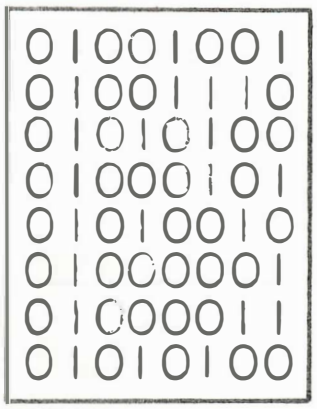
$ 3.00 A COPY
$15.00 FOR 6
ISSUES A YR.



INTERACTION INTERNATIONAL

A MAGAZINE FOR INTERACTORS AND FOR ALL PEOPLE WHO WANT THE COMPUTER KNOWLEDGE OF TOMORROW, FOR TODAY

JAN.- FEB. 1983 VOL. IV NO. 1

TABLE OF CONTENTS

CREDITS

EDITOR/PUBLISHER	George A. Leggett
FEATURES EDITOR	Lora A. Leggett
PROGRAMMER	George A. Leggett
DRAFTSMAN	Daryl Waananen
PRODUCTION ASSISTANT	George E. Leggett
ILLUSTRATOR	George A. Leggett
FRONT COVER DESIGN	George A. Leggett
PRINTER	Ted McAdam

THE WORD IS GIVEN:

PUBLISHER'S STATEMENT
By
George A. Leggett

It goes without saying that there has been a change. Not only in the name, but in the magazine itself. It is of my opinion and I hope you will agree, that the change is for the better. The magazine's purpose is to benefit all of you by giving you a more in-depth and detailed look at your Interact, BASIC programming, Machine Language, and Hardware. There's only one thing that could be wrong with this magazine. That is my mouth. I HAVE A BIG!!! MOUTH!!! (This is one of my favorite lines from Jackie Gleason's The Honeymooners) But this is very true. I do have a tendency to go on about some things I'm talking about for many pages as you will see. I do this not to bore you. I do this to inform you, to make sure everything is absolutely clear to the best of my abilities. The reason for this is that over the years I have received so many letters from people who always want to know more about a subject which I discuss or even those subjects discussed by someone else. I have found that by no means does it hurt to say too much about a program or article. Perhaps the opposite is true; saying too little can be more trouble than saying too much. After all, if you feel you know what I am trying to say and you feel you've had enough, you can always turn the page. But if there's not enough, if there's no page to turn to, it takes time for you to figure out what someone is talking about and time for me to do a follow-up and all in all, nothing but trouble and headaches. So, unless I hear otherwise, I will try to keep a very open and detailed description of at least all programs and other work that I do, including Hardware and Machine Shop Talk.

You may also notice a structural change in the magazine. I shall try with the utmost to keep everything in its own category so you may easily follow a subject. If some of you are not interested in a BASIC program, why should it pop up in the middle of a Machine Language program? If, however, you are only interested in what someone is advertising, why would you want to hunt through 20 pages to find it? Then, all of us are interested in different things at different times. That is why the Table Of Contents will always appear on Page 1.

There will be growing pains like any new endeavor. Right now, what we have to put up with for a little while is the fact that at this moment none of you know Vol. 4 no. 1 is done and ready to be in your hands. Though I cannot afford to pay a professional printer I shall do the best that I can to insure that everyone gets a perfect copy. Nothing less. In the future, once the ball is rolling, it will be as before a bound magazine. It will be bound in the same manner as Vol. 3 no. 1, the first of the 1982 year. I felt that the stapling down the side like a book was one of the best ways. If any of you feel differently, please let me know. But for now it will be 24 single pages stapled together and sent in envelope first class. Obviously, you can see that it's costing me more and taking me a lot longer to do this. But I am more than willing to make the sacrifice rather than to let you wait another single day for INTERACTIONS. This is inexcusable!!!

As of now, we are using two methods of putting together the magazine. One is a Remington Standard typewriter which my wife runs and you can always tell that by its upper and lower case. The other is a Teletype which you will know by its upper case only. When I do an article, I use the Interact as a word processor, which is an area I will touch on in an upcoming issue. When my wife transcribes one of my articles, I dictate on a cassette tape from which she transcribes it for me. Her English and spelling will be far superior to mine. At this time I would like to say that if she or I make a typo you'll have to excuse it. If I misspell a word or use improper grammar, tough! I'm not running an English course. When I write an article I sit down and begin at the first paragraph and go. I never look back at what I wrote. I start from the first sentence to the last in sequence. So, if you find any errors of this kind, please pardon me. This magazine is to learn about computers, their uses and what you can do with yours. So, for English, you'll have to go somewhere else. But I hope I'm not that that you cannot figure it out.

The reason this issue and probably the next was written completely by me is because as of now, I have no other input. When I do hear from you or when information is turned over to me via Steve Cook ... So I put it upon myself to if I must write every article, every program all year long but you will get a magazine every two months come hell or high water. There is one exception--if I died, but I'm sure I'll figure out a way to get it to you if that happened. You deserve the best I can give you and that's all I will give you is my best. From Vol. 4 no. 1 to Vol. 99 no. 6 You will get my best and nothing less. As soon as you can, send me some material. I welcome all and any. But for now, sit back, enjoy this magazine please send me any and all comments whether good or bad.

About This Issue: There's a new series called The Hard Facts Of Life. If you want to hear a little more about my life story, you will find it there, no reason to repeat here. I hope you will at least read it through. I'm sure it will open a whole new world for your Interact which you never before thought could exist.

Also in this issue, as I will try in all upcoming issues, is to maintain a theme. A group of programs will all have the same theme in common. I don't know how possible this will be down the road, but at least for a while I shall try and give it my best. The theme for this issue is an article called Fast Graphics and Get Key. Please read this first as every BASIC program in this issue deals with this article. In upcoming issues, we will have programs dealing with card games, and in another issue, programs dealing with an introduction to basic electricity and electronics. So, there are some of the upcoming categories. I hope you'll be looking forward to reading them as much as I look forward to doing them.

For now, this is all I have to say, and I thank you for the time in reading this first of what I hope will be many chats like this. Finally, I will leave you with this: Why the name change? Well, I felt that a Detroit user group or a Detroit Newsletter or Interact Newsletter is no longer appropriate. There are many people around the world who own an Interact, and while this magazine is mainly for Interactors and will be that way, it's also for computer people. Computer people in all walks of life. Perhaps I am dedicating it more to the computer person who has an ultimate love for the computer over anything else. I know of one person. And that's why I'm here doing this. Let me know if you're another.

To coin a phrase: From one computer lover to another
This one's for you.

Sincerely,

George A. Leggett

RANDOM REMS
By
Lora A. Leggett

This is my first chance to formally introduce myself to all of you, although I have met many of you over the phone and through helping George with correspondence. I should like to begin by saying that it has been both challenging and pressuring to put together a magazine of what we hope is of fine quality with a little something for a variety of interests. But, as we near the end of the rainbow and pages are completed, numbered, and have meaning, it is an exciting feeling, too. Comparable, I would say, to the final weeks before the birth of a new baby. I hope our baby, INTERACTION INTERNATIONAL, has a long and purposeful life.

We wish to salute and thank our very first subscribers: Richard G. Young of Salem, OR and Fred M. Garver of Grand Haven, MI.

Three cheers to Micro Video on the new keyboard for the Interact. It should give every Interact many more years of Interactive life.

Your responses are our life blood--Keep them coming!!!

FAST GRAPHICS AND GET KEY COMMANDS FOR LEVEL II BASIC

BY

GEORGE A. LEGGETT

THIS IS A VERY SHORT PROGRAM THAT CAN GIVE LEVEL II BASIC A LOT MORE POWER WITHOUT USING A LOT OF MEMORY. ACTUALLY THERE ARE TWO PROGRAMS IN THESE 7 LINES. FOR THOSE OF YOU WITH MICROSOFT 8K FAST GRAPHICS BASIC YOU WILL NOT NEED THIS PART OF THE PROGRAM BUT DO NOT GO AWAY THERE WILL BE SOMETHING FOR YOU LATER ON. FOR NOW I SHALL EXPLAIN WHAT FAST GRAPHICS CAN DO.

IF YOU HAVE SEEN ANY OF THE PROGRAMS USING 8K FAST GRAPHICS YOU KNOW THE FORMAT IS PLOT X,Y,C,XL,YL WELL THIS PROGRAM WORKS THE SAME WAY. LINES 10, 15, 59999, 60000, AND 60010 CONTAIN THE ROUTINE TO DO THIS. LINES 10 AND 15 POKE THE ROUTINE INTO THE MACHINE AT THE START OF THE PROGRAM AND AS LONG AS YOU DO NOT USE CSAVE* OR CLOAD* YOU WILL NOT LOSE IT. IF YOU PLAN TO USE THESE TWO FUNCTIONS SIMPLY MAKE THE POKES A SUBROUTINE AND CALL THEM UP AGAIN AFTER YOU HAVE FINISHED WITH CSAVE* OR CLOAD*. LINE 59999 IS THERE ONLY TO MAKE SURE YOU DO NOT RUN INTO THIS SUBROUTINE BEFORE LOADING IN THE CORRECT PARAMETERS. I CHOSE LINE 60000 AND UP SO THAT IT IS OUT OF THE WAY OF MOST PROGRAMS AND COULD EASILY BE APPENDED TO MOST PROGRAMS.

HERE IS HOW IT WORKS. WRITE YOUR PROGRAM ANY WHERE YOU WANT BETWEEN LINES 30 AND 59999. WHENEVER YOU NEED FAST GRAPHICS LOAD THE FOLLOWING VARIABLES;

XA = STARTING X AXIS
YA = STARTING Y AXIS
CO = COLOR 0, 1, 2, OR 3
XL = LENGTH OF X LINE
YL = LENGTH OF Y LINE
GOSUB 60000

THERE'S YOUR BOX OR RECTANGLE OR LINE IN THE SPEED OF MACHINE LANGUAGE. A TYPICAL LINE COULD LOOK LIKE THIS;

```
100 XA=10:YA=60:CO=2:XL=50:YL=20:GOSUB60000
```

THAT IS ALL YOU HAVE TO DO TO MAKE IT WORK. Vol 4,2,7

NOW SOMETHING FOR EVERYONE THAT TAKES ONLY TWO LINES AND GREATLY IMPROVES YOUR CONTROL OVER YOUR BASIC PROGRAMS. IT IS THE GET KEY FUNCTION. I HAVE SEEN THIS FUNCTION IN MANY OTHER COMPUTERS AND ALWAYS WISHED INTERACT HAD IT. I FIRST USED IT IN MY VIC-20 COMPUTER AND SINCE THEN I KNEW INTERACT HAD TO HAVE IT. *WE HAVE THE INSTR$(N) FUNCTION BUT THIS FUNCTION ALWAYS WAITS FOR A KEY TO BE PRESSED. WITH THE GET KEY THERE IS NO WAITING SO YOU CAN HAVE YOUR PROGRAM RUNNIG AND STILL EXTRACT INFROMATION FROM YOUR KEYBOARD SIMPLY DO THE FOLLOWING ANY TIME YOU NEED TO USE IT;

```
K=PEEK(GK)
```

VARIABLE K WILL HAVE THE ASCII OF WHATEVER KEY IS PRESSED.
NOTE: THE ASCII VALUE WILL BE UPPER AND LOWER CASE
WHEN YOU TRY THIS OUT YOU MAY FIND THAT THERE WILL BE SOME DELAY. THIS WORKS JUST LIKE ANY OF THE CONTROL COMMANDS YOU MUST WAIT

UNTIL THE BASIC COMMAND IS EXECUTED BEFORE IT SCAN THE KEY BOARD. IN FACT THIS IS WHERE THE ROUTINE IS INSERTED. RIGHT IN THE BASIC KEY BOARD LOOP. IF YOU WANT TO USE CSAVE* AND CLOAD* YOU WILL HAVE TO POKE THE ORIGINAL BYTES 24650,231 AND 24651,7 BACK OR YOU WILL BE IN TROUBLE. BASIC WILL JUMP INTO YOUR CSAVE* OR CLOAD* DATA AFTER IT IS DONE AND NEVER RETURN TO BASIC AND MOST LIKELY WIPE OUT EVERYTHING IN THE COMPUTER. AFTER YOU ARE DONE JUST REPOKE THE ROUTINE AND AWAY YOU GO.

ALL THE PROGRAMS IN THIS ISSUE HAVE BOTH ROUTINES IN THEM SO YOU CAN STUDY HOW THEY ARE USED IN THEM. I HOPE YOU CAN BENIFIT FROM THEM AS MUCH AS I HAVE.

```
1 REM FAST GRAPHICS AND GET KEY CODE NAME  "FG+GK"
2 REM GEORGE A. LEGGETT 20562 WOODWARD MT. CLEMENS, MICH 48043
3 REM DECEMBER 9,1982
10 POKE19215,25:POKE19473,9:POKE19474,74:PK=18953:POKEPK,1:POKEPK+1,16
15 POKEPK+2,74:POKEPK+3,205:POKEPK+4,162:POKEPK+5,5:POKEPK+6,201
20 POKEPK+13,205:POKEPK+14,231:POKEPK+15,7:POKEPK+16,50:POKEPK+17,21
25 POKEPK+18,74:POKEPK+19,201:POKE24650,22:POKE24651,74:GK=18965
59999 END
60000 POKE18960,YL:POKE18961,XL:POKE18962,CO:POKE18963,77-YA
60010 POKE18964,XA:US=USR(0):RETURN
OK
```

THE I'S DON'T HAVE IT

GEORGE A. LEGGET 20562 WOODWARD MT. CLEMENS, MICH. 48043

I BEFORE E EXCEPT AFTER C MAY BE A GOOD RULE IN ENGLISH BUT LIKE EVERY RULE THERE ARE EXCEPTIONS TO IT.

THIS ARTICAL IS PROMPTED BY AN ARTICAL I READ IN POPULAR COMPUTING DEC. 1982. IT HAS TO DO WITH THE USE OF CERTAIN VARIABLES. MAINLY THE I (EYE) VARIABLE. WHY USE IT? IT IS VARY DIFFICULT TO DISTINQUISH IT USE FROM THE NUMBER ONE.

I WONDER HOW MANY OF YOU HAVE SPENT COUNTLESS HOURS DETERMAINING IF IT IS AN I OR A ONE NOT TO MENTION I1 , I(I), I(1) I(I1) AND MORE. WHY DO PROGRAMMERS DO THIS? MAYBE IT IS SO OTHER PROGRAMMERS WON'T KNOW HOW THE PROGRAMS WORK! WHEN YOU CONSIDER THERE ARE 25 OTHER LETTERS IN THE ALPHABET WITH COUNTLESS THE RECORD I DO USE I IN ONE AND ONLY ONE SITUATION AND THAT IS WHEN I JOIN IT WITH THE DOLLAR SIGN SYMBOL AS IN THE FOLLOWING I$=INSTR$(1). THIS WAY I (PARDON THE I) ALWAYS KNOW ANY INSTR$ I$ IS ALWAYS ASING TO I$. THERE ARE MANY OTHER VARIABLES IN MY PROGRAMS THAT ARE ASSIGN A SPECIFIC JOB IN A PROGRAM AND EVERY PROGRAM I DO. IN THIS WAY I DO NOT SPENT A LOT OF TIME FINDING OUT WHAT SOMETHING DOES IN A PROGRAM.

AS IN ANY ARTICAL I WRITE OR REVIEW I WELCOME ALL COMMENTS AND SHALL TRY TO PRINT THEM IN FUTURE ARTICALS.

