM-CAN BE MODIFIED TO YIELD FFT OF A USER-SUPPLIED FUNCTION
7 PRINT"MAX M=7"
10 INPUT"ENTER M= LOG2(N)";M
12 N=2^M
13 FF=0.434295
15 DIM A(N,2)
20 REM:DC% PULSE
21 INPUT"ENTER DUTY CYCLE";DC
25 FOR Z=1 TO N*DC
30 A(Z,1)=1
35 NEXT Z
37 INPUT"ENTER 1 TO PRINT     THE  TIME    FUNCTION, ELSE      ENTER 0";P
38 INPUT"ENTER 1 TO PRINT     THE  FREQ    FUNCTION, ELSE      ENTER 0";Q
39 IF P=0 GOTO 55
40 FOR K=1 TO N
45 PRINT K;A(K,1);A(K,2)
50 NEXT K
55 REM
340 GOSUB 500
350 GOSUB 400
360 GOSUB 450
369 PRINT FRE(0)
370 STOP
400 REM:PRINT SUBROUTINE
405 FOR K=1 TO N
406 XX=SQR(A(K,1)^2+A(K,2)^2)/N
407 IF Q=1 THEN PRINT K;XX
408 IF XX=0 THEN YY=-77:GOTO 414
410 YY=10*FF*LOG(XX)
414 A(K,1)=YY
415 NEXT K
420 RETURN
450 REM:PLOT SUB-ROUTINE
452 CLS
453 IF N>117 THEN N=117
455 FOR J=1 TO N
456 ZZ=77-ABS(A(J,1))
457 IF ZZ<0 THEN ZZ=0
458 COLOR 4,7,3,0
460 FOR K=1 TO ZZ
465 PLOT J,K,2
470 NEXT K
473 NEXT J
475 RETURN
```

FAST FOURIER TRANSFORM, cont.

```
500 REM:BEGIN FFT SUBROUTINE
505 PI=3.141573
510 N2=N/2
520 NM1=N-1
530 J=1
540 FOR I=1 TO NM1
550 IF(I>=J)THEN GOTO 610
560 T1=A(J,1):T2=A(J,2)
570 A(J,1)=A(I,1)
580 A(J,2)=A(I,2)
590 A(I,1)=T1
600 A(I,2)=T2
610 K=N2
620 IF(K>=J)THEN GOTO 650
630 J=J-K:K=K/2
640 GOTO 620
650 J=J+K
655 NEXT I
660 FOR L=1 TO M
670 LE=2^L
680 L1=LE/2
690 U1=1:U2=0.
700 W1=COS(PI/L1)
710 W2=SIN(PI/L1)
720 FOR J=1 TO L1
730 FOR I=J TO N STEP LE
740 IP=I+L1
750 T1=A(IP,1)*U1-A(IP,2)*U2
760 T2=A(IP,1)*U2+A(IP,2)*U1
770 A(IP,1)=A(I,1)-T1
775 A(IP,2)=A(I,2)-T2
780 A(I,1)=A(I,1)+T1
790 A(I,2)=A(I,2)+T2
800 NEXT I
810 X=U1*W1-U2*W2
820 Y=U1*W2+U2*W1
830 U1=X:U2=Y
835 NEXT J
837 NEXT L
840 RETURN
900 END
OK
```

THANK YOU

The listings and the front page graphic for this issue were done by Bob Noel of Detroit, MI. We used his Slagh System Services equipped Interact and a KSR-33 Teletype.

PLANETARY ORBITS

by Harry Holloway PO box 2263 Ann Arbor, MI 48106

A program to plot the position of the planets in their respective orbits. This program was a big attraction when demonstrated at the Midwest Computer Show in Chicago in October. After plotting the initial positions, moving the joystick forward moves the planets forward in time the interval set by the pot control. Pulling the joystick back moves the display back in time. Pushing the joystick to the left returns the program to the menu.

```
10 REM"ORRE", 4/80
20 K=.0174533: CLS: COLOR0, 7, 3, 4: DIMMD(12), NM$(12)
30 FORJ=1TO9: READM0(J), MU(J), A(J), EC(J), E0(J), X1(J), Y1(J): NEXT
40 FORJ=0TO12: READMD(J), NM$(J): NEXT
45 FORJ=0TO3: READPL$(J): NEXT
50 CLS: PRINT"OPTION"; TAB(9); "PLANETS": PRINT
60 FORJ=0TO3: PRINTTAB(2); J; TAB(11); PL$(J): NEXT
70 PRINT: INPUTO: O=INT(O): INPUT"YEAR, MONTH"; Y, M: Y=INT(Y): M=INT(M): CLS
80 IFO=0THENN1=1: N2=3
90 IFO=1THENN1=1: N2=4
100 IFO=2THENN1=3: N2=6
110 IFO=3THENN1=5: N2=9
120 K1=34/A(N2): FORJ=N1TON2: R(J)=A(J)*K1: X2(J)=X1(J)*K1: Y2(J)=Y1(J)*K1
130 NEXT: XS=56.5+6*(O=3): YS=40.5+7*(O=3): PLOTXS, YS, 1
140 FORJ=N1TON2: S=15
150 IF(J=N1)*(O<2)=1THENS=30
160 IF(J=N1)*(O>1)=1THENS=45
170 IF(J=N1+1)*(O>1)=1THENS=30
175 X(J)=0: Y(J)=0
180 FORJ1=0TO360-SSTEPS: T=K*J1
190 PLOTXS+X2(J)+R(J)*COS(T), YS+Y2(J)+R(J)*SIN(T), 3: NEXTJ1, J
240 IP=0: DP=1: MP=M: YP=Y
245 OUTPUTDP, 0, 6, 3: OUTPUTNM$(M), 20, 6, 3: OUTPUTYP, 74, 6, 3
247 OUTPUTPL$(0), 10, 75, 3: OUTPUT"DAY", 90, 69, 3
250 GOSUB800: DY=1+MD(M-1)-LY*(M>2)
260 D=365*(Y-1964)+INT((Y-1961)/4)-INT((Y-1901)/100)+INT((Y-1601)/400)
265 D=D+DY-1
270 FORJ=N1TON2: E=M0(J)+D*MU(J): E=E+E0(J)+EC(J)*SIN(E)
280 PLOTX(J), Y(J), C(J)
290 X(J)=XS+X2(J)+R(J)*COS(E): Y(J)=YS+Y2(J)+R(J)*SIN(E)
300 C(J)=POINT(X(J), Y(J)): PLOTX(J), Y(J), 1: NEXT
450 IFJOY(0)=1GOTO50
460 IN=(1+INT(POT(0)/5))*(1-6*(O=2)-29*(O=3)): IFIN=IPGOTO480
470 OUTPUTIP, 84, 75, 0: OUTPUTIN, 84, 75, 3: IP=IN
480 IFJOY(0)<>8GOTO500
490 IN=-IN: GOTO520
500 IFJOY(0)<>4GOTO450
520 D=D+IN: DY=DY+IN
530 J=(DY<1)-(DY>365+LY): IFJ=0GOTO570
540 Y=Y+J: IFJ=1THENDY=DY-365-LY
550 GOSUB800: IFJ=-1THENDY=DY+365+LY
560 GOTO530
570 IFY=YPGOTO590
580 OUTPUTYP, 74, 6, 0: OUTPUTY, 74, 6, 3: YP=Y
590 FORJ=0TO12: IFDY>MD(J)-LY*(J>1)THENM=J+1
```

PLANETARY ORBITS, cont.