OK

HI-Q

By George A. Leggett, 20562 Woodward, Mt. Clemens, MI 48043

HI-Q is a peg jumping game played on a game board. I first played the game some 20 years ago. Before that time, my father had a small version of it called YOGO a pocket-sized version. The pegs were way too small for me to handle, therefore, several years later, they bought me Giant Double HI-Q. There are two game boards in this set and the pegs are easy to handle. The object of the game is simple. You must overjump pegs, always jumping horizontally or vertically into an empty hole. At the beginning of the game there are 32 pegs with one empty hole in the middle. Thus, you only have four possible opening moves, from the North, South, East or West. From there, the possibilities are endless. The game has automatic scoring based on HI-Q and Yogo. The real object of the game is to leave one peg in the middle. I've played this and seen it played for over 20 years and have never seen it happen. But, believe it or not, on the day I finished writing the program and gave the game its first full test run, I did just that--I left one peg--right in the middle! Somehow a freak of nature. I hope you will enjoy it.

You use the left Joy Stick to move your flashing X curser up, down, left or right. Don't worry about a wrong move. The computer will not let you make an illegal move. When you can no longer make any jumps, turn the POT fully clockwise. You will see the message and hear the tone: Press Fire Button to end play. This is a precaution in case you accidentally move the Pot during play. If you do not wish to end play, turn the Pot fully counterclockwise and continue playing. Once you hit the button however, your score is tallied up and the game is over. Whereby you may elect to play again. I hope you have as much fun with it as I have had over the years. A truly enjoyable solitaire game.

For variations on the game if you really become an expert, you may move the starting hole anywhere you wish on the game board. This is accomplished by adding a few simple statements to the program.

The game board is set up in a 7 by 7 matrix, however, not all squares are used to play the game because of the shape of the game board. Lines 30 and 35 are the data that up the playing board. Thus, spaces which are not used for the game ar represented by a 0, spaces with pegs are 1 and spaces with a "hole" for a peg are referred to as 2. Thus, all 0 spaces remain the same, and your space with a 2 is now moved from the middle square, or the fourth hole in the fourth row, to anywhere you wish. Rather than retyping data and risking making a mistake, here is an easier way to change the starting hole. Add a line 55 A(4,4) = 1 followed by the statement A(whatever square you wish in this example, 1,3) = 2 We are putting a peg in 4, 4 of the board and taking away a peg in 1, 3 This is much quicker and easier and you can do it in 30 seconds.

Good luck and let me know if you end up with what is called a perfect score! If you have a solution, I'd love to see it.
To all of you, Happy Jumping!

```
1 REM NAME"HI-Q"
2 REM GEORGE A. LEGGETT 20562 WOODWARD MT. CLEMENS, MICH 48043
3 REM NOVEMBER 30, 1982
10 POKE19215,25:POKE19473,9:POKE19474,74:PK=18953:POKEPK,1:POKEPK+1,16
15 POKEPK+2,74:POKEPK+3,205:POKEPK+4,162:POKEPK+5,5:POKEPK+6,201
20 POKEPK+13,205:POKEPK+14,231:POKEPK+15,7:POKEPK+16,50:POKEPK+17,21
25 POKEPK+18,74:POKEPK+19,201:POKE24650,22:POKE24651,74:GK=18965
30 DATA0,0,1,1,1,0,0,0,0,1,1,1,0,0,1,1,1,1,1,1,1,1,1,2,1,1,1
35 DATA1,1,1,1,1,1,1,0,0,1,1,1,0,0,0,0,1,1,1,0,0,0
40 DIMA(8,8)
50 RESTORE:FORY=1TO7:FORX=1TO7:READA(X,Y):NEXT:NEXT
55 PRINTCHR$(8)
60 CH$=CHR$(1)
70 GOSUB1000:AX=4:AY=4
80 GOSUB2000:CF=0:GOTO80
1000 CLS:COLOR0,1,3,7
1010 XL=112:YL=75:XA=0:CO=2:YA=76:GOSUB60000
1030 CY=1:FORY=60TO24STEP-6:CX=1:FORX=38TO76STEP6
1040 IFA(CX,CY)=0THENOUTPUTCHR$(1),X,Y,2
1050 IFA(CX,CY)=1THENOUTPUTCHR$(1),X,Y,1
1060 IFA(CX,CY)=2THENOUTPUTCHR$(1),X,Y,0
1070 CX=CX+1:NEXT:CY=CY+1:NEXT
1094 RETURN
1096 XL=112:YL=12:CO=0:XA=0:YA=15:GOSUB60000
1097 PRINTCHR$(8)
1099 RETURN
2000 OUTPUT"X",32+(AX*6),66-(AY*6),3
2010 AJ=AX:AK=AY:A=A(AX,AY)
2020 IFJOY(0)=1ANDA(AX-1,AY)>0THENAX=AX-1
2040 IFJOY(0)=2ANDA(AX+1,AY)>0THENAX=AX+1
2060 IFJOY(0)=4ANDA(AX,AY-1)>0THENAY=AY-1
2080 IFJOY(0)=8ANDA(AX,AY+1)>0THENAY=AY+1
2100 IFA=2THENA=0
2110 IFPDT(0)>100GOTO5000
2120 TONE10,20
2200 OUTPUTCH$,32+(AJ*6),66-(AK*6),A
2210 IFCF<>0THENRETURN
2240 IFFIRE(0)=0THENGOSUB2300
2250 IFFIRE(1)=0THENGOSUB2400
2299 RETURN
2300 FX=AX:FY=AY:CF=1:SOUND3,16:FORT=1TO250:NEXT:SOUND7,4096
2310 IFA(AX,AY)=2GOTO3100
2340 GOSUB2000:IFFIRE(0)=1GOTO2340
2350 IFA(AX,AY)<>2GOTO3100
2355 GOSUB4000:IFFG=1GOTO3100
2360 CX=AX:CY=AY:GOSUB2600:IFOK=1GOTO3100
2370 FORX=1TO5:TONE10,10:FORT=1TO50:NEXT:NEXT
2380 A(AX,AY)=1:A(FX,FY)=2:A(PX,PY)=2
2390 GOSUB1030:RETURN
```

```
2600 OK=0:IFAX=FXANDAY=FY+2THENPX=AX:PY=AY-1:RETURN
2610 IFAX=FXANDAY=FY-2THENPX=AX:PY=AY+1:RETURN
2620 IFAY=FYANDAX=FX+2THENPY=AY:PX=AX-1:RETURN
2630 IFAY=FYANDAX=FX-2THENPY=AY:PX=AX+1:RETURN
2640 OK=1:RETURN
3100 OUTPUT"WRONG MOVE",27,12,0:FORT=1TO100:TONET,3:NEXT
3110 OUTPUT"WRONG MOVE",27,12,2:RETURN
3200 XL=112:XA=0:CO=2:YL=15:YA=18:GOSUB60000:RETURN
4000 QX=(AX+FX)/2:QY=(AY+FY)/2
4010 IFA(QX,QY)=1THENFG=0:RETURN
4020 FG=1:RETURN
5000 OUTPUT"PRESS FIRE BUTTONTO END PLAY",6,18,0
5010 IFPOT(0)<100GOTO5500
5020 IFFIRE(0)=0GOTO6000
5030 TONE250,50:GOTO5000
5500 GOSUB3200:GOTO2200
6000 CT=0:FORX=1TO7:FORY=1TO7
6010 IFA(X,Y)=1THENCT=CT+1
6020 NEXT:NEXT:GOSUB3200
6030 IFCT>5THENO$="YOU NEED PRACTICE"
6040 IFCT=5THENO$="       GOOD"
6050 IFCT=4THENO$="      BETTER"
6060 IFCT=3THENO$="  REALLY CLEVER"
6070 IFCT=2THENO$="    A SHARPIE"
6080 IFCT=1ANDA(4,4)=2THENO$="TAKE A DEEP BOW"
6090 IFCT=1ANDA(4,4)=1THENO$="PERFECT SCORE    YOUR A GENIUS"
6100 OUTPUTO$,6,18,0
6110 FORT=1TO500:SOUND3,332:NEXT:SOUND7,4096
6120 OUTPUT"PLAY AGAIN Y/N",12,70,0:I$=INSTR$(1)
6130 IFI$="Y"GOTO50
6140 CLS
59999 END
60000 POKE18960,YL:POKE18961,XL:POKE18962,CO:YA=77-YA:POKE18963,YA
60010 POKE18964,XA:US=USR(0):RETURN
OK
```

```
    X X X
    X X X
X X X X X X X
X X X O X X X
X X X X X X X
    X X X
    X X X
```

THE HI-Q GAME BOARD

TIC TAC TOE

By George A. Leggett, 20562 Woodward, Mt. Clemens, MI 48043

What can you say about an old standby? This is TIC TAC TOE. I did it on the computer just to demonstrate the Fast Graphics routine and Get Key command discussed in a feature article of this issue. I have also made use of a new technique that I have found to be quite useful in my VIC-20 work. This is the ability to select the joy stick or the keyboard. I myself find it easier and more enjoyable to control the curser via the keyboard. However, I found a great demand for joy sitck control of game pieces in selling programs for the Interact commercially. Now, a happy medium again, the ability to select your preference.

This is a game for two players. I'm sure everyone knows how to play TIC TAC TOE. Three in a row, either up, down, or diagonally wins. When someone wins, the computer will tell you.

There is a nice way, I thought, of asking you to enter your name at the beginning of the program. Only 8 letters can be used for your name. Also, I take full benefit of several Pokes to change the tone of the keyboard. I have also used a Poke that to my knowledge is not included in any of the books or documentation on Interact so I share it with you here. POKE 29462, Y where Y is the Y Axis. What this does is output the normal PRINT statement at any Y Axis you wish and begins scrolling down the screen. If you want to output a lot of print real fast, rather than using OUTPUT this is another way to do it.

If you elect to use the keyboard to play the game, you will use the Get Key routine. I first came across this command in VIC-20 CP/M BASIC. Where our INSTR$ waits for a key to be presses this Get Key command will scan the keyboard while your program or game is running, much like our Control C or Control S functions. One thing you may notice is that you may find yourself pressing a key several times. This is because the computer may be in a part of its program where it is not scanning the keyboard at the particular time you press a key to move the curser. So, there's nothing wrong with your keyboard. I'm using the new keyboard from Micro Video and of course it works great. It is because of the time it takes BASIC to finish all of its work and scan the keyboard.

The TIC TAC TOE game also keeps score how many wins each person has. So, I hope you get a lot of 3-in-a-rows and Away you go!!!

TIC TAC TOE CHALLENGE

Can anyone out there adapt TIC TAC TOE so that one person plays against the computer? It can be done, however, I just do not have time to devote to the project. We shall all look forward to seeing TIC TAC TOE for one in an upcoming issue of INTERACTION INTERNATIONAL.

```
1 REM TIC TAC TOE  NAME"TTT"
2 REM GEORGE A. LEGGETT 20562 WOODWARD MT. CLEMENS, MICH. 48043
10 POKE19215,25:POKE19473,9:POKE19474,74:PK=18953:POKEPK,1:POKEPK+1,16
15 POKEPK+2,74:POKEPK+3,205:POKEPK+4,162:POKEPK+5,5:POKEPK+6,201
20 POKEPK+13,205:POKEPK+14,231:POKEPK+15,7:POKEPK+16,50:POKEPK+17,21
25 POKEPK+18,74:POKEPK+19,201
30 POKE24650,22:POKE24651,74
40 W$="WINNER":LN$=" ":RN$=" ":SO$="SCORE"
100 CLS:COLOR0,7,2,1:POKE19462,6:PRINT"LEFT PLAYER ENTERYOUR NAME."
105 Y=47:POKE24624,3
110 FORX=106TO6STEP-5:OUTPUT"<",X,Y,1:TONEX,50:OUTPUT"<",X,Y,0:NEXT
120 OUTPUT"<",6,Y,1:PRINT:PRINT"    ";:INPUTLN$
125 COLOR0,7,1,2
130 CLS:POKE19462,6:PRINT"RIGHT PLAYER      ENTER YOUR NAME."
140 FORX=6TO112STEP6:OUTPUT">",X,Y,1:TONEX,50:OUTPUT">",X,Y,0:NEXT
150 OUTPUT">",6,Y,1:PRINT:PRINT"    ";:INPUTRN$
160 CLS:COLOR0,4,6,7:POKE24624,4:POKE19462,6:PRINT"DO YOU WANT TO"
170 PRINT"USE THE KEYBOARD OR JOYSTICKS?"
180 OUTPUT"J = JOYSTICKS",6,Y,1:OUTPUT"K = KEYBOARD",6,Y-6,2:I$=INSTR$(1)
190 JK=ASC(I$):IFI$<>"J"ANDI$<>"K"GOTO180
200 CLS:COLOR0,1,2,3:POKE19462,6:IFI$="K"GOTO300
210 PRINT"DO YOU WANT TO   USE BOTH JOY      STICKS IN THE    GAME?"
220 OUTPUT"Y=YES N=NO",24,18,2:I$=INSTR$(1):IFI$<>"Y"ANDI$<>"N"GOTO220
230 YN=ASC(I$):GOTO350
300 PRINT"R KEY = RIGHT    L KEY = LEFT      U KEY = UP        D KEY = DOWN"
310 PRINT"CR KEY= ENTER":OUTPUT"PRESS ANY KEY TO START GAME",6,18,1
320 I$=INSTR$(1)
350 L=LEN(LN$):L2=LEN(RN$):IFL>8THENLN$=LEFT$(LN$,8)
355 X=100*RND(1):GF=0:IFX>50THENGF=1
360 IFL2>8THENRN$=LEFT$(RN$,8)
370 CLS:COLOR3,1,2,4:POKE24624,0:OUTPUTLN$,6,70,1:OUTPUTRN$,60,70,2
375 TX=41:TY=42:TC=1:POKE18965,0
380 OUTPUTSO$,6,64,1:OUTPUTSO$,60,64,2:XL=56:YL=8:XA=0:YA=57:CO=1
385 OUTPUTLN,36,64,1:OUTPUTRN,90,64,2
390 GOSUB60000:XA=56:CO=2:YA=57:GOSUB60000
400 XL=112:YL=45:CO=0:XA=0:YA=49:GOSUB60000
410 XL=1:YL=36:CO=3:XA=50:YA=45:GOSUB60000
420 XA=62:YA=45:GOSUB60000:XL=36:YL=1:XA=38:YA=33:GOSUB60000:XA=38:YA=21
430 GOSUB60000:FORX=1TO9:A(X)=0:NEXT
435 TX=41:TY=42
440 IFGF=0THENXS=27:C=1:SS$="O":GOSUB3000:GOTO500
450 IFGF=1THENXS=83:C=2:SS$="X":GOSUB3000
500 OUTPUT"O",27,55,1:OUTPUT"X",83,55,2
510 CX=0
520 XL=36:YL=36:CO=3:XA=38:YA=45:CX=CX+1
530 IF(A(1)ANDA(2)ANDA(3))=CXGOTO700
540 IF(A(4)ANDA(5)ANDA(6))=CXGOTO710
550 IF(A(7)ANDA(8)ANDA(9))=CXGOTO720
560 IF(A(1)ANDA(4)ANDA(7))=CXGOTO730
570 IF(A(2)ANDA(5)ANDA(8))=CXGOTO740
580 IF(A(3)ANDA(6)ANDA(9))=CXGOTO750
590 IF(A(3)ANDA(5)ANDA(7))=CXGOTO760
600 IF(A(1)ANDA(5)ANDA(9))=CXGOTO770
610 IFCX<2GOTO520

615 X=1
```

```
620 IFA(X)=0GOTO440
630 X=X+1:IFX<10GOTO620
640 XL=112:YL=45:CO=0:XA=0:YA=45:GOSUB60000:WINDOW42
650 FORX=1TO25:XL=0:YL=0:CO=X:XA=0:YA=0:GOSUB60000:PRINT" TIE ";
660 TONEX*10,20:NEXT:GOTO830
670 XL=112:YL=45:CO=0:XA=0:YA=45:GOSUB60000:PRINTCHR$(8)
700 YL=1:YA=40:GOSUB60000:GOTO800
710 YL=1:YA=28:GOSUB60000:GOTO800
720 YL=1:YA=16:GOSUB60000:GOTO800
730 XL=1:XA=43:GOSUB60000:GOTO800
740 XL=1:XA=55:GOSUB60000:GOTO800
750 XL=1:XA=67:GOSUB60000:GOTO800
760 XA=XA+34:FORX=1TO36:PLOTXA,YA,3:XA=XA-1:YA=YA-1:NEXT:GOTO800
770 FORX=1TO36:PLOTXA,YA,3:XA=XA+1:YA=YA-1:NEXT:GOTO800
800 XL=56:YL=30:CO=0:YA=77:IFCX=2GOTO900
810 XA=56:GOSUB60000:OUTPUTW$,6,55,3:LN=LN+1
815 OUTPUTLN-1,36,64,0:OUTPUTLN,36,64,1
820 SOUND0,332:FORX=0TO255:COLOR3,1,2,X:TONE256-X,20:NEXT
830 XL=112:YL=8:CO=3:XA=0:YA=12:GOSUB60000
835 OUTPUT"SAME GAME (Y/N)?",6,11,0:COLOR3,1,2,4
840 I$=INSTR$(1):IFI$<>"Y"ANDI$<>"N"GOTO840
850 IFI$="Y"THENCOLOR3,1,2,4:GOTO350
860 XL=112:YL=77:CO=0:XA=0:YA=77:GOSUB60000:POKE19462,20
870 PRINTCHR$(8):COLOR6,4,3,0:PRINT"DO YOU WANT TO  PLAY A NEW GAME (
Y/N)?"
880 I$=INSTR$(1):IFI$<>"Y"ANDI$<>"N"GOTO880
890 IFI$="Y"GOTO100
899 WINDOW77:END
900 XA=0:GOSUB60000:OUTPUTW$,62,55,3:RN=RN+1
910 OUTPUTRN-1,90,64,0:OUTPUTRN,90,64,2:GOTO820
3000 GOSUB4000:OUTPUTTS$,TX,TY,TS:OUTPUTSS$,XS,55,0
3010 FORX=1TO30:P=PEEK(18965):OUTPUT"-",TX,TY-2,3:NEXT
3015 OUTPUT"-",TX,TY-2,0:TONE20,20
3017 IFJK=74THENGOSUB3200
3020 IFP=117ANDTC>3THENTY=TY+12:TC=TC-3
3030 IFP=100ANDTC<7THENTY=TY-12:TC=TC+3
3040 IFP=108ANDTC<>1ANDTC<>4ANDTC<>7THENTC=TC-1:TX=TX-12
3050 IFP=114ANDTC<>3ANDTC<>6ANDTC<>9THENTC=TC+1:TX=TX+12
3060 IFP=13ANDA(TC)<>0THENSOUND1,20:FORX=1TO50:NEXT:SOUND1,21:GOTO3000
3070 IFP=13ANDA(TC)=0THENA(TC)=C:GOSUB4000:OUTPUTTS$,TX,TY,TS:GOSUB3100
3090 POKE18965,0:RETURN
3100 IFGF=0THENGF=1:RETURN
3110 IFGF=1THENGF=0:RETURN
3200 IFYN=78THENJ=0
3210 IFYN=89THENJ=C-1
3230 IFJOY(J)=1THENP=108:RETURN
3240 IFJOY(J)=2THENP=114:RETURN
3250 IFJOY(J)=4THENP=117:RETURN
3260 IFJOY(J)=8THENP=100:RETURN
3270 IFFIRE(J)=0THENP=13:RETURN
3280 P=0:RETURN
4000 IFA(TC)=1THENTS$="O":TS=1:RETURN
4010 IFA(TC)=2THENTS$="X":TS=2:RETURN
4020 TS$=" ":RETURN
60000 POKE18960,YL:POKE18961,XL:POKE18962,CO:YA=77-YA:POKE18963,YA
60010 POKE18964,XA:US=USR(0):RETURN
```