```
600 NEXT:IFM=MPGOTO620
610 OUTPUTNM$(MP),20,6,0:OUTPUTNM$(M),20,6,3:MP=M
620 DM=DY-MD(M-1)+LY*(M>2):OUTPUTDP,0,6,0:OUTPUTDM,0,6,3:DP=DM
690 GOTO270
800 LY=(Y-100*INT(Y/100)=0)-(Y-4*INT(Y/4)=0)
810 LY=LY-(Y-400*INT(Y/400)=0):RETURN
900 DATA.046,71423E-6,.387,.206,1.342,-.018,-.078
902 DATA3.896,279624E-7,.723,.007,2.288,.003,-.004
904 DATA6.23,172028E-7,1,.017,1.777,.004,-.817
906 DATA5.739,914611E-8,1.523,.093,5.854,-.129,.06
908 DATA.139,14504E-7,5.2,.048,.234,-.244,-.856
910 DATA4.201,580875E-9,9.57,.054,1.548,-.009,-.517
912 DATA6.086,205521E-9,19.1,.045,2.961,-.015,-.861
914 DATA3.26,104859E-9,30,.001,.683,-.023,-.819
916 DATA5.617,699162E-10,39.9,.246,3.913,6.955,6.716
930 DATA0," ",31,JAN,59,FEB,90,MAR,120,APR,151,MAY,181,JUN
935 DATA212,JUL,243,AUG,273,SEP,304,OCT,334,NOV,365,DEC
950 DATA1-3,1-4,3-6,5-9
```

ADDRESS CORRECTION

In Interaction no. 4, the zipcode of Harry Holloway on page 16 is incorrect. The correct address is :

Harry Holloway
P.O. Box 2263
Ann Arbor, MI 48106

INTERACT PROGRAMS shipped postpaid, 1 each, Video Chess, Star Track, $10. each; Regatta, Concentration, Calculator, $5. each... Would also like to hear from other users in the Golden Triangle area.
Scott Parker 5775 Kristin Beaumont, TX 77706

RF

INTERACTION

NEWSLETTER
DETROIT
INTERACT
GROUP

DATE DECEMBER, 1980 VOLUME NO. I, no.6

INTERACTION CONTINUES FOR 1981

Yes, INTERACTION will continue in 1981. And no, I am not burned out or running out of material to publish, as the rumors have been saying. In actuality, I hope to make Interaction better next year. I hope you have seen an improvement as this year has progressed. As part of the improvement effort, there is a questionaire in the back of this issue. Next year's contents will be affected by your answers. Upon return of the questionaire, hopefully with your 1981 renewal, I will send you a list of subscribers in your state if you wish. My apologies to those of you who are alone in your state. I'm willing to send an adjoining state list if it would be of any help to you. I will not distribute a complete mailing list to anyone as I'm sure you all get enough junk mail already and don't need someone else to have your name.

INTERACTION started out early in 1980 as a cooperative newsletter to be done by the Detroit Interact Group. Well, I've ended up doing most of the work so it's really my newsletter, though it would not have been even half as good without the reader contributions. My thanks to all contributers, some of whose work has yet to be published. I have been trying to keep the Detroit Interact Group independent of the newsletter. Meeting notices are mailed separately from the newsletter, since they are little interest except to local owners. However, if you are outside the Detroit metro area and still wish to receive meeting notices let me know as everyone is welcome at our meetings

For the benefit of all subscribers, the Detroit Interact Group is becoming a member of the Midwest Affiliation of Computer Clubs (MACC). MACC is a group of fifteen plus computer clubs that try to help one another. The CACHE club in Chicago is also a member. We will exchange newsletters with other clubs and information of interest to Interact owners will be reprinted in INTERACTION. One important function of MACC is their annual COMPUTERFEST. COMPUTERFEST '81 will be in May in Columbus, Ohio. The Detroit Interact Group and INTERACTION will be exhibiting at the convention demonstrating the Interact and promoting the newsletter. Maybe we can make COMPUTERFEST '81 a midwest Interact owners' convention.

The lack of group organization, negative cooperation from Interact vendors (people are still being told that I sell a memory expansion kit), and the unexpected amount of work were some of the reasons why I contemplated dumping the whole thing. But thanks to the encouraging letters I've received and my increasing involvement in the computer hobbyist world, I intend to face and overcome any adversities in the year to come. I hope you decide to join me for another year of INTERACTION!

1981 RENEWAL
INFORMATION
ON PAGE 19

WHERE IS YOUR VARIABLE, AND WHAT DOES IT LOOK LIKE

by Edward Berne 5626 Appleridge Trail West Bloomfield, MI 48033

Interact Basic stores program variables in two tables immediately following the user program in memory . The beginning of the first table is pointed to by an address stored at memory locations 19709 and 19710.

All addresses are stored in "swapped" form in the computer. This means that the "page" or high order portion of the address is stored after the low order portion. For example, the address would be stored as

1st Byte - 32_D (00100000_B)
2nd Byte - 78_D (01001110_B)

Putting these together in reverse order gives 0100111000100000_B, which is 20000. Since the PEEK function returns the contents of a memory location in decimal, the address can be interpreted by adding the first byte to the second byte times 2^8 (256).

$$32 + (78 \times 256) = 20000$$

The first variable table contains all non-array variables, as well as header records for user defined functions (DEF FN statements), divided into 6 byte records.

The first two bytes of each record identify the variable name in ASCII form. The Basic interpreter takes advantage of the fact that the ASCII code is only 7 bits long (i.e. 0 to 127_D). The high order bit in the first two bytes identifies the type of variable:

High Order Bit Of Byte 1	Byte 2	Variable Type
0	0	Numerical
1	0	String
0	1	DEF Statement Header

String variable records are laid out as follows:

Byte	
1	2nd digit of string ID
2	1st digit of string ID
3	length of string in bytes
4	always 0
5 } 6 }	address of first digit of string

String literals that have not been manipulated are located within the user program. All other strings are located in a separate string space controlled by the CLEAR statement.

WHERE IS YOUR VARIABLE, cont.

Numerical variables are more complicated. They are stored in a binary floating point representation similiar to scientific notation. First the variable is expressed as a 24 bit binary number with the bytes in reverse order. Then the number is shifted (rotated in assembly language) until the high order bit of the first byte is a 1. An exponent byte is the number of places of shift + 128. As an example, I will use X1 = 44:

1. Express in binary - $44_D = 00101100_B$
2. Shift left two places = $101100000_B = 176_D$
3. Exponent = 2 + 128

The full record:

Byte	Value	Meaning
1	31	1 in ASCII
2	58	X in ASCII
3	0	
4	0	
5	176	see above
6	130	exponent

Since the high order bit in byte 5 is always 1, it conveys no information. Therefore Basic uses this bit to designate negative numbers by changing that bit to 0.

All of this may seem unnecessarily complex, however it is straightforward and fairly easy in 8080 machine language. This is the language that Interact Basic is written in.

The array variables are contained in a second table immediately following the regular variables. Its beginning is pointed to by the address in locations 19711 and 19712. Each array starts with a header record in the following form:

Byte	Contents
1	ID of array
2	
3	distance to next array in bytes
4	
5	number of dimensions
6	number of elements in 1st dimension
7	
8	number of elements in 2nd dimension
9	
etc.	

The header is followed by the data elements. Each is 4 bytes long, no ID being required, and in the same format as regular variables.

The following routine, which I wrote as an exercise, will list out all the variables in a program to which it is appended.

WHERE IS YOUR VARIABLE, cont.

Since Basic does not permit the use of local variables, I used non-descriptive variable names in the hope that they won't conflict with variable names in the main program.

If you run the routine note that at least one variable must appear before line 1000. Otherwise the routine will try to list its own variables and garbage will result.

By changing the END in line 1120 to RETURN you could make this into a subroutine which might be helpful in debugging large programs.

```
LIST
1000 POKE19215,25
1010 DEFFNA(Z4)=PEEK(Z4)+PEEK(Z4
+1)*256
1020 Z0=FNA(19709)
1030 Z3=1:IFPEEK(Z0)>127THENZ3=2
1040 IFPEEK(Z0+1)>128THENZ0=Z0+6
:GOTO1030
1050 IFZ0=FNA(19711)-36GOTO1110
1060 ONZ3GOSUB1250,1350
1070 PRINTZ5$"=";
1080 ONZ3GOSUB1260,1360
1090 Z0=Z0+6
1100 GOTO1030
1110 Z0=Z0+36
1120 IFZ0=FNA(19713)THENEND
1130 Z3=1:IFPEEK(Z0)>127THENZ3=2
1140 ONZ3GOSUB1250,1350
1150 IFPEEK(Z0+4)=1GOTO1190
1160 PRINTZ5$" IS A"PEEK(Z0+4)"D
IMENSIONAL ARRAY"
1170 Z0=FNA(Z0+2)+Z0+4
1180 GOTO1120
1190 Z0=Z0+5
1200 FORZ0=0TOFNA(Z0)-1
1210 PRINTZ5$"("STR$(Z4)")=";
1220 ONZ3GOSUB1260,1360
1230 Z0=Z0+4
1240 NEXT:Z0=Z0+2:GOTO1120
1250 Z5$=CHR$(PEEK(Z0+1))+CHR$(P
EEK(Z0)):RETURN
1260 Z2=PEEK(Z0+5)-128
1270 IFZ2=-128THENPRINT" 0":RETU
RN
1280 Z1=PEEK(Z0+4)
1290 IFZ1<128THENZ1=Z1+128:GOTO1
310
1300 PRINT"-";
1310 Z1=Z1/2^(8-Z2)
1320 Z1=Z1+PEEK(Z0+3)/2^(16-Z2)
1330 Z1=Z1+PEEK(Z0+2)/(24-Z2)
1340 PRINTZ1:RETURN
1350 Z5$=CHR$(PEEK(Z0+1))+CHR$(P
EEK(Z0)-128)+"$":RETURN
1360 Z2=PEEK(Z0+2)-1
1370 IFZ2=-1THENPRINT:RETURN
1380 FORZ1=FNA(Z0+4)TOFNA(Z0+4)+
Z2
1390 PRINTCHR$(PEEK(Z1));
1400 NEXT:PRINT:RETURN
Ok
```

CHECKERS

by Bob Draganski 14301 Harrison Livonia, MI 48154

Bob wrote this program about two years ago. It is based on the Creative Computing version of the game. When the game prompts you with FROM type first the column (x coord.) then the row (y coord.) position of the peice you want to move. When it prompts TO type where you want peice moved to. After you make a jump, you will get the prompt + asking you if you want to make a double or triple jump. If you cannot make a jump type in two 8's or two 9's to indicate so to the computer. If you are like me after a few games you should be able to see the computer's strategy and beat it almost every time. It is still a good, challenging game, though.

```
LIST
10 COLOR4,3,0,1:CLS:WINDOW11
80 DIMR(4),S(7,7):G=-1
85 R(0)=-99
90 DATA1,-3,1,-3,0,-3,-1,-3,-3,1
,-3,0,-3,-1,-3,-1,15
91 FORX=29TO77:FORY=25TO73STEP6:
PLOTX,Y,2:NEXT:NEXT
93 FORY=25TO73:FORX=29TO77STEP6:
PLOTX,Y,2:NEXT:NEXT
100 OUTPUT"01234567",30,24,1
105 FORX=0TO7:OUTPUTX,18,X*6+30,
1:NEXT
120 FORX=0TO7:FORY=0TO7:READJ:IF
J=15THEN180
160 S(X,Y)=J:GOTO200
180 RESTORE:READS(X,Y)
200 NEXT:NEXT:GOSUB1420:Z=0
220 GOSUB1895
230 FORX=0TO7:FORY=0TO7:IFS(X,Y)
=-1THENFORA=-1TO1STEP2:B=G:GOSUB
650:NEXT
330 IFS(X,Y)=-2THENFORA=-1TO1STE
P2:FORB=-1TO1STEP2:GOSUB650:NEXT
:NEXT
350 NEXT:NEXT:GOTO1140
650 U=X+A:V=Y+B:IFU<00RU>70RV<00
RV>7THEN870
740 IFS(U,V)=0THENGOSUB910:GOTO8
70
770 IFS(U,V)<0THEN870
790 U=U+A:V=V+B:IFU<00RV<00RU>70
RV>7THEN870
850 IFS(U,V)=0THENGOSUB910
870 RETURN
910 IFV=0ANDS(X,Y)=-1THENQ=Q+2
920 IFABS(Y-V)=2THENQ=Q+6.5
960 IFY=7THENQ=Q-1.5
980 IFY=00RU=7THENQ=Q+1
1030 FORC=-1TO1STEP2:IFU+C<00RU+
1035 IFS(U+C,V+G)<0THENQ=Q+1:GOT
O1080
1040 IFU-C<00RU-C>70RV-G>7THEN10
80
1045 IFS(U+C,V+G)>0AND(S(U-C,V-G
)=00R(U-C=XANDV-G=Y))THENQ=Q-2
1050 IFB<>1THEN1076
1052 IFS(U+C,V-G)>0ANDS(U-C,V+B)
=0THENQ=Q-2:GOTO1080
1055 IFS(U+C,V+B)=2AND(S(U-C,V-B
)=00R(U-C=XANDV-B=Y))THENQ=Q-2:G
OTO1080
1060 IFS(U-C,V-B)>0ANDS(U+C,V+B)
=0THENQ=Q-2:GOTO1080
1065 IFS(U+C,V+B)=1THENQ=Q+1.5:G
OTO1080
1070 IFY>2THENQ=Q-1:GOTO1080
1076 IFS(U+C,V-B)>1ANDS(U-C,V-B)
=0THENQ=Q-2
1077 IFS(U+C,V-B)=1THENQ=Q+1.5
1080 NEXT:IFQ=R(0)ANDRND(1)>.7TH
ENQ=Q+.1
1085 IFQ>R(0)THENR(0)=Q:R(1)=X:R
(2)=Y:R(3)=U:R(4)=V
1100 Q=0:RETURN
1140 IFR(0)=-99THEN1880
1230 R(0)=-99
1240 X=R(3):Y=R(4):IFR(4)=0THENS
(R(3),R(4))=-2:GOSUB1430:GOTO131
0
1250 S(R(3),R(4))=S(R(1),R(2))
1255 GOSUB1430
1310 S(R(1),R(2))=0:X=R(1):Y=R(2
):GOSUB1430:IFABS(R(1)-R(3))<>2T
HEN1590
1330 S((R(1)+R(3))/2,(R(2)+R(4))
/2)=0:X=(R(1)+R(3))/2:Y=(R(2)+R(
4))/2
1335 GOSUB1430
```

CHECKERS, cont.

```
1340 X=R(3):Y=R(4):IFS(X,Y)=-1TH
ENB=-2:FORA=-2TO2STEP4:GOSUB1370
:NEXT
1350 IFS(X,Y)=-2THENFORA=-2TO2ST
EP4:FORB=-2TO2STEP4:GOSUB1370:NE
XT:NEXT
1360 IFR(0)<>-99THENR(0)=-99:GOT
O1240
1365 GOTO1580
1370 U=X+A:V=Y+B:IFU<0ORU>7ORV<0
ORV>7THEN1400
1380 IFS(U,V)=0ANDS(X+A/2,Y+B/2)
>0THENGOSUB910
1400 RETURN
1420 FORY=7TO0STEP-1:FORX=0TO7
1424 I=X*6+30:I1=Y*6+30
1425 IFS(X,Y)=-3THENOUTPUT"O",I,
I1,3:OUTPUT"*",I,I1,3:GOTO1550
1430 I=X*6+30:I1=Y*6+30:IFP=0THE
NOUTPUT"O",I,I1,2:OUTPUT"*",I,I1
,2
1432 IFS(X,Y)=0THENOUTPUT"O",I,I
1,2:OUTPUT"*",I,I1,2
1435 IFS(X,Y)=1THENOUTPUT"0",I,I
1,3
1440 IFS(X,Y)=-1THENOUTPUT"X",I,
I1,1
1450 IFS(X,Y)=-2THENOUTPUT"X",I,
I1,1:OUTPUT"+",I,I1,1
1460 IFS(X,Y)=2THENOUTPUT"0",I,I
1,3:OUTPUT"+",I,I1,3
1545 IFP=1THENPRINTCHR$(7):RETUR
N
1550 NEXT:NEXT:P=1:RETURN
1580 GOSUB1895
1590 OUTPUT"FROM",6,11,1:A$=INST
R$(1):OUTPUTA$+",",36,11,2:E=VAL
(A$)
1591 A$=INSTR$(1):OUTPUTA$,46,11
,2:H=VAL(A$)
1592 IFE<0ORE>7ORH<0ORH>7THENPRI
NT:TONE190,200:GOTO1590
1650 IFS(E,H)<=0THENPRINT:TONE19
0,200:GOTO1590
1670 OUTPUT"TO",56,11,1:A$=INSTR
$(1):OUTPUTA$+",",74,11,2:A=VAL(
A$):X=A
1674 A$=INSTR$(1):OUTPUTA$,86,11
,2:B=VAL(A$):Y=B
1676 IFA<0ORA>7ORB<0ORB>7THENPRI
NT:TONE190,200:GOTO1670
1677 IFABS(E-A)=2ANDS((E+A)/2,(H
+B)/2)>-1THEN1690
1678 PRINT
1679 IFS(E,H)=1ANDH>BTHEN1690
1680 IFS(A,B)=0ANDABS(A-E)<=2AND
ABS(A-E)=ABS(B-H)THEN1700
1690 PRINT:TONE190,200:GOTO1590
1700 I=46
1750 S(A,B)=S(E,H):X=A:Y=B:GOSUB
1430:S(E,H)=0:X=E:Y=H:GOSUB1430
1751 IFABS(E-A)<>2THEN1810
1755 IFB=7THENX=A:Y=B:S(A,B)=2:G
OSUB1430
1800 S((E+A)/2,(H+B)/2)=0:X=(E+A
)/2:Y=(H+B)/2:GOSUB1430
1802 OUTPUT"+TO",6,11,1:A$=INSTR
$(1):OUTPUTA$+",",30,11,2:A1=VAL
(A$)
1803 A$=INSTR$(1):OUTPUTA$,42,11
,2:B1=VAL(A$):IFA1>7THEN1810
1804 IFS(A1,B1)<>0ORABS(A1-A)<>2
ORABS(B1-B)<>2THENPRINT:TONE190,
200:GOTO1802
1805 PRINT
1806 E=A:H=B:A=A1:B=B1
1807 IFABS(E-A)=2ANDS((E+A)/2,(H
+B)/2)>-1THEN1802
1809 GOTO1750
1810 PRINT:IFB=7THENS(A,B)=2:X=A
:Y=B:GOSUB1430
1830 GOTO220
1880 OUTPUT"YOU WIN",36,17,2
1890 OUTPUT"AGAIN(Y/N)",26,11,1:
A$=INSTR$(1):IFA$="N"THENWINDOW7
7:END
1891 CLEAR:GOTO10
1895 FORX=0TO7:FORY=0TO7:IFS(X,Y
)>0THENRETURN
1896 NEXT:NEXT:OUTPUT"I WIN",36,
17,2:GOTO1890
Ok
```

LETTERS TO THE EDITOR

LEVEL 2 BASIC INFORMATION FROM: MARK SLAGH

While it is true that Level 2 Basic has no printer commands, neither can you poke or peek, as it is supplied from Microsoft. In a similar manner to enabling pokes with the'Poke 19215,25' command, you can upgrade Level 2 Basic and even Fast graphics Basic to provide U80M Port operations.

The user can write either a Basic language sub-routine to handle data transfer to the port address, or a Machine language User sub-routine.

If desired, the user of a U80M Port can purchase an inexpensive 'overlay' program on cassette, that updates Level 2 Basic, and provides for port operation.

I encourage Interact owners to compare the breadth of information and technical expertise available before choosing a supplier.

Sincerely,

Mark Slagh
Slagh System Services

ADVENTURE!!! Byte-Creative Computing-Recreational Computing and Softside have devoted entire issues to it.
ADVENTURE!!! Programmers have made whole careers of it.
ADVENTURE!!! The most popular game ever written for personal computers.
ADVENTURE!!! Now available for the Interact

If you've played an Adventure you already know. If you havn't you're in for a treat. In Adventure you use your mind instead of your thumb.Exploring new worlds. Facing dangers and solving mind bending puzzles. This is not a game you play for an hour and then quit bored. It may take you days or weeks to get a good score.
Over four months in the writing. A vocabulary of over 100 words. Over 30 rooms. The fast machine language program takes almost all of the available ram. It even has a save game feature so you can continue a game in progress.
This is not a rehash of another program or a copy of anything else, but a new original Adventure designed specifically for the Interact. There is nothing else like it.
Can you get the pricless ruby from the King Cobra?What does that strange inscription say? Why do evil eyes watch your every move?Can you solve these and the many other mysteries of THE TROLL HOLE ADVENTURE??? Will you come out rich? Will you come out at all?
Written by Long Playing Software. Available only from Micro-Video

THE TROLL HOLE ADVENTURE---$14.95-AT LAST INTERACT ADVENTURE

SOME EDU-BASIC BITS

by Steve Smith 22364 Oxford Dearborn, MI 48124

For my first year with the Interact, I communicated with it entirely in the Edu-Basic language. Since my 16K upgrade and my Level II tape purchase in January of this year (when they were "closed" for inventory), Edu-basic gathers dust in my drawer and gets very little yse. To write this article, I blew away the cobwebs so that I might help those who still use the language and introduce newcomers to Interact's Integer Basic.

(Ed. note- Edu-Basic is based on Li-Chen Wang's Palo Alto Tiny Basic first published in Doctor Dobb's Journal, May, 1976. For those who don't know, Dr. Dobb's Journal was started primarily as a forum for small languages for hobbyist computer builders. Edu-Basic in an 8K Interact left 1.3K of program space and 8K of program space in a 16K Interact.)

Edu-Basic is not a flashy language so the advantages it does have should be exploited to the fullest. The ability to abbreviate commands is one of the best advantages. I have included a list of shortest abbreviations for each word in this statement and function list.

STATEMENT	ABBREV.	FUNCTION
ABS	A.	Absolute value
CLEAR	C.	Clear screen
COLOR	CO.	Color command
FIRE	F.	Fire button
FOR	F.	For ... next
GOSUB	GOS.	Go sub
GOTO	G.	Go to
IF	IF	If ... then
INCHR	I.	Input - keyboard press delivers ASCII
INPUT	IN.	Input variable
JOY	J.	Joystick
LET		Let (optional)
LIST	L. or .	List
LOAD	LO.	Load program tape
NEW	N.	New program
NEXT	NE.	Next
OUTCHR	O.	Prints character from ASCII value
PEEK	PE.	Peeks screen location for color no.
PLOT	PL.	Plots pixel
POT	P.	Potentiometer knob
PRINT	P.	Print
REM		Remark
RETURN	R.	Return from go sub
REW		Rewind tape
RND	R.	Random RND(X) gives number between 1 and X
RTAPE	RT.	Read data array !(n) is the only array
RUN	R.	Run
SAVE	S.	Save program on tape
SIZE	S.	Print SIZE gives memory free
SOUND	SO.	Sound
STEP	S.	Step for ... next ... step
STOP	S.	Stop
TONE	TO.	Tone
WTAPE	WT.	Write array to tape

EDU-BASIC BITS, cont.

One trick I learned, quite by accident, is the ability to print a backspace using the statement OUTCHR(8). Its use can create some fancy effects for Edu-Basic, and thus may find its way into one of your programs. Here is a simple demo --