THE UTILITIES

By George A. Leggett, 20562 Woodward, Mt. Clemens, MI 48043

This is the first contest offered by INTERACTION INTERNATIONAL. It started out as a program and ended up as a nightmare. Oh, not the program. It was easy. It's in solving the problem. First off, let me say before I explain the game, that I am offering a $10.00 cash prize to the first person who can successfully solve the puzzle. The offer is open to anyone; Interactor or non-Interactor.

The game is called UTILITIES. When you type RUN it will ask you to turn the left control Pot all the way down so there is no line or almost no line. Then press the Fire Button. Like the other BASIC programs being featured in this issue, the game was designed to demonstrate the use of my Fast Graphics subroutine. You will see three houses and three boxes labeled G, E, and W or Gas, Electric and Water. You must connect all three houses with Gas, Electric and Water and never cross a line. The computer will not let you cross a line with some rare exceptions which I will explain later. You cannot legally cross. The way to connect lines is simple: Take the Joy Stick which can move up, down, left or right go to a utility (surrounded by green, yellow in the middle) hit the Fire Button to make contact, the dot will start blinking, and then move the Joy Stick anywhere you wish. Go up, down, around the screen anywhere you wish, and connect to the green part of any house. After you have made a connection, you will see two cursers flashing; one to the left and one to the right of the line. You may select which side of the line you want to be on and you simply push the Joy Stick right or left. Now your dot is again blinking and you may again join up with any utility you wish. After you can no longer go or simply give up, turn the Pot all the way right and use the Fire Button to give up. If you do not want to give up, turn the Pot back down to continue. Please note that if your lines are connecting at the top of the houses, they will be erased by the message being printed on the screen. When you hit the Fire Button, your score will appear with an option to play again. The most that I can get or any of my friends who have tried has been 8. Scoring 8 is always possible and is always easy. But 9 is the goal... I haven't seen it done lately. I first saw this problem in high school and have just plain forgotten how to do it. After a month I'm not too sure it can be done!

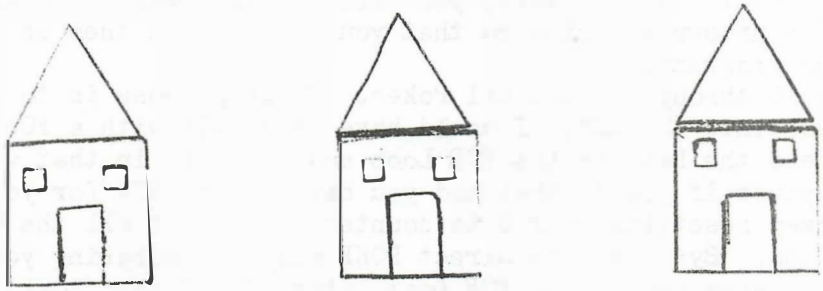
There must be three lines leaving each utility--you cannot have have one line leave Gas, for example, and branch off to all three houses.

If you ever do get 9, you will run across lines 3190 and 3195 and 3199 which determine and display the winner. Well, if you win, I would like you to write any routines, any graphics that you would like to announce yourself as the winner, and of course I will be happy to print your routine along with the answer in an upcoming issue. Of course, submit the solution on paper. It is your option to submit the routine on paper or tape. This is a challenge for everyone, and I'm sorry to say, as of this writing, I don't have the answer and my Dad and all of his friends at work are driving me nuts! So, I hope someone out there can end my nightmares--I give up!!!

```
1 REM THE UTILITIES GAME  NAME"UTILS"
2 REM GEORGE A. LEGGETT 20562 WOODWARD MT. CLEMENS, MICH 48043
3 REM DECEMBER 9,1982
10 POKE19215,25:POKE19473,9:POKE19474,74:PK=18953:POKEPK,1:POKEPK+1,16
15 POKEPK+2,74:POKEPK+3,205:POKEPK+4,162:POKEPK+5,5:POKEPK+6,201
20 POKEPK+13,205:POKEPK+14,231:POKEPK+15,7:POKEPK+16,50:POKEPK+17,21
25 POKEPK+18,74:POKEPK+19,201:POKE24650,22:POKE24651,74:GK=18965
30 DIMA(3,3)
40 FORX=1TO3:FORY=1TO3:A(X,Y)=0:NEXT:NEXT
50 CLS:COLOR7,5,2,0:OUTPUT"PLUG IN THE LEFT CONTROLLER AND",6,70,2
60 OUTPUT"TURN THE POT LEFTUNTIL THE LINE ISGONE THEN PRESS  FIRE",6,58
,2
70 XA=0:YL=1:CO=3:YA=20:XL=POT(0):IFXL>112THENXL=112
80 GOSUB60000:IFFIRE(0)=1THENCO=0:GOSUB60000:GOTO70
100 CLS:COLOR0,1,2,3:XA=12:YA=45:GOSUB1000:YA=45:XA=48:GOSUB1000:XA=84
110 YA=45:GOSUB1000
120 OUTPUT"G",19,20,0:OUTPUT"E",55,20,0:OUTPUT"W",91,20,0
150 XP=58:YP=32
160 GOTO3000
999 END
1000 XL=20:YL=11:CO=2:GOSUB60000:XL=20:YL=10:CO=1:GOSUB60000
1002 GOSUB60000:YL=15:CO=2:YA=YA-20:GOSUB60000:YL=13
1005 XL=18:XA=XA+1:YA=YA-1:CO=3:GOSUB60000:XL=20:YL=10:XA=XA-1:YA=YA+21
1020 CO=1:XL=XL-2:YL=1:XA=XA+1:GOSUB60000:YA=YA+1:IFXL>2GOTO1020
1100 YL=3:XL=3:CO=0:XA=XA-5:YA=YA-11:GOSUB60000:XA=XA+10:GOSUB60000
1110 XA=XA-5:YA=YA-3
1120 YL=5
1130 GOSUB60000:RETURN
2000 IFJOY(0)=1ANDXP>=0THENXP=XP-1
2010 IFJOY(0)=2ANDXP=<112THENXP=XP+1
2020 IFJOY(0)=4ANDYP<=77THENYP=YP+1
2030 IFJOY(0)=8ANDY>=0THENYP=YP-1
2040 IFFIRE(0)=0ANDPOINT(X,Y-1)=2THENCF=2:FI=1
2099 RETURN
3000 X=XP:Y=YP:GOSUB2000:PLOTX,Y,CF:PT=POINT(XP,YP)
3010 IFPOT(0)>100GOTO3300
3020 IFPT=1ORPT=2THENXP=X:YP=Y
3030 PLOTXP,YP,2
3040 IFFI=1THENFI=2:GOSUB3200:UH=U
3050 U=0:GOSUB3200
3060 IFY=34ANDPOINT(X,Y+1)=2ANDU>0GOTO3100
3099 GOTO3000
3100 X1=X:Y1=Y
3105 PLOTX-1,Y,3:PLOTX+1,Y,3
3110 PLOTX-1,Y,0:PLOTX+1,Y,0
3120 IFJOY(0)=1THENXP=XP-1:YP=YP-1
3130 IFJOY(0)=2THENXP=XP+1:YP=YP-1
3140 IFJOY(0)<>1ANDJOY(0)<>2GOTO3100
```

```
3150 CF=0:GOSUB3200:A(UH,U)=1
3160 XC=1:YC=1
3170 IFA(XC,YC)=0GOTO3099
3180 XC=XC+1:IFXC=4THENXC=1:YC=YC+1
3185 IFYC<4GOTO3170
3190 REM THE WINNER
3195 PRINT"THE WINNER"
3199 STOP
3200 IFX=>12ANDX<=31THENU=1
3210 IFX>=48ANDX<=67THENU=2
3220 IFX>=84ANDX<=103THENU=3
3230 RETURN
3300 OUTPUT"HIT FIRE BUTTON  TO GIVE UP",6,70,3
3310 IFPOT(0)<100THENXL=112:YL=24:CO=0:XA=0:YA=76:GOSUB60000:GOTO3020
3320 IFFIRE(0)=1GOTO3010
3330 XL=112:XA=0:CO=0:YA=76:YL=24:GOSUB60000
3340 A=0:FORXC=1TO3:FORYC=1TO3:A=A+A(XC,YC):NEXT:NEXT
3350 OUTPUT"YOU HAVE JOINED",6,72,3:OUTPUTA,6,66,3
3360 OUTPUT"UTILITIES",24,66,3:OUTPUT"WITH HOUSES",6,60,3
3370 CO=3:YL=8:YA=9:GOSUB60000:OUTPUT"PLAY AGAIN Y/N?",12,8,0
3380 I$=INSTR$(1):IFI$="N"THENCLS:END
3390 IFI$<>"Y"GOTO3380
3400 GOTO40
59999 END
60000 POKE18960,YL:POKE18961,XL:POKE18962,CO:POKE18963,77-YA
60010 POKE18964,XA:US=USR(0):RETURN
OK
```

THE UTILITIES



MACHINE SHOP TALK

THE HAPPY MARRIAGE

With a title like this you will assume that I am writing to discuss my love life. Hardly the case. The marriage I'm talking about is between BASIC and Machine Language. There can be a happy medium. Up to now, when we have talked in our Machine Shop, I have told you to use your Monitor to enter your Machine code and save it on tape. Then I have said to load in BASIC and load in your program. Well, in this issue we are going to do things a bit differently. Now, I don't want to confuse you. "now that I've got this way down pat, he's changing the rules on me!" That will still apply to other articles. But for now, what I am trying to do is to get the rest of you involved in Machine Language through BASIC. I spent a great deal of time this past year in working with the VIC-20 computer, where the marriage of BASIC and Machine Language is a must. In fact, the manual that comes with the computer gives countless examples of Pokes and Peeks to use in your BASIC programs. So I have become more acquainted with BASIC and and have found that one may have the best of both worlds by combining them. The Interact is no exception. It has all the capabilities to do the same job.

Our first problem is that we cannot PEEK and POKE all of the memory by using Level II or Fast Graphics BASIC. There are certain parameters restricting which areas you can examine. It is true that you have read in a past article in INTERACTIONS that you can PEEK the ROM. But how do you PEEK or for that matter POKE higher addresses in BASIC and for those of you with 32K how do you PEEK and POKE above 32,767? We're going to answer those questions now.

I have divided the BASIC program which follows in a very plain and simple manner just to illustrate how the use of a small Machine Language program with a USR can PEEK and POKE any memory location that the 8080 addresses. The purpose of this program is to give you a better understanding of how to combine a USR with a BASIC program. I must remind you that I have used this format and th line numbers as a demonstration. In your work, you will probably want to assign numbers of your own choosing so that you may call on them as a routine in your own programming.

Lines 10 through 40 are all Pokes. Their purpose is to load the Machine Code into the RAM. I could have used DATA with a FOR Loop and then Read the Data in the FOR Loop and Poked it in that way. I did not because if you do that and you have other DATA for your program you must keep resetting your Data counter to go past all the Machine Language data. By using the Direct POKE and just entering your data, you can save time and a long FOR Loop later. The direct POKE does not take too much more memory and is a fast way of entering data and leaves you free from worry about incrementing your Data Counter pointer if you need other data in your program.