```
  1   REM INTERACT EDU-BASIC
  5   REM BACKSPACE DEMONSTRATION
 10   CLEAR;PRINT"INTERACT"
 20   PRINT"EDU-BASIC",
 25   N=9;GOSUB100
 30   PRINT"BACKSPACE",
 35   N=10;GOSUB100
 40   PRINT"DEMONSTRATION",
 45   N=13;GOSUB100
 50   COLOR(RND(7)-1,7,7,7)
 60   GOTO20
100   FORI=1TO500;NEXTI
110   FORI=1TON
120   OUTCHR(8)
130   NEXTI;RETURN
```

If you pause this while a word is displayed, you will note that the word INTERACT is pushed one pixel further to the right then the line below it. This is a bug which occurs only after a CLEAR statement. Using the line 10 CLEAR;PRINT;PRINT"INTERACT" lines everthing up nicely.

Another trick I learned is the ability of Edu-Basic to set variables outside the program. This can be very useful in long programs where you're fighting for every byte of memory to make it fit. In a diluted version of Star Trek I wrote in Edu-Basic I used this system. The Star Trek was written in two separate programs, the first one set the number of Klingons, starbases, and their corresponding quadrants, along with other initializing variables. The second program used the data without using up memory to set them. One just loads and runs the first program then does likewise with the second.

PRODUCT REVIEW

QUEST in Edu-Basic by Dave Schwab

Dave has successfully translated an adventure style program written for an 8K PET into Edu-Basic. While this game does not have the complexity of The Troll Hole Adventure, it is the first excellent use of Edu-Basic I've seen. In the game you must retreive a treasure from an underground maze. The pirate lurks in the maze and may steal his treasure back from you once you have found it.

The game was originally published in the July, 1979 Byte magazine. Dave had to eliminate all the string variables from the original program and had to transform the two dimensional array into Edu-Basic's one dimensional (!) array. Dave has received permission from the author, Roger Chaffee, to make QUEST available to Interact owners.

Send $5.00 to cover tape, duplication and mailing costs to:
DAVID J. SCHWAB 10 Jay Lee Court Ann Arbor, MI 48104

reviewed by Stephen Cook

★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★

★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★

COUNT THE LETTERS

by Stephen Cook

This preschool educational game uses the new FASTLINE graphics overlay by Harry Holloway. The remarks should help outline the subroutines. The Draw Box routine can easily be converted to Fast Graphics Basic. It first outlines a box (lines 1010, 1020, 1030, 1040) and then fills it in (line 1050). The fireworks routine is a reinforcement reward to encourage playing the game. It draws a line from a random point (XB) of a random length (YL) with a random offset angle (O). Converting this to Graphics Basic would require a little more work. The offset would have to be eliminated as Fast Graphics Basic can only draw a vertical or horizontal line.

My four year old daughter likes this game. My hope is it will teach her the keyboard layout and improve her counting.

```
LIST
1 REM COUNT THE LETTERS
2 REM BY STEPHEN COOK - 11/80
3 REM INTERACT LEVEL II BASIC
4 REM *FASTLINE REQUIRED* !!
10 CLS
20 N=.13
30 Q=0
100 GOSUB1000
110 GOSUB2000
120 GOSUB3000
125 FORB=1TO3:PRINTCHR$(7);:NEXT
130 GOSUB4000
140 GOSUB5000
200 GOTO10
```

```
500 REM PAUSE LOOP
510 FORPT=0TOP
520 NEXTPT
530 RETURN
900 END
1000 REM DRAW BOX
1010 LINE10,10,1,10,60
1020 LINE60,10,1
1030 LINE60,60,1
1040 LINE10,60,1
1050 BOX11,59,3,49,49
1090 RETURN
```

COUNT THE LETTERS, cont.

```
2000 REM FILL BOX
2010 L$=CHR$(INT(26*RND(1))+65)
2100 FORX=14TO49STEP7
2110 FORY=20TO56STEP7
2150 M=RND(1)
2170 IFM<NTHENGOSUB2500
2190 NEXTY:NEXTX
2300 RETURN
2500 REM NO. OF LETTERS
2520 OUTPUTL$,X,Y,2
2540 Q=Q+1
2560 RETURN
3000 REM FLASH
3010 OUTPUT"LETTER ?",11,70,1
3020 COLOR4,3,0,4
3040 P=150:GOSUB500
3060 COLOR4,3,4,4
3080 P=150:GOSUB500
3100 COLOR4,3,0,4
3120 P=150:GOSUB500
3140 COLOR4,3,0,7
3160 P=300:GOSUB500
3180 IFPEEK(24529)=ASC(L$)+32THENRETURN
3200 GOTO3020
4000 REM NUMBER
4020 OUTPUT"HOW",77,50,1
4040 OUTPUT"MANY",74,43,1
4060 OUTPUT"?",83,36,1
4080 Q$=INSTR$(1)
4100 IFQ=ASC(Q$)-48THENRETURN
4120 SOUND3,32
4140 P=50:GOSUB500
4160 SOUND7,4096
4180 GOTO4080
```

```
5000 REM FIREWORKS
5010 CLS
5020 FORE=1TOQ
5030 O=20*RND(1)-10
5040 XB=80*RND(1)+10
5060 YL=60*RND(1)+10
5080 SOUND5,10
5100 LINEXB,10,3*RND(1)+1,XB+O,YL
5120 SOUND1,0
5140 OUTPUT"*",XB+O-2,YL+3,3*RND(1)+1
5160 P=100:GOSUB500
5200 NEXTE
5220 P=150:GOSUB500
5240 SOUND7,4096
5260 RETURN
Ok
```

★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★

CONCENTRATION

by Kevin VanBrook 8701 Town Park # 3166 Houston, TX 77036

Kevin writes that this is his wife's favorite game. It uses a 7 by 8 character board. You can play against the computer or another player. Make your move using first the X coordinate then the Y coordinate. (Warning: The computer is very good beyond level 2)

```
LIST
5 PRINTCHR$(8)
10 CLS:CLEAR
20 DEFFNX(X)=INT(X/10)
30 DEFFNY(X)=X-FNX(X)*10
90 COLOR7,1,6,0
100 CLS:DIMAR(8,5),M(50),SC(2)
110 PRINT"   WELCOME TO      CON
CENTRATION":PRINT:PRINT:PRINT:PR
INT
120 FORI=0TO1000:NEXTI
125 N=54
130 PRINT"DO YOU WISH TO   PLAY
AGAINST THE COMPUTER(0) OR   AGA
INST ANOTHER"
140 PRINT"PLAYER(1)?"
150 F1=VAL(INSTR$(1)):PRINTF1:IF
F1=0ORF1=1THEN170
160 PRINT"WRONG RESPONSE.":GOTO1
30
170 IFF1=1THEN600
180 PRINT"TYPE IN YOUR      NAME.
"
190 INPUTL$
200 PRINT"INPUT YOUR SKILL LEVEL
(0 TO 5).":L=VAL(INSTR$(1))*10:P
RINTL/10
205 IFL=0THENL=1
210 IFL/10<0ORL/10>5THENPRINT"PA
Y ATTENTION.":GOTO200
220 PRINT"DO YOU WISH TO GOFIRST
?":J$=INSTR$(1)
230 GOSUB2000
240 GOSUB5000
250 GOSUB7000
270 IFJ$="Y"THEN670
300 GOSUB10000
305 FORI=0TOL:FORJ=I+1TOL
310 IFM(I)=0THEN380
320 IFM(J)=0THEN370
330 IFAR(FNX(M(I)),FNY(M(I)))<>A
R(FNX(M(J)),FNY(M(J)))THEN370
340 B1=M(I):B2=M(J):IFAR(FNX(B1)
,FNY(B1))=0THEN370
345 IFB1=B2THEN370
350 IFAR(FNX(B2),FNY(B2))=0THEN3
70
360 GOTO490
370 NEXTJ
380 NEXTI
390 B1=INT(RND(1)*8.5)*10+INT(RN
D(1)*5.5)
400 IFAR(FNX(B1),FNY(B1))=0THEN3
90
410 FORI=0TOL
420 IFM(I)=0THEN460
430 IFAR(FNX(M(I)),FNY(M(I)))<>A
R(FNX(B1),FNY(B1))THEN460
440 IFM(I)=B1THEN460
450 B2=M(I):GOTO490
460 NEXTI
470 B2=INT(RND(1)*8.5)*10+INT(RN
D(1)*5.5)
480 IFAR(FNX(B2),FNY(B2))=0THEN4
70
485 IFB2=B1THEN470
490 GOSUB2500:GOSUB2750
500 IFAR(FNX(B1),FNY(B1))=AR(FNX
(B2),FNY(B2))THEN540
520 GOSUB4000:FORI=0TO800:NEXTI:
GOSUB3000
530 GOTO690
540 OUTPUTSC(0),88,19,0:SC(0)=SC
(0)+2:OUTPUTSC(0),88,19,3
550 N=N-2:IFN<=0GOTO900
555 FORI=0TO600:NEXTI
560 GOSUB3500:GOTO300
600 PRINT"LEFT PLAYER, TYPEIN YO
UR NAME.":INPUTL$
610 PRINT"RIGHT PLAYER,    TYPE
IN YOUR      NAME.":INPUTR$
620 PRINT"WHO WILL GO FIRSTLEFT(
1),OR RIGHT (2)?"
630 A=VAL(INSTR$(1))
640 IFA<>1ANDA<>2THENPRINT"LEFT(
1),OR RIGHT (2)?":GOTO630
650 GOSUB2000
655 GOSUB5000:GOSUB6000
```

CONCENTRATION, cont.

```
660 IFA=2THEN680
670 GOTO690
680 GOSUB10000:SI=2:GOTO700
690 SI=1:GOSUB11000
700 OUTPUT"FIRST BOX?",4,12,3:B1
=VAL(INSTR$(1))*10+VAL(INSTR$(1)
)
701 GOSUB8000
710 IFFNX(B1)<9ANDFNY(B1)<6THEN7
20
711 GOSUB9000
712 GOTO700
720 IFAR(FNX(B1),FNY(B1))>0THEN7
30
721 GOTO711
730 GOSUB2500
740 OUTPUT"SECOND BOX?",4,12,3:B
2=VAL(INSTR$(1))*10+VAL(INSTR$(1
))
741 GOSUB8000
750 IFFNX(B2)<9ANDFNY(B2)<6THEN7
60
751 GOSUB9000:GOTO740
760 IFAR(FNX(B2),FNY(B2))>0THEN7
70
761 GOTO751
770 IFB1=B2THEN751
771 GOSUB2750
780 IFF1=0THENGOSUB4000
790 IFAR(FNX(B1),FNY(B1))=AR(FNX
(B2),FNY(B2))THEN240
805 FORI=0TO500:NEXTI
810 GOSUB3000
820 SI=(3-SI)*F1
825 ONSI+1GOSUB10000,11000,10000
830 ONSI+1GOTO300,700,700
840 ONSIGOTO850,880
850 OUTPUTSC(1),0,19,0:SC(1)=SC(
1)+2:OUTPUTSC(1),0,19,3
860 N=N-2:IFN<0THEN900
870 GOSUB3500:GOTO700
880 OUTPUTSC(2),88,19,0:SC(2)=SC
(2)+2:OUTPUTSC(2),88,19,3
890 GOTO860
900 CLS
910 B=0:FORI=1TO2:IFSC(I)>SC(B)T
HENB=I
920 NEXTI
930 ONF1+1GOTO940,1010
940 PRINT"I SCORED";SC(0):PRINT"
POINTS."
950 PRINT"YOU SCORED";SC(1):PRIN
T"POINTS."
960 ONB+1GOTO970,980
970 PRINT"I WIN! I WIN! TOOBAD."
:GOTO990
980 PRINT"LUCKY. YOU WIN."
990 PRINT"CARE TO TRY AGAIN?":IF
INSTR$(1)="Y"THEN125
1000 END
1010 PRINTL$;" SCORED":PRINTSC(1
);"POINTS."
1020 PRINTR$;" SCORED":PRINTSC(2
);"POINTS."
1030 IFB=1THENPRINTL$;" WINS.":G
OTO990
1040 PRINTR$;" WINS.":GOTO990
2000 PRINT"PLEASE WAIT. I'M THIN
KING."
2005 FORL=0TO8:FORM=0TO5:AR(L,M)
=0:NEXTM:NEXTL
2010 FORI=64TO90:FORJ=0TO1
2020 X=INT(RND(1)*8.5):Y=INT(RND
(1)*5.5)
2030 IFAR(X,Y)<>0THEN2020
2040 AR(X,Y)=I
2050 NEXTJ:NEXTI
2060 CLS
2070 FORX=20TO92:FORY=27TO75STEP
8:PLOTX,Y,1:NEXTY:NEXTX
2080 FORY=28TO74:FORX=20TO92STEP
8:PLOTX,Y,1:NEXTX:NEXTY
2090 FORX=22TO88STEP8:FORY=33TO7
3STEP8
2100 OUTPUTCHR$(1),X,Y,2:NEXTY:N
EXTX
2110 FORX=16TO80STEP8
2120 I=(X-16)/8:OUTPUTI,X,25,3:N
EXTX
2130 FORY=33TO73STEP8:I=(Y-33)/8
2140 OUTPUTI,8,Y,3:NEXTY
2150 OUTPUT"SCORES",38,19,1
2160 OUTPUTSC(0),88,19,3:OUTPUTS
C(0),0,19,3
2170 RETURN
2500 X=FNX(B1):Y=FNY(B1):GOTO276
0
2750 X=FNX(B2):Y=FNY(B2)
2760 SOUND0,100
2770 OUTPUTCHR$(1),X*8+22,Y*8+33
,0
2780 OUTPUTCHR$(AR(X,Y)),X*8+22,
Y*8+33,3
2790 SOUND7,4096:RETURN
```

CONCENTRATION, cont.

```
3000 C=2
3010 FORI=0TO1:IFI=0THEN3030
3020 X=FNX(B2):Y=FNY(B2):GOTO304
0
3030 X=FNX(B1):Y=FNY(B1)
3040 SOUND0,100
3045 IFC=0THENAR(X,Y)=0
3050 OUTPUTCHR$(1),X*8+22,Y*8+33
,C
3060 SOUND7,4096
3071 NEXTI
3080 RETURN
3500 C=0:GOTO3010
4000 FORI=0TOL-2
4010 M(I)=M(I+1)
4020 NEXTI
4030 M(L-1)=B1:M(L)=B2
4040 RETURN
5000 A$=L$:X=5
5010 LT=LEN(A$):IFLT>8THENLT=8
5020 FORI=1TOLT
5030 OUTPUTMID$(A$,I,1),X,74-I*6
,2
5040 NEXTI
5050 RETURN
6000 A$=R$:X=95:GOTO5010
7000 A$="COMPUTER":X=95:GOTO5010
8000 FORI=4TO110STEP6:OUTPUTCHR$
(1),I,12,0
8010 NEXTI
8020 RETURN
9000 OUTPUT"WRONG RESPONSE",4,12
,2
9010 FORI=0TO500:NEXTI
9020 GOSUB3000
9030 RETURN
10000 SOUND0,10
10010 OUTPUTCHR$(1),5,74,0:SOUND
7,4096
10015 SOUND0,100
10020 OUTPUTCHR$(1),95,74,3
10030 SOUND7,4096:RETURN
11000 SOUND0,100
11010 OUTPUTCHR$(1),95,74,0:SOUN
D7,4096
11020 SOUND0,100
11030 OUTPUTCHR$(1),5,74,3:SOUND
7,4096
11040 RETURN
Ok
```

MOVING BANNER

by Bob Draganski 14301 Harrison Livonia, MI 48154

This short program shows how to produce a moving banner or billboard effect. Used in an otherwise lifeless program it can be quite an attention getter. Strings and string functions will have to be adjusted for different messages.