The Machine code sits from 4A80 to 4A86. Note that this memory is used by BASIC for the CSAVE* and CLOAD* functions. (Refer to "The Innards of BASIC" by Harry Holloway, INTERACTIONS Vol. 3 no. 1 p2) Thus, your machine code will be destroyed if you use either of those functions. If you do need to use CSAVE* or CLOAD* in your program, simply execute Lines 10, 20, 30 and 40 before running the USR. This group of lines can be called as a subroutine to initialize the USR and you will never have the fear of it not being there.

Lines 50 and 60 select whether you want to PEEK or POKE. Lines 100 through 199 are for Peeking. Enter any Decimal address and it will PEEK it for you. Lines 200 through 260 are for Poking.

I am also including the listing of the Machine Code for the USR routine. This is to show you how the program would look if you entered it in Machine Language. The listing gives all of the mnemonics and a Decimal listing so you can compare the Pokes in the opening lines with what they mean in Hex and Machine Code. This is only to clarify things for you so you can see, for example, that in address 19,079 is a 201. What's a 201? Well, in Hexadecimal it's a C9 or RETURN. I have given the listing to help you in your study of Machine Language.

Because of the Machine Language Code, this program will PEEK and POKE anywhere in the Machine and it does not matter what address you want to look at.

And now, I would like to share another great discovery. At least it was a discovery to me. After completing work on this Machine Shop Talk, I had a very enjoyable and useful conversation with Mr. Dave Ross of Micro Video. I want to thank him for his cooperation and graciousness. After telling him about the above article, he suggested to me that I use negative Poking and Peeking. Well, the thought never occurred to me until then although I do this and use it in the EDUBASIC Overlay which is a great piece of work by by R. P. Williams. It never occurred to me, I don't know why, to try it with LEVEL II BASIC. I did as Dave said, give it a go and it worked great.

For those of you with 32K machines, your BASIC will allow you to look at addresses within the 32,767 range. But what if you want to look at addresses above 32,767 which is 7FFF Hex? You use a negative number in this way: The number needed for the correctly Poked or Peeked address = your desired number minus 65,536. Let's assume the address you wish to look at is 42,000 Decimal. Thus, PRINT 42000-65536 Our answer is -23576. This is the number you would POKE or PEEK to get the equivalent of of saying POKE 42000. Here is why. The line numbers, Pokes and Peeks are in integer BASIC, meaning there's no such thing as POKE 4.82, 3.2. All numbers are whole numbers and furthermore, the addresses are all less than 32,767. (assuming BASIC would allow it.) In binary, 32,767 2 to the 15th power. Or, as we say, 15 Bits. Well, 8 bits and 8 bits are 16 bits. Where did we lose 1 bit? The machine and the software use the 16th bit as a sign bit. 0 is Positive and 1 is Negative. Youre actually turning on the 16th bit. Those of us who have used this with Edubasic are already familiar with use of Negative numbers for addresses above 32,767.

So, you ask, why the program now that we all know about the use of positive and negative addresses for Poking and Peeking? As I have mentioned before, BASIC does put restrictions on what you can and cannot do. In my program, via the use of the USR, theese restrictions don't matter. There are many possibilities to explore. Here are some of my uses, and you may wish to experiment around and come up with some of your own.

I had no way of receiving data through my RS232 in LEVEL II BASIC. Or, for that matter, RS232 BASIC. With this program, you can use LEVEL II or Fast Graphics BASIC for RS232. Normally, for the Slagh port, you simply put in the overlay tape and for the Micro Video port, you would use RS232 BASIC or 32K BASIC. However, neither port gives a command to receive data. This may not seem like much to you, but after working with the VIC-20 there are such commands. So, upon returning to the Interact, I wanted to receive information without using a Communicator Overlay or something of that nature because I still wanted the power of BASIC. Now, if I lit up some of your eyes, great! You can begin to see the enormous potential we are up against. We now have a way to receive as well as transmit data. That is one of the primary reasons why I developed this program.

Other uses could be for a Monitor to work with your BASIC. Or, just have fun Poking and Peeking around. You will have to decide on that, but I hope I have laid the groundwork for what could be some very interesting software in the future. Thanks for the chat, and we'll get together next time for another Machine Shop Talk.

Sincerely, George A. Leggett, 20562 Woodward, Mt. Clemens, MI 48043

```
1 REM HAPPY MARRIAGE  NAME "HAPMY"
2 REM GEORGE A. LEGGETT 20562 WOODWARD MT. CLEMENS, MICH. 48043
10 POKE19215,25:PK=19072:POKEPK,33:POKEPK+1,0:POKEPK+2,0:POKEPK+3,126
20 POKEPK+4,50:POKEPK+5,136:POKEPK+6,74:POKEPK+7,201
30 POKEPK+9,33:POKEPK+10,0:POKEPK+11,0:POKEPK+12,62:POKEPK+13,0
40 POKEPK+14,119:POKEPK+15,201
50 CLS:PRINT"P = PEEK":PRINT"K = POKE":I$=INSTR$(1):IFI$<>"P"ANDI$<>"K"
GOTO50
60 IFI$="K"GOTO200
100 CLS:INPUT"ENTER DECIMAL    ADDRESS OF PLACE TO PEEK";EP
110 H=INT(EP/256):L=EP-(256*H)
120 POKE19073,L:POKE19074,H
130 POKE19473,128:POKE19474,74:U=USR(0)
140 PRINTPEEK(19080):PRINT"PRESS S KEY TO   STOP OR ANY KEY  TO GO ON."
150 I$=INSTR$(1):IFI$<>"S"GOTO100
199 END
200 CLS:INPUT"ENTER DECIMAL    ADDRESS OF PLACE TO POKE";EP
210 H=INT(EP/256):L=EP-(256*H)
220 POKE19082,L:POKE19083,H
230 INPUT"ENTER DECIMAL    NUMBER TO BE    POKED";P
240 POKE19085,P:POKE19473,137:U=USR(0)
250 PRINT"PRESS S KEY TO   STOP OR ANY KEY  TO GO ON":I$=INSTR$(1)
260 IFI$<>"S"GOTO200
OK
```

```
HEXADECIMAL LISTING
4A80 21 LXI H
4A81 L
4A82 H
4A83 7E MOV A,M
4A84 32 STA
4A85 88
4A86 4A
4A87 C9 RET
4A88 PV
4A89 21 LXI H
4A8A  L
4A8B  H
4A8C 3E MVI
4A8D  P
4A8E 77 MOV M,A
4A8F C9 RET
```

```
DECIMAL LISTING
 19072  33
 19073  00
 19074  00
 19075  126
 19076   50
 19077  136
 19078   74
 19079  201
 19080   PV   PEEK VALUE
 19081   33
 19082    0
 19083    0
 19084  162
 19085    0   POKE VALUE
 19086  119
 19087  201
```

INTERACTORS INPUT

By Lora A. Leggett, 20562 Woodward, Mt. Clemens, MI 48043

This section is devoted to your questions, thoughts, information about yourselves and your computers. Other Interactors will be interested to know that they are not alone in their desire to know more about the Interact. Also, behind every Interact keyboard is a different person with a different interest, application and lefestyle all his/her own. By the way, are there any "Interactresses" around?

EXPANDING INTERACT'S WORLD: Mr. Charles Smith of Friendswood, Texas, posed some very timely questions in his letter of Nov. 10, 1982. He would leke to see some basic interfacing circuits for photo cells, strain guages, thermistors, and use of the Interact as a Volt Meter or Ohm meter. He is interested in any A. to D. applications for the Interact. (Analog to Digital) He tells us that his Pot. will only go up to 175. Mr. Smith has also been investigating the possibility of using a Commodor VIC disc which is available in his area for only $295.00.

The discrepancy of Pot values in different Interacts is the fault of internal resistors in the machines. Our 16K goes to 220 and our 32K goes to 219. A quick test with no joy stick plugged in with BASIC loaded to find the upper limit of your A to D. Converter: PRINT POT(0); POT(1) and press CR key. All of your hardware questions will be dealt with in George's new series, THE HARD FACTS OF LIFE.

When asked about the VIC disc, George replied, "As time goes on and knowledge permits, I someday hope to couple the VIC and the VIC Disc with the Interact. For now, it's a dream. But tomorrow, it will be fact. "

Thank you for writing, Mr. Smith, and do keep the letters coming!

THE HARD FACTS OF LIFE

BY

GEORGE A. LEGGETT 20562 WOODWARD MT. CLEMENS, MICH. 48043

<=><=><=><=><=><=><=><=><=><=><=><=><=><=><=><=><=><=><=><=>

INTRODUCTION

THIS IS THE START OF A BRAND NEW SERIES WHICH I SINCERELY HOPE WILL BENEFIT EVERYONE. THE SERIES DEALS WITH HARDWARE AND THE USE OF YOUR COMPUTER.

......................DO NOT TURN THE PAGE....................

IF YOU HAVE BEEN SCARED TO DEATH BY THE WORD HARDWARE DON'T BE. UNLIKE ANY OTHER HARDWARE ARTICALS YOU HAVE READ I ASSURE YOU THIS WILL BE A WHOLE NEW APPROACH IN DOING AND LEARNIG ABOUT HARDWARE. ALSO WHAT IS MOST IMPORTANT IS I SHALL SUPPLY ALL THE SOFTWARE TO RUN THE HARDWARE.

THERE ARE PROBABLY A MILLION QUESTIONS AT THIS POINT AS TO WHAT I HAVE IN MIND AND WHAT IS GOING ON. LET ME START THIS OFF BY TELLING YOU WHO I AM.

MOST OF YOU KNOW ME BY MY MACHINE SHOP TALK AND PROGRAMS I HAVE WRITTEN IN PAST INTERACTIONS. BESIDES THIS I HAVE WRITTEN FOUR PROGRAMS FOR MICRO VIDEO WHICH ARE IN THEIR CATALOG PLUS PART OF A COMMERCIAL PROGRAM FOR THEM. OKAY YOU SAY SO YOU MUST KNOW YOUR SOFTWARE FAIRLY WELL BUTC WHAT DOES THIS HAVE TO DO WITH HARDWARE?

I BEGAN IN ELECTRONICS WHEN I WAS 13 (I AM NOW 29) BECAUSE I WANTED TO BELIEVE IT OR NOT BUILT MY OWN COMPUTER. I WENT INTO K-MARTS (AT THAT TIME THEY HAD AN ELECTRONIC BOOK SECTIONS) AND ASKED WHAT BOOK I WOULD NEED TO START BUILDING A COMPUTER. WELL IN 1966 THAT WOULD BE LIKE GOING IN A STORE NOW AND ASKING TO BUY A BOOK TO MAKE AN INTERSTELLER SPACE SHIP. NEEDLESS TO SAY I GOT A STRANGE LOOK. THE MAN ASKED WHAT I NEW OF ELECTRONICS I SAID NOTHING, THEN HE ASKED WHAT I NEW ABOUT ELECTRICITY, WELL I HAD HIM THERE! FORTUNATE FOR ME MY FATHER HAD LET ME USE A SET OF HIS BOOKS ON ELECTRICITY FROM AROUND 9 YEARS OF AGE. SO THIS THOUGHT OF LEARNING ELECTRONICS DID NOT BOTHER ME AT ALL. I MEAN BY THAT AGE I KNEW DC CIRCUITS, AC CIRCUITS, DC AND AC MOTORS, DC AND AC GENERATORS, TRANSFORMERS, AND MORE. SO I FIGURED THIS COULDN'T BE MUCH HARDER IF ANY THEN THAT. I HAD A GREAT MATH BACKGROUND BY THEN WITH ALL THE ALGEBRA AND TRIGONOMETRY I WOULD NEED SO ON I WENT. I HAD THE GREATEST SUPPORT FROM MY FATHER WHO NOT ONLY HELPED FINANCE 99% OF MY ELECTRONIC ENDEAVORS BUT ALSO SPENT COUNTLESS HOURS HELPING ME BUILD WHATEVER CRAZY CIRCUIT I CAME UP WITH. THIS HELP WAS EXTREMELY IMPORTANT IN THE BEGINNING SINCE I WAS BORN WITH CEREBRAL PALSY WHICH AFFECTS MY HANDS AND WALKING WITH A SLIGHT SPEECH IMPAIRMENT. OVER MY LIFE I AM GLAD TO SAY IT MY CONDITION ONLY IMPROVES WITH AGE. NOW I CAN SOLDER IC CIRCUITS AND WORK WITH PC BOARDS AND MANY OTHER THINGS I WOULD HAVE NEVER THOUGHT OF 5 YEARS AGO. I STILL HAVE TROUBLE WITH SOMETHINGS BUT MAINLY IT IS JUST FINDING TIME AND WITH TWO KIDS

MY GIRL DEBBIE AGE 2 AND MY SON OF 4 MONTHS. TIME IS ONE THING THAT IS HARD TO FIND THESE DAYS.

FOR THE NEXT 8 YEARS FROM 13 TO 21 I SPENT MY TIME ON ELECTRONICS AND INVENTIONS. TO MAKE BRIEF I CAME UP WITH OVER 90 DIFFERENT ELECTRONIC INVENTIONS AND TOOK SEVERAL TO MANY DIFFERENT COMPANIES. I DON'T KNOW IF YOU CAN FIGHT CITY HALL BUT YOU CERTAINLY CAN NOT FIGHT LAWYERS AND BIG BUSINESS AND AFTER THREE PATENT ATTORNEYS AND 9 YEARS OF TRYING I LEARNED MY LESSON, THE HARDWAY.

SO THIS IS WHY I THINK I AM QUALIFIED TO TRY AND HELP YOU WITH NEW AND DIFFERENT HARDWARE AND YOU CAN TAKE ADVANTAGE OF SIXTEEN YEARS OF CIRCUITS THAT I HAVE KEPT IN LOG BOOKS. OKAY NOW THAT YOU HEARD MY LIFE STORY LET'S GET ON WITH A NEW STORY IN YOUR LIFE

...

WHEN I TALKED ABOUT A NEW APPROACH I MENT IT. IF YOU ARE LIKE ME YOU ARE AFRAID TO GO INTO YOUR INTERACT TO DO MINNOR REPAIRS, NEVER MIND INSTALLING CIRCUITS THAT COULD BLOW UP THE WHOLE THING. SO WHAT OTHER WAY IS THERE TO HAVE HARDWARE AND NOT GO INTO INTERACT? WELL THIS IS WHAT THE HARD FACTS OF LIFE IS ALL ABOUT. EACH ISSUE I HOPE TO SHOW YOU ANOTHER CIRCUIT THAT MAY BE OF HELP TO YOU, WITHOUT REWIRING YOUR INTERACT.

THE NEXT QUESTION IS WILL I HAVE TO SOLDER? YES, BUT IF I CAN ANYONE CAN, RIGHT, RIGHT! IT IS NOT THAT HARD BUT LIKE ANY-THING IN LIFE THE MORE YOU PRACTICE THE BETTER YOU GET. YOU CAN USE WIRE WRAPPING BUT, THIS I HAVE FOUND MAY WORK GREAT WHEN YOU ARE DEALING WITH ALL IC'S (INTERGRATED CIRCUITS) BUT WITH OTHER COMPOENTS IT CAN BE A PROBLEM.

NEXT WHAT TYPE OF CIRCUIT BOARD SHOULD YOU USE. THIS IS ALL UP TO THE CIRCUIT SIZE AND I SHALL TRY AND TELL YOU WHAT I HAVE USED IN THE PARTICULAR CIRCUIT WE ARE BUILDING. FINALLY IF I DON'T BUILD IT AND TEST IT YOU WILL NOT SEE IT.

TONE DECODER

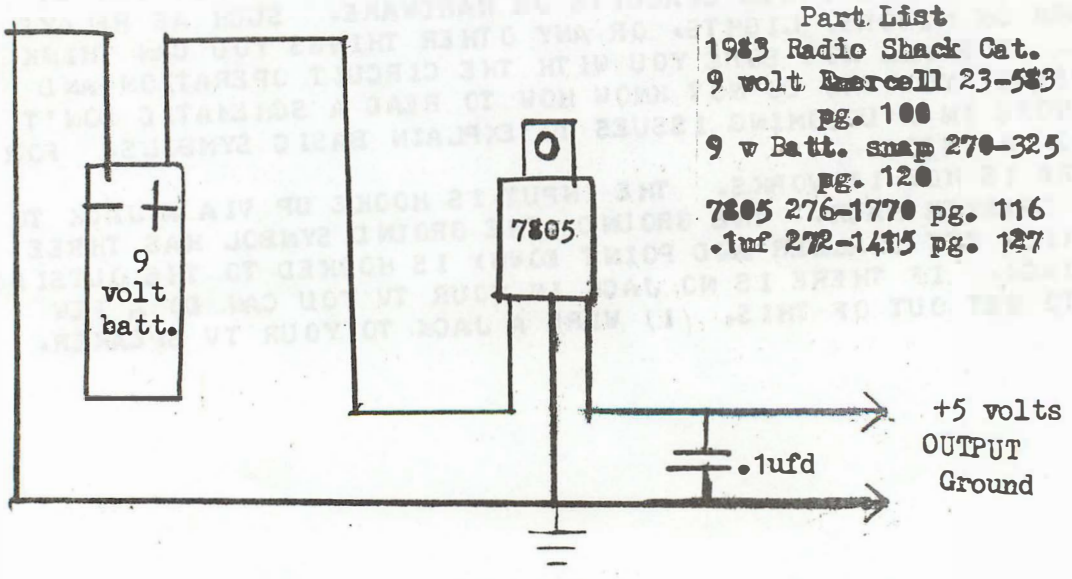
WITH THIS CIRCUIT YOU CAN HAVE A TTL HIGH AND LOW OUTPUT TO USE TO ACTIVATE OTHER CIRCUITS OR HARDWARE. SUCH AS RELAYS THAT TURN ON MOTORS, LIGHTS, OR ANY OTHER THINGS YOU CAN THINK OF AND USE. I SHALL NOT BORE YOU WITH THE CIRCUIT OPERATION AND FOR THOSE OF YOU WHO DO NOT KNOW HOW TO READ A SCHEMATIC DON'T GO AWAY I HOPE IN A UPCOMING ISSUES TO EXPLAIN BASIC SYMBOLS. FOR NOW HANG IN THERE.

HERE IS HOW IT WORKS. THE INPUT IS HOOKE UP VIA A JACK TO YOUR TV SPEAKER JACK. THE GROUND (THE GROUND SYMBOL HAS THREE LINES WHICH GET SMALLER AND POINT DOWN) IS HOOKED TO THE OUTSIDE OF THE JACK. IF THERE IS NO JACK IN YOUR TV YOU CAN DO A FEW THINGS TO GET OUT OF THIS. (1.) WIRE A JACK TO YOUR TV SPEAKER.

MANY ELECTRONIC STORES WILL SELL A DEVICE THAT YOU CAN HOOK UP WITH JUST CLIPS (THIS MEANS NO SOLDERING OR CUTTING WIRES). I SHALL USE RADIO SHACK PARTS LIST ONLY BECAUSE THEY ARE WORLD-WIDE AND EVERYONE HAS ONE NEAR THEM BUT, THERE ARE MANY OTHER LESS EXPENSIVE COMPANIES. (2) YOU COLD USE A TELEPHONE PICKUP BY YOUR TV SPEAKER (PHONE PICKUP COIL 44-533) ANDC THEN RUN IT TO YOUR CASSETTE MIC INPUT AND HOOK THE CIRCUIT TO YOUR CASSETTE RECORDER SPEAKER JACK. (3) IF YOU DO NOT HAVE A CASSETTE RECORDER YOU CAN USE A MINI AMPLIFIER SUCH AS RADIO SHACK'S 277-1008 PAGE 121. YOU CAN USE A MICROPHONE INSTEAD OF A PHONE PICKUP. ANY OF THESE WAYS WILL WORK FIND BUT I LIKE THE DIRECT WAY BEST. THIS ELIMINATES OUTSIDE NOISE.

ONCE YOU HAVE THIS HOOKED UP YOUR ARE READY TO GO ON. I DID NOT SAY THIS WOULD BE EASY I SAID YOU WOULD NOT HAVE TO GO INTO YOUR INTERACT. THE CIRCUIT CAN WORK FROM 4 VOLTS TO 24VOLTS BUT I SUGGEST YOU USE A REGULATED 5 VOLT POWER SUPPLY SO YOU ARE COMPATIBLE WITH OTHER CIRCUITS WE WILL ADD ON LATER. NOW I REALIZE A 5 VOLT POWER SUPPLY MAY NOT BE IN YOUR HOUSE AND YOU MAY NOT WISH TO INVEST. NO PROBLEM LOOK AT THE QUICK WAY YOU CAN MAKE ONE WITH ONE IC A CAPACITOR AND A 9 VOLT BATTERY FOR ABOUT $2 AND IN LESS THEN 30 MINS. IF YOU WISH TO REPLACE THE BATTERY WITH A BATTERY ELIMINATOR YOU WILL NEED MORE CAPACITORS TO FILTER OUT THE AC HUM. IF I RECEIVE ANY LETTERS IN THE INTEREST OF WHAT AC HUM IS ALL ABOUT I WILL DISCUSS IT BUT THIS IS NOT THE TIME.

IC-1 IS A 7805 REGULATOR. AT FIRST THIS MAY LOOK LIKE A TRANSISTOR WITH ONLY THREE WIRES BUT, NEVER THE LESS IT IS A VERY COMPLEX IC. IT PURPOSE IS TO TAKE AN UNREGULATED VOLTAGE ON THE INPUT AND SUPPLY A CONSTANT VOLTAGE OVER A WIDE CURRENT RANGE. CHECK THE SPEC. SHEET THAT COMES WITH THE IC FOR MORE DETAIL. THE PURPOSE OF C1 IS TO FILTER OUT ANY FEEDBACK OSCILLATION PRODUCED BY THE CIRCUIT IT POWERS. I CHOOSE .1UF FOR THIS BUT IT ALL DEPENDS ON WHAT FREQUENCIES THAT ARE GENERATED BY THE CIRCUIT. THE MORE CIRCUITS YOU ADD THE MORE CAPACITORS LIKE C1 YOU WILL NEED. I WILL EXPLAIN MORE OF THIS WHEN THE TIME COMES.



Part List
1983 Radio Shack Cat.
9 volt Duracell 23-583
pg. 100
9 v Batt. snap 270-325
pg. 120
7805 276-1770 pg. 116
.1mf 272-1415 pg. 127

NOW THAT WE HAVE A POWER SUPPLY WE CAN GO ON. I ASSEMBLED THE CIRCUIT ON A SINGLE IC BOARD (RADIO SHACK 276-024 PAGE 123) AND USED AN 16 PIN IC LOW PROFILE SOCKET. THE 567 IC IS ONLY A 8 PIN IC SO THIS ALLOWED ME TO PUT TWO TONE DECODER CIRCUITS ON ONE BOARD. I DRILLED HOLES IN THE PADS BECAUSE IT WAS EASIER FOR ME TO POKE THE COMPOENTS THROUGH THE HOLES AND BEND THE LEADS BEFORE I SOLDERED. THIS HOWEVER IS NOT NECESSARY IF YOU CAN HOLD THE SOLDER GUN, THE COMPOENT, THE SOLDER AND THE IC BOARD ALL AT ONCE.

THERE IS NO EASY WAY TO TELL YOU HOW TO ASSEMBLE THIS BECAUSE EXPERIENCE IS YOUR BEST TOOL. I FIRST INSTALL THE SOCKET FOLLOWED BY THE RESISTORS AND THEN CAPACITORS. IN MY CIRCUIT C4 IS NOT MOUNTED ON THE BOARD BECAUSE IT DEPENDS WHAT YOU ARE GOING TO USE AS AN INPUT. MY CIRCUIT REQUIRES .65 VOLTS TO TRIGGER IT AND C4 EQUALS .1 UFD. YOUR VALUES MAY VARY. YOU MAY BE SOMEWHAT RELIEVED TO KNOW THAT I HAVE BUILT 6 OF THESE CIRCUITS (3 DOUBLE BOARDS) AND THEY ALL WORKED FIRST TIME.

R2 IS A TRIMMER POT AND SHOULD BE THE LAST THING YOU INSERT BECAUSE THE CONSTANT BENDING AND FLIPPING OF THE BOARD COULD BREAK IT. FINALLY INSERT THE IC AND MAKE SURE YOU HAVE IT IN THE RIGHT WAY. NOW YOU ARE READY TO TEST IT OUT. FIRST CHECK ALL YOUR WORK AND MAKE SURE NOTHING IS TOUCHING ANYTHING IT SHOULD NOT BE TOUCHING. SOLDER A WIRE FOR POWER (RED), GROUND (BLACK), INPUT AND OUTPUT.

THE WAY I TESTED IT AND THE WAY YOU MUST TEST IT ARE TWO DIFFERENT THINGS. I USED A SCOPE, VOLT OHM METER, AND FREQUENCY COUNTER. I WILL ASSUME YOU HAVE NONE. I MUST SAY THIS WILL PRESENT ME WITH A BIGGER PROBLEM THEN YOU. I MUST TRY AND EXPLAIN IT AND YOU ONLY HAVE TO DO IT. WELL HERE GOES.

FIRST YOU MUST KNOW WHAT EXACTLY THIS CIRCUIT DOES. WHEN YOU TURN ON POWER THE OUTPUT IS HIGH MEANING NO CURRENT IS FLOWING THROUGH RL (RESISTOR LOAD). RL COULD BE A RELAY COIL OR BETTER YET FOR THIS PURPOSE AND LED AND RESISTOR. THE LED MUST HAVE A DROPPING RESISTOR OR YOU WILL BLOW OUT YOUR LED (USUALLY LED'S ARE 2 TO 3 VOLTS NOT 5 VOLTS LIKE OUR POWER SUPPLY). WITH POWER ON THE LED WILL BE OFF. WHEN THE CORRECT TONE IS INPUTTED THE LED WILL GO ON. THE TONE THAT WILL TRIGGER IT IS ADJUSTED BY R2 (THE POT). SET THIS IN THE MID OF ITS ROTATION. ON MY CIRCUIT THE FINAL OUT COME THAT THE CIRCUIT ACCEPTED WAS 1003 HZ (HZ STANDS FOR HERTZ WHICH MEANS CYCLES PER SECOND) ACCORDING TO MY FREQ. COUNTER. NOW WHAT DOES THIS HAVE TO DO WITH THE INTERACT. WELL THAT'S SIMPLE TO EXPLAIN. THE INTERACT IS A NICE TONE GENERATOR AND SO WHEN YOU PUT A TONE THROUGH THE TV THAT THE CIRCUIT WILL ACCEPT THE LED WILL GO ON AND WHEN YOU TAKE AWAY THE TONE OR CHANGE IT THE LED WILL GO OFF.

WITH THIS KNOWLEDGE WE HAVE POWER TO THE OUTSIDE WORLD THRU OUR INTERACT WITH NO PORTS OR COMPLEX WIRING TO THE INTERACT. WE NOW CAN CONTROL OTHER DEVICES. WHAT IS SO GREAT ABOUT THIS CIRCUIT IS THAT EVEN IF SOMETHING WENT WRONG AND YOU HAPPEN TO OVERLOAD IT OR SHORT OUT SOMETHING ALL YOU LOSE IS A HANDFULL OF PARTS NOT THE INTERACT! AND 99% OF THE TIME YOU WILL AT WROST ONLY HAVE TO REPLACE THE IC. THIS IS WHAT THE HARD FACTS OF LIFE IS ALL ABOUT.

THE PROGRAM THAT I USED TO DETERMAIN THE TONE REQUIRED IS A SIMPLE ONE LINER.

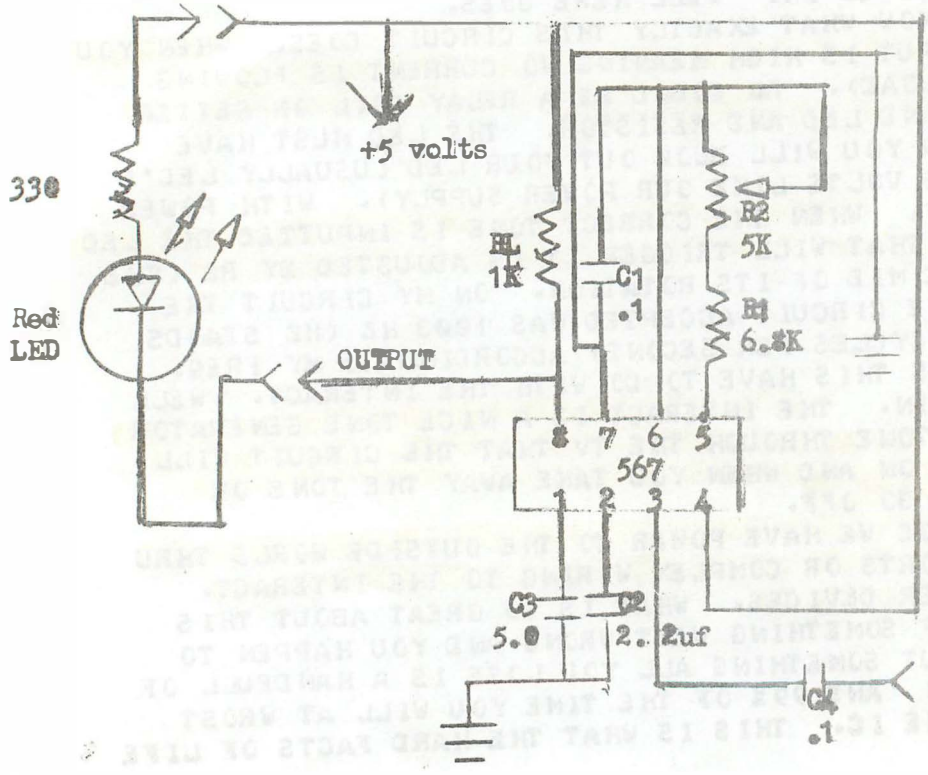
10 INPUT T: TONE T, 65: GO TO 10

THAT ALL IT IS. IN MY CIRCUIT I NEEDED A DURATION OF 65 OR > FOR FOR BEST RESULTS BUT THIS CAN BE CHANGED IF YOU ALTER C1, C2, C3 IF A FASTER TIME IS NEEDED. I FOUND THAT WHEN T =36 TO 40 THE CIRCUIT TURN ON. YOUR NUMBERS MAY VARY. TRY TURNING THE POT TO SET THE DECODER FOR ANOTHER VALUE. BY CHANGING C1 YOU CAN VARY YOUR INPUT FROM A FEW CYCLES TO WELL OVER 500,000 WHICH IS MORE THAN INTERACT OR YOUR TV COULD PRODUCE. IF YOU DO HAVE A SCOPE OR FREQUENCY COUNTER TOUCH THE PROBE TO PIN 5 OF THE IC FOR THE FREQUENCY IT IS TUNED TO. DO NOT HAVE AN INPUT AT THIS TIME OR THE READING WILL BE FALSE.

ON THE FOLLOWING PAGE IS THE SCHEMATIC AND PARTS LIST. IF YOU NEED ANY OF THE PARTS I WILL BE HAPPY TO SEND YOU A COMPLETE KIT TO BUILD ONE TONE DECODER OR TWO SEE PARTS LIST. I AM SORRY THAT I CAN NOT OFFER THE CIRCUIT BUILT I SIMPLY DO NOT HAVE THE TIME.