```
LIST
1 X=0
2 CLS:CLEAR100
10 DATARED ALERT," DANGER"," ALI
ENS ATTACKING"," TAKE COVER"
15 DATA"                    "," "
20 A$="                    "
25 READB$
26 L=LEN(B$)
27 IFL=1THENSTOP
35 X=X+1
40 A$=RIGHT$(A$,16)+MID$(B$,X,1)
50 OUTPUTA$,6,50,1:OUTPUTA$,6,50
,0
55 IFX=LTHENX=0:GOTO25
60 GOTO35
Ok
```

CHRISTMAS MUSIC

by Marv Long 1661 College Ferndale, MI 48220

Here's two Christmas songs for your Interact. The first set of DATA statements are O Come All Ye Faithful. The second set are Good King Wencelas. If you add a RESTORE and a GOTO statement after the program you can play the songs over and over.

```
LIST
10 REM*CHRISTMAS MUSIC*
20 REM*TRANSCRIBED BY*
25 REM  **MARV LONG**
30 REM*ADD YOUR OWN GRAPHICS AND
 MESSAGE*
40 FORM=1TO62
50 READX,Y
60 TONEX,Y
70 NEXTM
80 DATA124,98,124,196,168,72,124
,98,110,220,168,144
90 DATA97,125,110,110,97,125,91,
133,97,250,110,110,124,98
100 DATA124,196,131,92,148,82,13
1,92,124,98,110,110,97,125
110 DATA131,184,148,164,168,36,1
68,288,80,302,91,133
115 DATA97,125,91,266,97,250,110
,110,97,125,124,98,110,110
120 DATA131,174,148,30,168,65,12
4,98,124,98,131,92
130 DATA124,98,110,110,124,196,1
68,72,97,125,97,125,110,110
140 DATA97,125,91,133,97,250,110
,110,97,125,91,133,97,125
150 DATA110,110,124,98,131,184,1
24,98,91,133,97,250,110,220
160 DATA124,49,124,392
165 FORT=1TO500:NEXT
170 FORM=1TO54
180 READX,Y
190 TONEX,Y
200 NEXTM
210 DATA124,98,124,98,124,98,110
,110,124,98,124,98
220 DATA168,144,148,82,168,72,14
8,82,131,92
230 DATA124,196,124,196,124,98,1
24,98,124,98,110,110
240 DATA124,98,124,98,168,144,14
8,82,168,72,148,82
250 DATA131,92,124,196,124,196,8
0,151,91,133,97,125,110,110
260 DATA97,125,110,110,124,196,1
48,82,168,72,148,82,131,92
270 DATA124,196,124,196,168,72,1
68,72,148,82,131,92,124,98,124,9
8
280 DATA110,220,80,151,91,133,97
,125,110,110,124,196,91,266
290 DATA124,49,124,394
Ok
```

★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★

PETALS AROUND THE ROSE

by Jerry Ravary 1149 Sunset Blvd. Monroe, MI 48161

This is a single player pattern guessing game. Jerry says it was based on a TRS-80 program. It has a slow (what do you expect in Basic!) but nice dice graphics routine. As for the puzzle itself I've spent several hours unsuccessfully trying to solve it without studying the listing while my wife solved it in less than 10 tries. Frustrating!

```
LIST
0 CLS
1 PRINT"TO START THE GAME":PRINT
:PRINTTAB(5);"PRESS 'S'"
2 PRINT:PRINT"TO START THE TEST"
:PRINT:PRINTTAB(5);"PRESS 'T'"
3 A$=INSTR$(1)
4 IFA$="T"THEN2000:GOTO10
10 CLS
12 PRINT">>PETALS AROUND<<"
13 PRINT">>>>THE  ROSE<<<<":PRIN
T:PRINT:PRINT
14 PRINT"THE OBJECT OF THEGAME I
S TO GUESS THE ANSWER TO THETHRO
W";
15 PRINT" OF FIVE  DICE. YOU G
ET TWO CLUES.":PRINT:PRINT"PRESS
 ANY 'KEY'"
16 A$=INSTR$(1):CLS:PRINT"1) THE
 NAME OF  THE GAME IS ":PRINT"P
ETALS AROUND THE"
17 PRINT"      ROSE":PRINT"2) AL
L ANSWERS ":PRINT"ARE EVEN NUMBE
RS."
18 PRINT"SO THAT I KNOW  THAT Y
OU ARE NOT JUST A LUCKY    GUES
SER YOU MUST"
19 A$=INSTR$(1):CLS
20 PRINT"GET SIX CORRECT  ANSWER
S IN A ROW.":PRINT:PRINT"IF YOU
DON'T KNOW"
21 PRINT"THE CORRECT      ANSWER
 TO ANY   ROLL, I WILL GIVE"
22 PRINT"IT TO YOU.":PRINT:PRINT
23 PRINT"FOR A SAMPLE ROLL PRESS
 THE ENTER KEY ";:INPUTA$:H=1:CL
S:GOTO41
24 PRINT"PRESS THE ENTER KEY TO
BEGIN THE GAME ";:INPUTA$
27 PRINT"  THE CORRECT  ANSWE
R IS";T:PRINT:PRINT"SEE HOW EASY
":PRINT"IT IS!!!!!!"
28 PRINT"PRESS ANY KEY TO BEGIN
";:A$=INSTR$(1)
35 H=2:C=0
37 CLS:OUTPUT"I'M SHAKING",10,65
,1:OUTPUT"THE DICE...",10,58,1
39 H=2:C=0
40 FORI=1TO2000:NEXTI:CLS
41 Y=24:T=0:FORX=15TO85STEP17:N=
INT(6*RND(1))+1
42 IF(N/2-INT(N/2)=0)THENGOTO50
44 T=T+N-1
50 FORI=XTOX+13:FORJ=YTOY+6
60 PLOTI,J,3:NEXTJ:NEXTI
100 ONNGOSUB1010,1020,1010,1030,
1010,1020
110 ONNGOSUB1070,1070,1020,1020,
1020,1030
120 ONNGOSUB1070,1070,1070,1070,
1030,1040
125 IFH=3GOTO2015
130 NEXTX
135 IFH=1GOTO27
150 PRINT"TYPE IN THE":PRINT"ANS
WER 'YOU THINK":PRINT"IS CORRECT"
155 PRINT"IF YOU HAVE NO   IDEA
TYPE 'P' FOR PLEASE HELP ME ";
160 A$="":INPUTA$
161 CLS:ONPGOTO1160
165 A=VAL(A$):IFA=TGOTO200
170 C=0:IFASC(A$)=80GOTO230
175 GOTO250
200 C=C+1:OUTPUT"**** YEA!!!! **
**",5,40,5
201 PRINT"THAT'S";C;"IN A ROW CO
RRECT. ";
203 IFC<6GOTO205
204 FORL=1TO3000:NEXTL:GOTO1100
205 PRINT"LET'S   SEE YOU DO IT
   AGAIN";
210 FORL=1TO3000:NEXTL:GOTO40
230 PRINT"DON'T GIVE UP SO EASIL
Y, THE":PRINT"ANSWER IS";T
231 PRINT"TYPE IN 'S' FOR  IT'S
SIMPLE I'LL TRY AGAIN!!";
232 INPUTA$:GOTO40
```

PETALS, cont.