WHAT CAN I DO WITH IT?

IN COMING ISSUES I SHALL SHOW YOU HOW TO USE THIS CIRCUIT TO DIAL YOUR PHONE AUTOMATICALY, SEND MORSE CODE, TURN ON OR OFF APPLIANCES, RUN OTHER TTL CIRCUITS, CONVERT THE ONE BIT TO 8 BITS AND MUCH, MUCH MORE. AND THIS IS ONLY ONE LITTLE CIRCUIT. YOUR INTERACT IS FAR FROM DEAD IN FACT IT IS JUST BEGINING A NEW LIFE. A NEW LIFE THAT IS THE HARD WAY!



Part List
1983 Radio Shack Catalog

Resistors Page 126
RL 1K 271-023
R1 6.8K 271-032
R2 5K PC Mount Pot 271-217
Capacitors Page 127
C1 .1uf 272-1069
C2 2.2uf 272-1420
C3 4.7uf 272-1422
C4 .1uf 272-1069
External Resistor for LED 330 271-017
LED Red 276-041 pg115
IC LM567 276-1721pg114
IC Board 276-024pg123

Kit Price $10.00
Payable to
G.A.L. Electronics

ADVERTISEMENT POLICY

Those wishing to advertise in INTERACTION INTERNATIONAL and who require five lines or less may do so without payment. All that is required is that I receive a working model or tape. I do this to insure that anyone who buys your product will receive a working product that does exactly what is advertised.

For products requiring additional space, such as full-page advertisement, please contact me to make further arrangements. INTERACTION INTERNATIONAL welcomes all advertisements of products or services as Interactors are a small but devoted computer family.

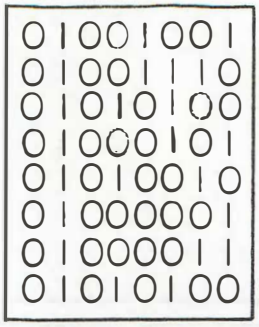
FINAL THOUGHT

It is with my utmost appreciation that at this time all Interactors give a special thanks to our printer, Ted McAdam of Warren, Mich., because no matter how good the programs are, how well-thought out the features or the articles, without him to put it down on paper, you could never be reading this now. Thank you, Ted!!!

G.A.L.

INTERACTION INTERNATIONAL
GEORGE A. LEGGETT
20562 Woodward
Mt. Clemens, Mi. 48043

$ 3.00 A COPY
$15.00 FOR 6
ISSUES A YR.



INTERACTION INTERNATIONAL

A MAGAZINE FOR INTERACTORS AND FOR ALL PEOPLE WHO WANT THE COMPUTER KNOWLEDGE OF TOMORROW, FOR TODAY

MAR.- APR. 1983 VOL. IV NO. 2

TABLE OF CONTENTS

CREDITS

EDITOR/PUBLISHER	George A. Leggett
FEATURES EDITOR	Lora A. Leggett
PROGRAMMER	George A. Leggett
DRAFTSMAN	Daryl Waananen
ILLUSTRATION DESIGN	George A. Leggett
TRANSCRIPTION	Lora A. Leggett
PRODUCTION ASSISTANCE	Daryl Waananen
PRINTER	Ted McAdam

INTERACT IS KING OF THE HILL
PUBLISHER'S STATEMENT
By George A. Leggett

A lot of things have been happening since our last Publisher's Statement. I've been working, of course, on this issue and no. 3 is coming along gorgeous. But in the meantime, as if this wasn't enough, I have had the chance to work with some other computers. I know it's sacrelige but you want to see what the competition is like. If you can call it that! Just this weekend I had the opportunity to work with a TRS80 for several hours doing a program for a small business. I created the program on the old Interact and later transferred it to the TRS80 removing all the colors and the graphics and the sounds that we're so used to. What began as a terrific program ended up rather blase and boring. It was quite a contrast seeing it on my screen and then the finished product on the TRS80 and this was a Model 3. A brand new system with two disks and 48K. Although you need one disk to run the BASIC, to my surprise. I did not appreciate this to say the least. Lucky I don't own one, nor would I. It's a nice machine for a hundred and fifty bucks but over that, it is not my cup of tea. It's more like...water.

On the other extreme, as if this weekend were not exciting enough, I got the chance to work with a 64K Commodore at my house to use for another project for a business. To make a long story short, I developed a program on the Interact, which the business bought. At the beginning, they were pleased with both the Interact and the program. Then, they saw a Commodore 64 demonstration and freaked out over the resolution and the 16 colors. It has 320 by 200 pixel resolution and all 16 colors may in some way or another be put on the screen at once. It has Sprites graphics which allows you up to 8 movable graphics images at one time. Now these are most amazing. They can go over each other, under each other, left to right, up to down, they tell you when they collide, so you can write some wonderful stuff and use it all with BASIC. If you're wondering what this all has to do with Interact, just hang in there. Luckily, I also have had the use of a demo done by Commodore to illustrate the capabilities of the machine. This demo is a Christmas display. It was extremely impressive when I first saw it, but with 38K of program, you had better come up with something impressive!

They have music playing in 3-part harmony with different instruments like pipe organs, banjos, violins---it's far out! Talk about graphics! Santa is flying over the town while snow is falling and another picture shows two children playing with Frosty the Snowman. But my job is to break it down and see how they made it tick. I had to know this in two days. Why? I must evaluate this machine for the company to decide whether I would be writing programs on it so unless I know all these things, you can see I'm in big trouble. The problem with being in this position is: After you break everything down to its simplest elements it loses all its wonder and magic. It's all, pardon the pun, an optical illusion. Nothing I saw was real.

That may sound totally stupid, but when we use the Interact (I told you we'd get back to it) to make a point on the screen, we simply say PLOT 50, 20, 2 ...it's up there. Not so with the Commodore 64 or the VIC-20. (I strongly recommend to any of you out there that it would be worth your money and time to buy a VIC-20 before you buy a 64 if you're going in that direction. It was only my knowledge of the VIC-20 inside and out that made me adjust to the 64 very quickly. Otherwise, you are going to be totally lost. The documentation of the 64 spends most of its time discussing the Sprite Graphics and the Sound chip It gives very little background on the original VIC manipulation of characters and and keyboard functions. I have both manuals on the 64, the one that comes with the machine and the Programmer's manual, 500 pages long. And believe me, they talk very little about the BASIC. Whereas the VIC-20 Programmer's Guide really goes in depth about the language. If any of you are familiar with either the VIC-20 or the Commodore 64, I'm sure you are aware that each key has several symbols on it and it is a must to know how they work.) So you say, what's so hard about putting a pixel on the screen? Nothing on the Interact. A nightmare, I say, on the Commodore systems. This is because the Commodore is a character generator oriented system. Our Interact is not. We set aside, as you all know, from 4000 Hex to 49A0 Hex for the screen memory. Then by merely calling up the first ROM to turn on or off a given pixel or make a box a rectangle or whatever and there's no problem. But in a character generator system.... On the Commodore, you're only allotted 1024 memory locations. Mind you, the Interact has 2560 locations. So how is resolution greater in the Commodore? Well, for each of their 1,024 locations, each character is a matrix of pixels. This is a dot matrix like our Interact, where our Interact has a matrix of 5 by 5 dots, and theirs is slightly higher. But, whether we put a character or a dot on the screen, we have the full ability to do so. Example: Output the letter O on the screen in white on black. Now, output a green dot in the middle of the O. One command to output the O and another command to plot the dot. It may take you 15 seconds to type that in. In the Commodore 64 or VIC-20 depending how good you are and how fast you are at interpreting binary, you would take up to 15 minutes. Then, we no longer have your O. We must make a character the O with a green dot in it, and store it in a character memory. When you do this, you lose every other character in the character set and unless you copy them bit by bit into other RAM, you won't see a thing on the screen but garbage. If you like playing with binary and characters bitwise, this system is for you. But surely, not for me. I know it, I do it, I have to. But it's a pain in the...anatomy!

Furthermore, you do not power up with 64K of free RAM. They use a technique with which I am familiar and which they call banking. This means that portions of the RAM are switched on and off at different times. Thus, you power up with 38,910 Bytes Free. The system uses all switching, swapping, time sharing. I'm not knocking these other systems, but merely pointing out some of the uniqueness of our old Interact. My friend and ally for life!

With Interact, you are the boss. You are in command. In fact, the man who delivered the Commodore to my house via the company was the Midwest sales rep for Commodore. Lucky me, eh! I sat and showed the man my Interact. Agreed, we do not have the pixels and all of the color. But the man was shocked at the total fasibility to do anything your mind can conceive. The machine is so flexible. That's why I say that the Interact is still the King of Computers!

ABOUT THIS ISSUE: The theme for this issue is cards and card games. As always, I hope you will enjoy and benefit from the material. You do not need FAST GRAPHICS BASIC to run these programs as we are using the Fast Graphics Routine in Vol. IV no. 1. For those of you who have never really played card games, load them in, type them in, and give them a try. There are a couple of what you would call graphic dazzlers, in card format, of course. MACHINE SHOP TALK is back--just read it, love it! Even if you've never programmed in Machine Language in your life--read it, type it in, and sit back and enjoy Interact--now in 8 colors! In HARD FACTS, we adapt the machine this time from the Joy Stick input, namely the Pot.

We're still small and agin appealing to everyone to help get the word out to other Interactors and computer people. The majority of the people are still finding us. I'm talking at this time to 41 people who are subscribers. We're small, we're growing, but hey! Don't give up the ship! We may only be on a single prop plane now. But one day, this single Prop plane may be a Jumbo Jet.

I thank everyone who has submitted material. Everything will be put in at some time. Because I am trying to maintain themes, it may be a couple of issues down the road. Don't feel bashful or shy. Please send it! As you'll read in RANDOM REMS, we are one of the last surviving dinosaurs in exposing good quality BASIC material. I urge you all to submit it, sell it, whatever form you wish. But get it out to the people. One of our Interactors, Dean Anschultz, has submitted so much material that very soon, yes, I will be shutting my mouth for a while, and he will get the entire issue devoted to his work. I felt the work was excellent in quality and extremely well documented. Documentation is important. When you can sit and read clearly what a program does it is a tremendous help. So, my hat off to you again, Mr. Dean Anschultz! Maybe some of you others could have an issue devoted to your work. Just think--if you do, you don't have to listen to my BIG!!! MOUTH!!!

Sincerely,
Your Editor and Publisher Always,

George A. Leggett

RANDOM REMS
By Lora A. Leggett

It is with a little burst of pride that I can say that INTERACTION INTERNATIONAL is to my knowledge the only surviving outlet for good BASIC games for the Interact. In preparing for this issue on cards, our publisher wrote a program which he thought was so well-done that it might have an appeal to the Interact market at large. So, he gave it a shot and submitted it to Micro Video for its consideration. His tape was returned with a brief note from John Stout, Director of Programming. John wrote that Micro Video is no longer purchasing such materials in BASIC, but rather, 8080 Machine Language software. I gave John Stout a call to confirm this and he said that they are no longer seeking games written in BASIC language. So this means that the pages of INTERACTION INTERNATIONAL are a major outlet for your software. Whether you submit it for publication or go ahead and sell it yourself, you can distribute your material directly to other Interactors.

We have been very pleased with the nice letters we have received from many of you. One such example appears in the INTERACTORS INPUT column. It never ceases to amaze me to hear of the different places and circumstances where Interacts can be found. About the most touching example of that is one of our Interactors who has been serving time in a prison. He has been using his time inside those walls to gain knowledge of computer and electronics. Furthermore, he is teaching that knowledge to other inmates in the hope that it will better their lives when they return to the outside world. All of us, in a sense, are imprisoned, maybe, not in a physical sense. But our study of new and different things--be it computer, which is the primary interest of this magazine--or music, or any new knowledge in any field which frees us from that prison of daily life and of ourselves. This kind of dedication and drive for self-improvement is the very thing that helps us to grow as individuals and as a whole species of people. It is my hope that as time goes on, the computer will make mundane tasks of life easier for everyone and that it becomes man's servant to a greater extent. As a programmer's wife, I sometimes casually observe that the reverse may be slightly closer to the truth. ....

We also would like to thank the Interactors who have taken the time to call us. Our telephone number is area 313-791-4243. Because of the fact that George works in the afternoon and night because he's a night p erson and it is quieter at night, and the other fact that our son is just going on 6 months old and has not learned to distinguish night from day, there is usually always someone up around here. So, all of you out on the West Coast need not fear about getting someone out of bed late at night. In fact, night time is usually great for us anyway.

We truly are INTERACTION INTERNATIONAL. We have subscribers in Canada and in France. In the next issue, you can look forward to some startling and exciting contributions from one of our Interactors who lives outside the U.S.A. I shall close for now and say again that I enjoy corresponding with all of you, as, in essence, every note or novel our publisher writes goes through my desk! Keep 'em coming!

MICRO VIDEO™
305 North First St.
P.O. Box 7357
Ann Arbor, MI 48107
(313) 996-0626

February 7, 1982

Mr. "i" responds:

Why are you trying to minimize my role in your first issue's article: "The I's don't have it" ? I (pardon the pun) have been around for a long time! Ever since they've been writing textbooks with standard mathematical notation, "i" and my brother "j" have had priviledged positions as subscripts. Two-dimensional arrays have been invariably referred to as (n x m) in size whose naming positions were claimed by my two distant cousins "n" and "m". "i" is invariably used to index over the rows of such matrices while "j" works down the columns. We've had that job for over 100 years!

So important is my family that we were accorded special integer status as variable names in early versions of Fortran back in 1952. I imagine you were just out of diapers in that year! In those days when matrix inversion was the big thing to do with computers, new programmers ~~invariably~~ copied the algorithms from the books and "i" and "j" became omnipresent in virtually every program. Since then I've been misused occasionally as an array name, but not as frequently as implied in your article. My role as a subscript, however, is secure.

If you want to pick on us letters, kindly look at the more pressing problem of 'O' (oh) and '0' (zero) who together cause a lot more trouble than 'i' has ever done.

Sincerely,

i

Mr. "i"
Micro Video

IN REPLY TO MR. i
By George A. Leggett

Dear Mr. i:

I'm sorry you felt offended and left out of the game. However, I was not really talking about you personally as much as your big brother Capital I. I simply meant to say that on most small printers (under a thousand dollars) it is difficult to distinguish your big brother I from the number 1. Maybe I need glasses for my eyes! The number 0 and letter O controversy is quite another story. On every printer I've seen, a 0 has a slash through it. An O has none. Now, if you, little fellow, were present with your dot over yourself unlike your big brother, you would be quite distinguished between a 1 and an i. But, somehow your big brother gets all the use.

After all, I realize how powerful you have been in mathematics as I have spent a lifetime in math. I know all about your work in FORTRAN and you have been quite powerful. But like all things your day has come and you must make way for newer and better things. We shall not debate your importance to mathematicians over what to use for matrix I prefer R's and C's for row and column. But who am I to say?

So please accept my appology, Small i and remember, I still use you. Every time I work with imaginary numbers. And maybe that's where you best are--a figment of my imagination! This whole story mya be ended with one brief saying: Aye! Aye! Aye!