```
250 OUTPUT"**** BOO!!!! ****",5,
40,2:PRINT"YOU JUST AREN'T  THIN
KING":PRINT"THE"
251 PRINT"ANSWER IS";T:PRINT"TYP
E IN 'A' FOR":PRINT"PETALS AROUN
D THE  ROSE
252 INPUTA$:GOTO40
1010 PLOTX+6,Y+3,2:PLOTX+7,Y+3,2
:RETURN
1020 PLOTX+10,Y+1,2:PLOTX+11,Y+1
,2:PLOTX+3,Y+5,2:PLOTX+4,Y+5,2:R
ETURN
1030 PLOTX+3,Y+1,2:PLOTX+4,Y+1,2
:PLOTX+10,Y+5,2:PLOTX+11,Y+5,2:R
ETURN
1040 PLOTX+3,Y+3,2:PLOTX+4,Y+3,2
:PLOTX+10,Y+3,2:PLOTX+11,Y+3,2:R
ETURN
1070 RETURN
1100 CLS
1105 PRINT"CONGRATULATIONS!!":PR
INT:PRINT"YOU ARE NOW ONE  OF TH
E HOLDERS OF"
1110 PRINT"THE  'SECRET OF  THE
ROSE'.":PRINT:PRINT"JUST AS YOU
CAME THROUGH MANY"
1112 A$=INSTR$(1)
1115 PRINT"TRAILS AND MANY  FRUS
TRATIONS TO  CRACK THE SECRET YO
U"
1120 PRINT"MUST NOW VOW TO  ALLO
W ALL OTHERS TO SHARE THE SAME"
1125 PRINT"FRUSTRATIONS.":PRINT:
PRINT"   EVEN UNDER "
1126 PRINT"THREATS TO YOUR  LIFE
 YOU MUST    PLEDGE NEVER TO "
1127 PRINT"TELL ANYONE THE
'SECRET'.
1135 Y=1:N=2
1140 PRINT:PRINT" WOULD YOU LIKE
   TO CONTINUE AND FRUSTRATE AND
THERTYPE Y/N ";
1145 INPUTA$:IFA$="Y"ORA$="YES"G
OTO10
1150 CLS:END
1160 CONT150
2000 CLS:RESTORE:C=0:H=3
2003 CLS:T=0
2005 FORX=15TO85STEP17:READN
2010 GOTO41
2015 NEXTX
2020 C=C+1:IFC=3THENRESTORE
2025 PRINT"THE ANSWER IS";T
2030 PRINTTAB(6);"ENTER"
2031 PRINT"1) CONTINUE TEST"
2032 PRINT:PRINT"2) START GAME"
2035 INPUTZ$
2036 IFZ$="1"GOTO2003
2040 GOTO10
2045 DATA4,1,6,3,6,5,6,5,4,4,6,5
,6,2,2
Ok
```

PROGRAM CORRECTION

The NUMBER BASE CONVERSION program in Issue 2 has an error. Line 200 should read -