HOW THE GET KEY GOT GOTTEN
By George A. Leggett

In Vol. 4 no. 1 of this magazine I demonstrated how the Interact could have a GET KEY. I found this command so helpful in my work with my VIC computer that I felt the Interact should have it. Well, lo and behold, to my amazement, it already has one! The very day the printed copies of Vol. 4 no. 1 were returned to me, I was using the book BASICALLY SPEAKING by Micro Video. In Chapter 4 Page 4, it is shown that you can PEEK the keyboard whereby PEEK(24529) returns the last key pressed. In the book, the example given is IF PEEK(24529) = "N" THEN... where N is the letter N that it is waiting for. This is a very important PEEK to keep around. So I feel at this time it is not that one way is truly better, however, the PEEK way does take less memory and is easier to use for most purposes. The only possible advantage in my routine is that it does look for the key in the BASIC Key In routine therefore, if you want BASIC to analyze the key pressed during each instruction, my routine may be of some benefit. Perhaps, in a game, rather than running by the IF PEEK(24529) statement over and over, my routine scans like a Control S or Control C.

Sincd taking over the publication of this magazine, I have done a great deal of research using the books and INTERACTIONS. Perhaps all theese references are the biggest asset we have. We live and learn every day. Use them wisely, so that one day, like me, you will not merely be be gotten, but had! We can all appreciate the many people who have worked long and hard in the past and shall continue to help us to learn more about our Interact.

INTERACTORS INPUT
By Lora A. Leggett

This is only one example of the kind things that so many of you have been writing to us. Time and space will not permit us to print every letter, but we wish to thank everyone for their interest and support.

Dear George:

Please find enclosed the question/renewal form and a check for the 1983 Interaction. I was glad to hear you are taking over this important news letter, it looks like you have everything under control and I know you will turn out a quality product like your "8080 For Everyone". I have passed a copy of your letter and form on to other Interacters so they can rush their subscriptions to you.

I would like to tell you about a bug I found in the program "Fire 3" (Interaction Vol. 2, no. 6), when I tried to run the program as it is listed the computer went to reset, upon checking I found the machine language routine at 5800H was wiped out. I think Basic , in its operation, was writing over the 5800H routine. The only way I could get the program to work was to move 5800H-5865H code to 5900H-5965H and take out the Basic REM statements. You may want to pass this along in the news letter.

Somethings I would like to see in Interaction:

A machine language program that would be a complete program and could be added to or changed by us. I think this would be a big help in getting started in machine language.

More on hardware, like; interfaceing with the joystick input and adding a audio output jack.

Also just for your input I have the Harry Holloway Tape Master program and I think this is an excellent tool for the Interact. I have used it for backing up all of my software (Tape Master was the first backup) and the ASCII dump, disassembler, and assembler are very good. The program is well worth the price.

I am looking forward to reading your newsletters this year, because it makes the Interact more enjoyable. I think the whole Interact group is lucky to have someone like you to tie us together. Thank you and good luck.

Sincerely,

James R Baer

Jim Baer
2325 N. 22nd St.
Lafayette, IN 47904
Phone 317-742-1597

DECK OF CARDS

Perhaps throughout modern history, there has been no single source of entertainment has been as popular as a deck of fifty-two Playing Cards. While there are only 52 cards, excluding the Jokers, the games that one may play are endless. People have gotten many millions of hours of entertainment, not to mention the millions of dollars won and loss. Nevertheless, they are still some of the most used game equipment today. They are easy to attain. They are portable. They can easily be slipped into a pocket and can go everywhere. A simple deck of cards has an unlimited potential; from the hundreds of solitaire games games for 2, 3, 4, 5, or more players. What this issue is all about is a a standard deck of 52 cards for the Interact computer. Now we can program our hearts out—or for that matter clubs, diamonds, or spades! We can not only be challenged by the programming end of it but then we can continue to be challenged by the end result.

The first program in this series is your deck of cards. I shall explain how they work, how to use them and how to get get started right away and get results. The deck of cards is much the same as in the old Interact game of Blackjack. It is refreshing to me to see that such high quality graphics and speed can still be achieved through BASIC. I have used LEVEL II BASIC plus a routine we have talked about and used from Vol. 4 no. 1 Fast Graphics. The Get Key routine is removed here because it did not agree with some of the card programs and I didn't have to use it. Those of you with FAST GRAPHICS BASIC, may alter it if you wish to save memory however, you will only be losing lines 10, 15, 60000, and 60010 so to keep it usable by everyone I left it in LEVEL II BASIC.

This program also has its own character set. While working on this group of programs, I have worked out all the bugs that I know of for the routine. Thus, I added line 61305. Theese are two Pokes to reinitialize the character set every time you're going to use it. When you are using any character set in BASIC and press the Backspace key the character set is disabled. If you do not readdress it, you'll come up with a lot of weird symbols. So, to avoid that problem, every time you call it, it will be initialized.

Lines 100 through 190 are simply in there to demonstrate the program. You may take them out later when you are going to write your own program. They display four rows of eight cards on the screen Upon pressing any key but the S, it will do it again, dealing a new random group of 32 cards. S means Stop to end the program.

About the Program: This is not really a program at all, but a set of four subroutines. As the program runs, you will notice Line 5 clears enough array space to handle the deck and Dimensions A$ for 52. Why A$? The cards are in string. They are not numeric. You could call an Ace of Hearts Card 1, but it is AH. When you look at the data in Lines 42-48 all cards are put out in that way. The only exception is that T means 10 I wanted to keep a simple two-letter code, not feeling it necessary to use a 10 with a suit. The card 10 when output on the screen is a 10 which you will see is a single character. The character set is Lines 30, 35, and 40. In Line 30, the data 7, 7 means that your is a 7 by 7 pixel matrix.

Any data that you wish to have in your own program must be after Line 4% or the interpretation of the subroutine will be all wrong. Line 50 stores the character set data right into the RAM. Once it is there, you need not come back to Line 50 unless you need to use the CSAVE* and CLOAD* functions. Since those two functions are rarely used, machine code is stored in the memory location that BASIC uses for them.

All of the four subroutines should be called by a GOSUB Command since they end with a RETURN. The first routine begins at Line 61000. It reads all the data into the A$ array. Lines 42 through 4% data or all 52 cards are read into A$. There is a RESTORE so that the data will begin loading from the right place. Note here that the variable CL = 52. CL means Cards Left. This is used to keep track of how many cards are left in your pile for stockpile games where you must keep track of that or for knowing how many cards have been dealt out. It can be used for a number of other things. CL is decremented by the routine for picking cards. But here, it is always set for 52 or a new deck of cards. Though the routine takes up two lines, one line is merely for RETURN, which would not fit on the line with the routine.

The next routine sits at 61100. This routine random generates a number from 1 to 52. It then looks to see if that card A$(R) R for Random, is in the deck. If it is not in the deck, there will be a 0 in its place. Thus, it will repeat the process. This is important to remember when the deck is getting low. When looking for the last two or three cards it will take a little longer because it is looping looking for what cards are left in the deck. It's all random so you never know which one it will pick. This card is stored in the variable A$ and a 0 is returned to A$(R) to take that card out of the pack. CL is then decreased by 1 thus, one less card is in the deck. The only place the card will appear is in A$. It could be a KC which is a King of Clubs. Don't worry about decoding this now because the other routine will do this. This is pointed out because you may want to store a Down card.

The next routine is at 61200. This requires two parameters before you may call it. They are XA and YA where XA is for X Axis and YA is for Y Axis. Where do you want to output a face-down card on the screen? This is purely a graphic routine. It does not affect the deck of cards. XA must be from 0 to 100 and YA must be from 77 to 126. If you go beyond that, there is no protection, no Syntax Error but it can goof up your BASIC and your program.

The card is dealt face-down and initialed on the back. You may have a difference of oppinion about the initials I have chosen, but for me, they are the best! You may wish, however, to have other things in mind such as your own initials. Though I cannot see why! But nevertheless if you look at Lines 61200 and 61210 they are simply put on the back of the playing card by an OUTPUT statement. Just substitute your own initials in those outputs.

The tone in Line 61210 is to make the sound of a card being laid down on the table.

The last of our four card routines, and the longest and most complex is to display the card face up. To be sure that cards are displayed in the proper color at the proper coordinate, we start with color. Before calling this routine in your program, set Color 0 as 0 and Color 1 as 1 to make sure that Hearts and Diamonds are red, and Clubs and Spades are black. As in the above routine, you must have the XA and YA parameters with XA between 0 and 100 and YA being between 77 and 16. A$ must have a legal card value in it. Anything other than the Data will give you an improper result. Upon giving the card the XA and YA parameters and a card value from the data in A$, calling the routine at 61300 will output a card on the screen. In our programs, we will use this routine to output a white card with black or red numbers and suit symbols as in a real deck of cards.

This routine returns several useful pieces of information to variables that you may find helpful. Variable Z returns what color was used. It will be 0 for black and 1 for red. The suit played will return in AR$ or A Right as I call it. In AN$ which is converted to T$ it will tell you if it was a T for 10, J for Jack, Q for Queen or K for King. It will also make the sound of the card being
And there you have a deck of cards for the Interact. I have had little difficulty working with this. Here is a brief recap to help make your programming job easier.

GOSUB 61000 Restores the deck.
GOSUB 61100 Picks card out of deck and puts it in A$
GOSUB 61200 Deals card face down given XA and YA parameters.
GOSUB 61300 Deals card face up given XA, YA, and A$ for card.

I hope you will find the routines as useful as I have in making great new card games, solitaire, games against the computer, and, as you will see, games your computer can play! In this issue, I have given what I believe to be a variety of games demonstrating the uses of the deck of cards routines. Although I hope you will enjoy all of the games, think of them as a study guide to help you come up with many more. I hope before very long to be seeing a lot of feedback on new and different card games for the Interact and Interactors.

```
1 REM DECK OF PLAYING CARDS.  NAME "CARDS"
2 REM GEORGE A. LEGGETT 20562 WOODWARD MT. CLEMENS, MICH 48043
3 REM DCECEMBER 11, 1982
5 CLEAR(250):DIMA$(52)
10 POKE19215,25:POKE19473,9:POKE19474,74:PK=18953:POKEPK,1:POKEPK+1,16
15 POKEPK+2,74:POKEPK+3,205:POKEPK+4,162:POKEPK+5,5:POKEPK+6,201
30 DATA7,7,108,254,254,124,56,16,0,16,56,124,254,124,56,16
35 DATA16,56,84,254,84,16,56,16,56,124,254,84,16,56
40 DATA184,168,168,168,184,0,0
42 DATAAH,2H,3H,4H,5H,6H,7H,8H,9H,TH,JH,QH,KH
44 DATAAD,2D,3D,4D,5D,6D,7D,8D,9D,TD,JD,QD,KD
46 DATAAC,2C,3C,4C,6C,6C,7C,8C,9C,TC,JC,QC,KC
48 DATAAS,2S,3S,4S,5S,6S,7S,8S,9S,TS,JS,QS,KS
50 FORX=PK+20TOPK+56:READA:POKEX,A:NEXT
100 CLS:COLOR0,1,2,7
110 GOSUB61000
120 FORYA=74TO10STEP-17
130 FORXA=6TO100STEP13
140 GOSUB61100
150 GOSUB61200
160 GOSUB61300
170 NEXT
180 NEXT
190 I$=INSTR$(1):IFI$<>"S"GOTO100
59999 END
60000 POKE18960,YL:POKE18961,XL:POKE18962,CO:POKE18963,77-YA
60010 POKE18964,XA:US=USR(0):RETURN
61000 CL=52:RESTORE:FORX=1TO37:READA$:NEXT:FORX=1TO52:READA$(X):NEXT
61010 RETURN
61100 R=INT(52*RND(1)+1):IFA$(R)="0"GOTO61100
61110 A$=A$(R):A$(R)="0":CL=CL-1:RETURN
61200 XL=12:YL=16:CO=3:GOSUB60000:OUTPUT"G",XA+2,YA-2,2
61210 OUTPUT"L",XA+5,YA-9,2:TONE5,15:RETURN
61300 XL=12:YL=16:CO=3:C=0:AR$=RIGHT$(A$,1):AN$=LEFT$(A$,1):T$=CHR$(5)
61305 POKE24545,29:POKE24546,74
61310 IFAR$="H"THENC=1:CH$=CHR$(1):GOTO61400
61320 IFAR$="D"THENC=1:CH$=CHR$(2):GOTO61400
61330 IFAR$="C"THENCH$=CHR$(3):GOTO61400
61340 IFAR$="S"THENCH$=CHR$(4)
61400 IFAN$="T"GOTO61500
61410 IFAN$="J"THENT$="J":GOTO61500
61420 IFAN$="Q"THENT$="Q":GOTO61500
61430 IFAN$="K"THENT$="K":GOTO61500
61440 T$=AN$
61500 GOSUB60000:OUTPUTT$,XA+1,YA-1,C:OUTPUTCH$,XA+4,YA-8,C:TONE5,15
61510 RETURN
OK
```

TWO POKER SOLITAIRE GAMES

By George A. Leggett, 20562 Woodward, Mt. Clemens, MI 48043

The following two programs deal with Poker. For those of you who have played the game, I'm sure you'll get a kick out of it. It's simple yet intriguing, and it can get addictive. For those of you who have never played the game, this can be a good teacher. Again, we will be using the Fast Graphics routine and our new Deck of Cards Routine.

In both games, you enter your name start with $1000 in your purse. You may bet any amount of money up to the total amount in your purse. If you go broke or have 0 left, then you must RUN like any gambler would! Only in this program you must type the word RUN to start all over again which is much easier than doing the real thing!

In the first program, STUD POKER, you are asked how much to bet on that hand of cards. Now, you are dealt five cards which are face down and then turned up. At that time, you will instantly know what you have in your hand. The computer will take several seconds to evaluate your hand and give odds based on what the chances are of those five cards being dealt to you from a newly shuffled deck. Thus, for a pair the odds are quite low because there are many combinations. For example, it only works out to 1.364, meaning it happens a lot. Of course the odds of you getting nothing are the greatest. Whereas the odds of getting a straight flush, the highest hand you can get (barring a royal flush which is still a straight flush in this game), will only come up once in 64,973 times that you are dealt a hand. So unless you want to sit around for a very long time, a straight flush will be quite rare. Of course I have never seen it. Please let me know if you do. You will find all theese odds and statistics starting at Line 2000 to 2080. This is based on The Complete Of Indoor Games Editor Peter Arnold.

After the computer has read your hand and displayed what you have, you must then press a key. For the beginners out there, this will give you time to sit back and analyze how the results were arrived at. When a key is pressed, your winnings (or losses) are tallied and put into (or deducted from) your purse. Now, with the new amount of money in your purse, you are asked to bet again and the game proceeds from there so long as you have money in your purse. The game will also keep track of how many hands are dealt. This is for you statistics nuts out there, and to give you some information to make the game more meaningful. What I like to do is to see how many hands it takes me to make a million dollars or more. Beginners may think this is sheer insanity, but I have done it in under 30 hands and right now at this testing took 34 hands. Thus, you can try to better your record as you go along. This to me puts a little spark of competition and meaning to ghe game. Or, how fast can you go broke? Well.... There is no logic for going broke, or 0 money. Just press Control C and type RUN again. I leave the ending up to you. If I receive any good submissions for losing the game, or even for making a million, we would all enjoy seeing a great ending. I leave it in your hands.

DRAW POKER uses the same foundation as as the STUD POKER game. The only changes are Lines 170 through and including 230. DRAW POKER allows you to select as many as three cards, discard them from your original hand, deal you that many new cards, and then give you the odds for that hand of five cards. You must note that because you are dealt extra cards, the odds are not true to what they would be in STUD POKER It does give you a winning edge in the game but because you need a little more strategy in deciding what to keep and what to throw away, it about breaks even. Of course it does take a little longer to play because you are making decisions. You can use the same guidelines as suggested in STUD POKER and try to make a million dollars in so many hands. After this point, your money purse value would start reading out in exponents. This is how LEVEL II BASIC operates, and without spending a great deal of time to put in conditions not doing this, you will just have to be content to have 1E+06 or more in your purse. Of course this leaves us open to become billionaires! And for that matter, as far as we can go, which is 10 to the 37th-aires! Whatever aires they are--this subject is getting a little too windy for me! So, when I take it on the air, I simply leave it at becoming a millionaire.