```
200   INPUT "INPUT BASE" ; B1
```

J. Bishop of Topeka, KS had to point this error out to me twice before I checked the listing to see that he was correct. My apologies to him and all the readers.

SLAGH SYSTEM SERVICES

BOX 438
YPSILANTI
MI 48197
(313)434-0033

PRODUCTS AND SERVICES

PRODUCTS

U80 General The U80 line of Interface Ports can be used with either Intel 8080 or Zilog Z80 Microprocessors. The U80 line features solderless installation, Baud rates from under 100 to over 9600. they can supply a 20mA current loop output, or operate RS232C. They feature a Motorola Asychronous Communications Interface Adapter instead of a UART, for simpler operation.

U80M This U80 is specifically designed for the Interact. It can be easily assembled and installed with the complete documentations and instructions provided. If the Interact is ever upgraded with more RAM or ROM, the U80M can be easily upgraded to the U80R.
Complete Kit $54.95 Assembled and tested $69.95

U80R This U80 is for general use, and can be set easily for any address with on-board dip switches. Compatible with any 8080 or Z80 machine.
Complete Kit $84.95 Assembled and Tested $99.95

D825P Cable with DB25P connector for interface with most modems. $8.95

Custom Custom cabling service for your application available.

T8011 Simple to operate terminal program, allows Interact with U80M Port to be used with modem to communicate over phone. $8.95

T8013 New terminal program that allows transfer of data from one computer to another, as well as providing expanded terminal operations $12.95

P8011 Printer output tape that 'overlays' Level 2 Basic and Fast Graphics Basic to provide printer output. $8.95

MONITOR This is the Holloway Hi-Lo Monitor, featuring High RAM or Low RAM operation and features a resident dis-assembler. Documentation includes tape formatting and tape writing routines.

SERVICE We provice complete repair and upgrading service for the Interact and many other electronic items. Call for full details.

DESIGN Slagh System Services provides design and consulting service for a variety of business and personal needs.

INFO Thursday Evening is CALL-IN NITE. Slagh System Services is available Thursday from 6p.m. till midnight, eastern time, to discuss problems, answer questions, and exchange ideas. Give us a call. We're glad to share information and ideas with everyone. WHY KEEP SECRETS when Interact has gone out of business.

ITEM	QUANTITY	PRICE

SUB-TOT _____ SHIPPING
4% TAX _____ Tapes.50 ea
(Mi. Residents) Port 3.00
Shipping _____
TOTAL _____

QUESTIONAIRE

What do you want to see in INTERACTION ?	MORE	ABOUT THE SAME	LESS
BASIC PROGRAMS	___	___	___
MACHINE LANGUAGE PROGRAMS	___	___	___
GAME PROGRAMS	___	___	___
PRACTICAL PROGRAMS	___	___	___
TECHNICAL ARTICLES	___	___	___
GENERAL ARTICLES	___	___	___
CLUB NEWS	___	___	___
ADVERTISEMENTS	___	___	___

WHAT WAS THE WORST ARTICLE OR PROGRAM IN 1980 ?

WHAT WAS THE BEST ARTICLE OR PROGRAM IN 1980 ?

WHAT SHOULD BE DONE TO IMPROVE INTERACTION ?

HAVE YOU WRITTEN ANY PROGRAM THAT YOU WOULD CONSIDER CONTRIBUTING FOR PUBLICATION ?

IF NOT, WHY (NOT GOOD ENOUGH OR WANT TO SELL) ?

DO YOU WANT A LIST OF SUBSCRIBERS IN YOUR STATE (REMEMBER SOME STATES HAVE ONLY ONE INTERACT OWNER THAT I KNOW ABOUT) ? YES ___ NO ___

RENEWAL FORM

___ YES, I WANT TO SUBSCRIBE TO INTERACTION FOR 1981

If you do renew your subscription for 1981 please enclose a check for $10.00 . Please make it payable to

STEPHEN COOK INTERACTION NEWSLETTER

(Canadians please mark your checks US Funds or send Postal money order in US funds)
(Overseas airmail rate $16.00)

**** SEND THIS WHOLE PAGE TO ****

STEPHEN COOK
INTERACTION NEWSLETTER
15356 PREVOST
DETROIT, MI 48227

INTERACTION Newsletter

1980 INDEX

Compiled by:
Jerry Lerner

NOTE: I,2,3 = Vol. I, No. 2, Page 3

ARMCO

INTERACTION NEWSLETTER INDEX 1980 1981

BASIC INTERPRETER

BASIC PROGRAMS, APPLIED

ARMCO

INTERACTION NEWSLETTER INDEX 1982

BASIC MAP	HOLLOWAY P2-5 HENDRICKSON P6-13	III, 1	P3-8	III, 2			
FORTH	P14-16 (22)	III, 1	P11-13	III, 3	(AD) P19	III, 3	
EDITEX - ASSEMBLEX (REVIEW)	P17-21	III, 1					
ML (AD)	22	III, 1					
TAPE MASTER (AD)	23						
MACHINE LANGUAGE LEGGETT	P1-2	LINE III, 2	P1-3	SLIDING TONE III, 3	P10-12	SOUND III, 4	
FOREST FIRE	P9-11	III, 2					
SABRE PORT (REVIEW)	P11-14	III, 2					
FIVE DICE GAMES	P14-18	III, 2					
✓ WORD GUESS	P18-20	III, 2					
WORD PROCESSOR (AD)	P20	III, 2					
ICS (AD)	P21	III, 2					
IOS (AD)	P22	III, 2					
MORSE TERMINAL (AD)	P23	III, 2					
MICROMODEM COMMUNICATOR	P3-5	III, 3					
WORD SEARCH	P6-8	III, 3					
SKETCH PAD (REVIEW)	P8	III, 3					
COMET 8300R (REVIEW)	P9-10	III, 3					
EDU-BASIC OVERLAY (AD)	P10	III, 3					
LITTLE LETTERS / LITTLE NUMBERS	P14-16	III, 3					
CSAVE BASIC AND PROGRAMS	P17	III, 3					
6502 (VIC)	P18	III, 3					
PRINTER OVERLAY (AD)	P19	III, 3					
PRINTING THE SCREEN	P1-5	III, 4					
HEAD ALIGNMENT	P6	III, 4					
I/O IN FORTH	P7-10	III, 4					
RANDOM CRAZY	P12-13	III, 4					
WORM	P14	III, 4					
MIRROR IMAGE	P14-15	III, 4					
RELOCATE HI-LO MONITOR	P15-18	III, 4					
REBUILD JOYSTICK	P18-20	III, 4					
MULTI BASIC EDITOR (AD)	P21	III, 4					
FORTH AD	P22	III, 4					
JOYRE EXPANSION	P23	III, 4					

FORM G-5670

ARMCO

INTERACT 80/81

GRAPHICS

BANNERS I,4,3: I,6,14
CIRCLE I,3,4
COLOR OF TEXT I,3,15
COLOR SEARCH II,2,15
DICE II,6,20
DOTS II,3,19
FLAGS II,2,11
KALEIDOSCOPE I,3,3: II,2,9
RCHRAD I,4,2
ROSES ~~I,2,4~~ I,3,3
RPLOT I,3,7
STAR PLOT II,2,9
RFILL 1,2,4
POLAR PLOTS II,2,12
SCROLLING SIDEWAYS II,3,12

MACHINE LANGUAGE

ANALOG/DIGITAL I,5,3 (FASTAD)
BASIC CURSOR II,3,10
DISPLAY FORMAT II,3,1
DRAW A BLOCK ~~II~~, 2,3
MEMORY TEST (BOBO ASSEMBLER) II,1,9
MERRY OLDSMOBILE I,3,9 : I,5,11
MICROVIDEO MONITOR
ADDING BANNERS I,4,3
BREAKPOINT I,4,3
LOWER CASE II,3,9
PEEK ROM I,3,2
PEEK RAM I,1,2
POKE SCREEN I,1,2: I,1,4: I,4,5
RCHRAD I,4,2
RFILL I,1,5 I,2,4
R PLOT I,3,7
SOUND I,4,8

GAMES, STRATEGY

✓ CAMEL II,5,5
? CANNON II,3,2 (Joystick)
CHECKERS I,6,5
COLOR MASTERMIND I,5,7
✓ CONCENTRATION I,6,12
? EGGHEAD II,1,15
FIRE 3 II,6,3
✓ GAME OF LIFE II,4,11
✓ ? HAMMURABI II,1,11 (Poke statement)
✓ L GAME II,5,14
MICROCHESS BUG I,3,14
PETALS AROUND THE ROSE I,6,16
✓ REVERSE II,1,8
→ * ROTATE II,4,5 (Good)
? JOTTO II,3,13
SHOOTING STARS II,2,3
SIMON I,1,5
? WARI II,6,13
✓ WUMPUS II,2,16

GAMES, ACTION

BREAKTHROUGH (FASTAD) I,5,3
GRAVITY I,2,9
HORSE RACE II,6,17
LUNAR LANDER, I,5,14
VEGAS II,6,8

GAMES, EDUCATIONAL

ADDITION PRACTICE II,1,6
BIG LETTERS I,5,17
COMPUTE-A-COLOR BUG I,2,2
COUNT THE LETTERS I,6,10
GUESS THE ANIMAL I,4,10

INTERACT 80/81

PRINTER

PRINT THE SCREEN II,2,1
BASIC MAILING LIST II,3,5
CHRISTMAS TREE II,6,11
12 DAYS OF CHRISTMAS II,6,16
HOLIDAY SUPRISE II,6,22
PHONE NUMBER WORDS II,1,9

REVIEW

MV MONITOR (ML) I,3,13
MV HAMMURAMI I,3,13
MICROCHESS BUG 1,3,14
QUEST (EDUBASIC) I,6,9
MV RS232 II,1,2 II,1,3
HI LO MONITOR II,2,7
ROM MONITOR II,2,7 II,3,8
ROM OPERATING SYSTEM II,2,19
UBOM PORTKIT II,3,7
LEONARDO GRAPHICS II,3,23 II,5,18
SPACE INTRUDER II,5,12
GOLD MINER II,5,12
INTERACT CHAT II,5 13

HARDWARE

ERASE HEAD I,1,2
KEYBOARD I,3,10 I,3,11&12
KEYBOARD I,5,16

MUSIC & SOUND

CHRISTMAS MUSIC I,6,15
C-MAJOR SCALE II,3,21
DATA SONGS I,3,10
SOUND I,4,8
SOUND VIA RADIO I,3,9

EDU BASIC

BUG I,2,1
MISC I,6,8
SOUND I,4,8

SOFTWARE

LIFE R. SCHNAPP II, 1, 8 (I, 4, 15)

ELIZA J.A. MILLER II, I; 13

SCHEMATIC JIRD II 1, 13

UNO CARD GAME (BASIC) } II, 1, 17
COLOR BAR G.A. LEGGETT }

ROAD RACER }
INCOMETAX } W. FAVORITE II, 2, 6
HORSE RACE } II, 3, 22

88 CONSTELLATIONS } G A LEGGETT
X-Y MONITOR } II, 3, 9
II, 4, 19 II, 3, 15
SPACE INTRUDERS D. YOHE

BASEBALL }
MORSE CODE } (BASIC) A. ALBERT II, 3, 15
SPEAK & SPELL }

GOLDMINE (BASIC) (ML) II, 5, 19 W. DEACON II, 4, 14

HOME OWNERSHIP E. ZOLANDZ II, 4, 19

BASIC
MOON LANDER }
9K GRAPHICS
FLASH CARDS } A. ALBERT II, 5, 19 (II, 6, 22)
9K GRAPHICS
SLOT MACHINE }

MICROVIDEO
TROLL HOLE ADVENTURE I, 6, 7

QUEST (EXT-BASIC) D. SCHWAB I, 6, 9

MICROVIDEO
HAMMURABI I, 3, 13

MICROVIDEO
EARTH OUTPOST I 1, 3, 17

MICROVIDEO
ATTRO-LOGIC 1, 3, 17

8080 DISASSEMBLER IN BASIC
D.J. SCHWAB II, 1, 17 (I, 4, 16) II, 2, 18

ROM MONITOR # 1 ROM # 1 AROM-1
RMAPPER (II 3, 6) II, 2, 10
W. HENDRICKSON II, 1, 18

ROM OPERATING SYSTEM
(REVIEW II, 3, 8)
R.A. FERRIS II, 2, 19; II, 3, 22

LEONARDO GRAPHICS

H. HOLLOWAY II, 3, 23

8080 FOR EVERYONE.

G.A. LEGGETT II, 4, 19

INTERACT CHAT
FULL ASCII R. PASCO II, 4, 19

GRAPHICS
FASTLINE BASIC OVERLAY }
HILO MONITOR H. HOLLOWAY } I, 6, 10 (I, 4, 16)

MICROVIDEO MONITOR I, 3, 13

ARMCO

MV INN

TAPE MOTOR CONTROL	P2	I, 1
ADD ML TO BASIC	P3	
GAME ADD ON	P3	
SMALL BYTES	P4	
SIDEWAYS SCROLL	P4	

SMALL BYTES	P4-5	I, 2
VOTRAX TYPE N TALK	P 3	
INTERACT HISTORY	P6-7-8	
POKE & PEEK	P8 9 11	
USR	P10	
JOYSTICK PORT	P11	
FASTER BASIC	P11	
32K EXPANSION	P1	

S

STRINGY SPONGE (92K)	P 1	II, 1
GAMESMANSHIP EXPLORED	P 2,3,8	
ASSEMBLER / EDITER	P 4	
PRINTER/MODEM SWITCH	P4	
20 QUESTIONS	P6-7	
KEYBOARD TROUBLESHOOTING	P5	
32K Corner	P5, 10	
SMALL BYTES	P12	

KEYBOARD	P1	II, 2
VIDEO CALC & DREAM	P4	
INTERACT GENEOLOGY	P5	
SOUND	P6	
CASSETTE PLAYER REPAIR	P7	
PROGRAMMING	P8	
SUBMITTING SOFTWARE	P9	
BETTER MOUSE TRAP	P10	
FEEDBACK	P11	
RESTORE	P12	
MICROTEXT	P13	
HELP	P14	
WAKE UP INTERACT	P16	

ARMCO

1/ICC

SIMPLE SWITCHING	P2-3	10/81
USEFUL SUBROUTINES	P4-5	10/81
MUSIC AID	P5-6	10/81
CHASE	P7-9	10/81

11/81 CAMEL

LOOPS	P2	12/81
SIDEWAYS SCROLL	P3	12/81
POT PORT	P3-4	12/81
HAMURABI	P6-8	12/81
ALPHABETIZE	P8-9	12/81

PEEK	P6	1/82
CHASE (MOD)	P7 P10	↓
HEAD ALIGN	P7	
MYSTERY HOUSE	P8-10	

SOUNDS	P2-3	2/82
CHECKERS	P4-6	↓
BOMBS AWAY	P7-8	
COLOR SEARCH	P9	
VOCABULARY QUIZ	P10	↓

P2 5/82

P1 5/82

ATARI JOYSTICK	P2	3/82
CAMEL (MOD)	P3	↓
SMALL CHARACTERS	P3-	
DRACULAS BLOOD	P5-7	
MUSEUM	P8-10	

MX-80	P1	4/82
BATTLESHIP	P4-5	↓
VEGAS SLOT	P6-7	
BUNKO	P8-10	

P4,5,6/82

BATTLESHIP II	P6-7	5/82
BRIDG-IT	P8-9	↓
INTERACT PROGRAM FILES	P10	
EXPAND MICRO VIDEO MONITOR	P3-4	

ARMCO

NICC

SOUND ROUTINE	P1-3	6/82
MEMORY SAVE	P5-6	
EXPANDO CHR$	P8	
THE RACE	P9-10	
R-PLOT	P1-6	78/82
RESTORE	P8-9	
HIGH ROLLER	P10	
TV MONITOR	P4	9/82
INTERACT COLORS	P6	
SLOT MACHINE (8K)	P7	
DECIMAL TO HEX	P9	
PRINT THE SCREEN	P3-8	10/82
WORD PROCESSING	P9	
BYTE-CHECK	P10	
LIFE	P5-7	11-12/82
HORSE RACE	P8-9	"
ML MUSIC	P2	1/83
TYM SHARE MODEM	P3	
ASTEROIDS	P7-8	
TOWERS OF HANOI	P8-10	
		2/83
INKEY$ (GET KEY - 24529)	P1	3/83
LOGIC QUIZ	P2	
INTERACT REPAIRS	P3-4	
MINE SWEEP	P7-10	

FORM G-5670