For those of you subscribing to INTERACTION INTERNATIONAL programs on tape, you will receive both complete programs on the tape so you will merely have to CLOAD either STUD POKER or DRAW POKER. To avoid needless repetition, the listing for DRAW POKER includes only the lines to be changed from the program of STUD POKER which is listed first.

So, I hope you will enjoy the games. I leave you with one thought to keep in your head: You got to know when to throw 'em, and know when to hold 'em.

REGARDING DISCARDING: The discarding option in DRAW POKER comes after you place your bet. You are asked how many cards you wish to discard from your hand. No other keys but 0, 1, 2, or 3 will get a response. Then, you are asked which cards. For example, if you have chosen 3 cards, the computer will wait for 3 numbers. The card in your hand that is farthest left on the screen is 1, Card 2 is to the right of Card 1 and so on until Card 5, which is farthest right. Just press the number of any card you wish to throw away, such as 1 3 5 which are simply prompts, so you need not press the CR key for quickness. You are immediately dealt new cards face down, then when all selections have been made, your new cards are turned over. Then play continues as above and the computer evaluates your hand and pays or deducts the winnings or losses to or from your account.

```
1 REM STUD POKER FOR ONE NAME "STUD1"
2 REM GEORGE A. LEGGETT 20562 WOODWAR MT. CLEMENS, MI. 48043
3 REM DECEMBER 15, 1982
5 CLS: COLOR0,1,2,7: CLEAR(250): DIMA$(52), C(18)
6 O$=" "
10 POKE19215,25: POKE19473,9: POKE19474,74: PK=18953: POKEPK,1: POKEPK+1,16
15 POKEPK+2,74: POKEPK+3,205: POKEPK+4,162: POKEPK+5,5: POKEPK+6,201
20 POKEPK+13,205: POKEPK+14,231: POKEPK+15,7: POKEPK+16,50: POKEPK+17,21
25 POKEPK+18,74: POKEPK+19,201: POKE24650,22: POKE24651,74: GK=18965
30 DATA7,7,108,254,254,124,56,16,0,16,56,124,254,124,56,16
35 DATA16,56,84,254,84,16,56,16,56,124,254,84,16,56
40 DATA184,168,168,168,184,0,0
42 DATAAH,2H,3H,4H,5H,6H,7H,8H,9H,TH,JH,QH,KH
44 DATAAD,2D,3D,4D,5D,6D,7D,8D,9D,TD,JD,QD,KD
46 DATAAC,2C,3C,4C,6C,6C,7C,8C,9C,TC,JC,QC,KC
48 DATAAS,2S,3S,4S,5S,6S,7S,8S,9S,TS,JS,QS,KS
50 FORX=PK+20TOPK+56: READA: POKEX,A: NEXT: PRINTCHR$(7)
60 INPUT"ENTER YOUR NAME ";N$: CLS: PU=1000
100 CLS: GOSUB61000: DD=DD+1: OUTPUT"SHUFFLE",36,35,3
110 TONE5,100: CLS: PU=PU+YB: POKE24545,29: POKE24546,74
115 OUTPUT"$",6,30,3: OUTPUTPU,6,30,2: OUTPUT"IN PURSE",56,30,1
120 POKE19462,6: PRINTN$;" ENTER": INPUT"YOUR BET";YB: IFYB>PUTHENCLS: GOTO120
125 CLS: OUTPUT"$",6,70,3: OUTPUTYB,6,70,2: OUTPUT"IN POT",70,70,1
127 OUTPUTDD,6,18,3: OUTPUT"HANDS DELT",42,13,1
130 OUTPUT"$",6,12,3: PU=PU-YB: OUTPUTPU,6,12,2: OUTPUT"IN PURSE",60,12,1
140 YA=45: FORXA=16TO90STEP16: GOSUB61100: GOSUB61200: C$(XA/16)=A$: NEXT
150 YA=45: FORXA=16TO90STEP16: A$=C$(XA/16): GOSUB61300
160 NEXT
290 FORX=1TO5: L$(X)=LEFT$(C$(X),1): R$(X)=RIGHT$(C$(X),1): NEXT
300 GOSUB1100
310 SF=0: FK=0: FH=0: FL=0: ST=0: TK=0: TP=0: OP=0
320 GOSUB3000
330 FORX=1TO5: Q=L(X): C(Q)=C(Q)+1: NEXT
400 IFR$(1)=R$(2)ANDR$(1)=R$(3)ANDR$(1)=R$(4)ANDR$(1)=R$(5)THENFL=1
410 FORX=2TO14: IFC(X)=4THENFK=1
420 IFC(X)=3THENTK=1
430 IFC(X)=2THENOP=OP+1
440 IFC(X)=1ANDC(X+1)=1ANDC(X+2)=1ANDC(X+3)=1ANDC(X+4)=1THENST=1
490 NEXT
500 IFFL=1ANDST=0GOTO2000
510 IFFK=1GOTO2020
520 IFTK=1ANDOP=0GOTO2050
525 IFTK=1ANDOP=1GOTO2030
530 IFOP=2GOTO2060
540 IFOP=1GOTO2070
550 IFST=1ANDFL=0GOTO2040
560 IFFL=1ANDST=1GOTO2010
600 IFFK=0ANDTK=0ANDFL=0ANDOP=0ANDST=0GOTO2030
900 OUTPUTO$,6,62,3: OUTPUT"PRESS ANY KEY",18,56,2: I$=INSTR$(1): GOTO100
```

```
1000 I$=INSTR$(1):IFI$>"5"ORI$<"1"THENTONE512,100:GOTO1000
1010 I=VAL(I$):RETURN
1100 XA=0:YA=63:CO=0:XL=112:YL=14:GOSUB60000:RETURN
2000 O$="YOU GOT A FLUSH":YB=YB*508:GOTO900
2010 O$="STRAIGHT FLUSH!!":YB=YB*64973:GOTO900
2020 O$="  FOUR OF A KIND":YB=YB*4164:GOTO900
2030 O$="  FULL HOUSE":YB=YB*693:GOTO900
2040 O$="  A STRAIGHT":YB=YB*254:GOTO900
2050 O$="THREE OF A KIND":YB=YB*46:GOTO900
2060 O$="  TWO PAIR":YB=YB*20:GOTO900
2070 O$="YOU'VE GOT A PAIR":YB=INT(YB*1.364):GOTO900
2080 O$="NOT EVEN A PAIR ":YB=0:GOTO900
3000 FORX=1TO5:L(X)=VAL(L$(X)):IFL$(X)="J"THENL(X)=11
3010 IFL$(X)="Q"THENL(X)=12
3020 IFL$(X)="K"THENL(X)=13
3030 IFL$(X)="A"THENL(X)=14
3035 IFL$(X)="T"THENL(X)=10
3040 NEXT:FORX=1TO14:C(X)=0:NEXT:RETURN
59999 END
60000 POKE18960,YL:POKE18961,XL:POKE18962,CO:POKE18963,77-YA
60010 POKE18964,XA:US=USR(0):RETURN
61000 CL=52:RESTORE:FORX=1TO37:READA$:NEXT:FORX=1TO52:READA$(X):NEXT
61010 RETURN
61100 R=INT(52*RND(1)+1):IFA$(R)="0"GOTO61100
61110 A$=A$(R):A$(R)="0":CL=CL-1:RETURN
61200 XL=12:YL=16:CO=3:GOSUB60000:OUTPUT"S",XA+2,YA-2,2
61210 OUTPUT"L",XA+5,YA-9,2:TONE5,15:RETURN
61300 XL=12:YL=16:CO=3:C=0:AR$=RIGHT$(A$,1):AN$=LEFT$(A$,1):T$=CHR$(5)
61305 POKE24545,29:POKE24546,74
61310 IFAR$="H"THENC=1:CH$=CHR$(1):GOTO61400
61320 IFAR$="D"THENC=1:CH$=CHR$(2):GOTO61400
61330 IFAR$="C"THENCH$=CHR$(3):GOTO61400
61340 IFAR$="S"THENCH$=CHR$(4)
61400 IFAN$="T"GOTO61500
61410 IFAN$="J"THENT$="J":GOTO61500
61420 IFAN$="Q"THENT$="Q":GOTO61500
61430 IFAN$="K"THENT$="K":GOTO61500
61440 T$=AN$
61500 GOSUB60000:OUTPUTT$,XA+1,YA-1,C:OUTPUTCH$,XA+4,YA-8,C:TONE5,15
61510 RETURN
OK
```

LINES 170 TO 230 MUST BE ADDED TO THE STUD POKER PROGRAM IN ORDER TO MAKE IT INTO DRAW POKER. YOU MAY THEN CHOSE FROM 0 TO 3 CARDS AFTER YOUR HAND IS DELT.

```
170 OUTPUT"HOW MANY CARDS DOYOU WANT?",6,62,3:I$=INSTR$(1)
180 IFI$>"3"ORI$<"0"GOTO170
190 CD=VAL(I$):GOSUB1100
200 IFCD=0GOTO290
210 OUTPUT"WHICH CARDS",24,62,3:FORX=1TOCD
220 GOSUB1000:YA=45:XA=16*I:GOSUB61200:GOSUB61100
230 C$(I)=A$:NEXT
OK
```

FOR THE INTERACT WHO HAS EVERYTHING:

COUNT and CLOCK PATIENCE GAMES

By Serial No. 006028, 20562 Woodward, Mt. Clemens, MI 48043

You know, I get sick and tired of my master pounding at my keys all day. I mean..7, 8, 9, 10 hours a day, 7 days a week. He never stops! Always writing silly programs for you humans. I could care les about what A$ = or what a FOR Loop does. And then sometimes, he says he's trying to talk to me on my level in something he calls "Machine Language". Hexadecimal is about as far from me as the moon is to you. I deal with highs and lows! What does he know of that? Maybe I'm the only computer who feels this way, but I'm sure I'm not. I mean, he wore my first keyboard to a frazzle, and I had to get a keyboard transplant! Boy, one time I got so mad I made my vertical roll and roll and roll. He had to shut me off for two hours. I wish I had bubble memory so I could remember how I did that. I'm always forgetting everything. Maybe one day. But for now, I'm pretty content in life. I don't ask for much in life--a nice sturdy table to live on and power in my CPU and ROM and RAM. And I'm not difficult to feed, because I don't eat a single byte!

So, as you Interacts can imagine, I was delighted when my master came up with two new games for us to play in our spare time--if we ever have any when we're not RUN-ning errands for those humans. Yes, my fellow Interacts, here are two card games that we can sit back, enjoy and let our highs and lows and our TV screens get carried away with us!

The first game I shall discuss is called COUNT. For you Interacts, just tell your master to load LEVEL II BASIC and enter the program listing and you'll be ready to play! But let's talk to the humans who will be watching us play the game. The object of the game is simple. I must turn over the top card on my pile and count to myself "1". Actually, I count to my variable but...and I give you humans a visual display as I count. You humans always need audiovisual stimulation...why can't you be more like a computer? Watch your white blood cells eat up bacteria or something...I look at my electrons going through my brain (8080 CPU to you humans)...anyway, back to our game. I must try to run out the deck of cards without counting the same number to myself as I turn over from the deck. I begin counting at "1" and count up to "10" and start all over again. An Ace counts as 1, and a Jack, Queen, or King counts as 10 just as does a 10 card. I don't pay any attention to suits in this game or for that matter, either game. They're just there to dazzle you humans and to make the game fair by using a new deck of cards for each deal. So, assume I begin the game by counting "1" and the card I turn over is 8 of Clubs, well, 1 does not equal 8, so I may go on. I say "2" (the screen is my mouth in this case, and turn up another card. which turns out to be a 6 of Hearts, and I'm safe again! Only 50 more cards! I go on like this until I say "10" to myself and it's a 7 of Diamonds, so I start over with "1" and turn up the next card. Oh-oh! It's the Ace of Spades! ER!!!! So, I try again! I will keep playing forever if some wise guy doesn't cut my power! I even display how many deals I've had and whether I have ever won, meaning, run out the deck without matching my counting number with the number on the face of a card.

Boy, those 10's, Jacks, queens, and kings will get you! Even a computer isn't fast enough to beat them! I must confess to you Interacts out there (I'll tell you in binary, because we could never let our fellow human beings know that we computers are any less than perfection) but as many hours as I have played COUNT, I have never yet won the game! I have come close though. My master and mistress say it is possible, but very rare. So, if any of my fellow Interacts can win, please let me know! My name is Serial no. 006028 and I live at the address of the publisher of this magazine. I hope you humans also enjoy watching this game as much as we computers enjoy playing it. Let's face it, it saves you lots of wrist action--therefore, you can save your wrists for popping food into your mouth or lifting drinks to sip on while you watch.

The next game I will show you is a very good game that my master tells me he played as a boy. He taught it to me, and I find it very entertaining. It keeps a computer like me off the streets. Not that I have ever been on a street...but if I were, I would much rather be playing CLOCK PATIENCE. I begin by dealing 12 piles of four cards each in the face of a clock. It's more like a rectangle on my TV screen, but you humans will get the idea. The 13th pile of four cards I put in the middle, thus, all cards in the deck are dealt out. I turn over the first card in the center of my clock. Again, I do not look at the suit, only the numbers on the cards. You know that we computers love numbers! If for instance, that card is a 4 of Spades, it would be placed in the pile that corresponds to 4:00 on the face of the clock. Now I turn over the first card in the 4:00 pile, which is a 9 of Hearts. This goes in the 9:00 pile, and so on. In the face of the clock, Aces are represented by 1:00, 2 through 9 by their corresponding places on the clock, and a Jack is for 11:00 and a Queen is at 12:00. The King--he's the card you really don't want to see in this game--goes into the center pile. The object of the game is to get all of the cards in the face of the clock turned over before the last king is turned over. Thus, if all four Kings come up and you still have 12 cards left, you try again! I have made things easier for you humans by putting little red numbers by each pile of cards. These are to indicate how many cards are left face down in each pile. I am happy to say, I do win at this game. In fact, when I do not win, which is a good deal of the time, I get so annoyed that I shoot myself! While I'm recovering, I display for you humans how many cards I have left. This is so you won't have to worry about counting up all my little red numbers. I know how slow your brains work at times, so that's why I put my little message in the center of my clock. After this I display another screen to stimulate you humans out there with colors. I'll tell you how many decks of cards I have dealt out, how many cards I have left in each effort and how many times I have been successful. Furthermore, I tell you the average of cards left behind in each deal since I began playing. I'm a statistics nut myself. I guess I get a lot of that from my master writing all those programs. He's crazy about keeping track of things like that. No, he's just plain crazy! He even left me on all night one time so I could play till my IC's were content! I played over 300 hands. I had a pretty good average too, leaving an average of 10 cards and I ran out the deck several times.

Let's see what my fellow Interacts out there can do. But, beware! Your human masters will be loading up this game all the time because they will love to watch you play. Talk about a pain in the cassette! You humans would have a headache and be a little out of alignment too if somebody was turning a screwdriver in your head all the time!

I really enjoy this game. I hope you computers, and yes, you living humans too, enjoy it as well. I know you humans colse things by telling each other to have pleasant memories and a fond farewell. So, to my fellow Interacts and to all other computers reading this, I'll close by saying: May all your mrmories be high and low! Our human friends just want to remember the highs. So, from Serial no. 006028, a fond farewell!. Oh, and I guess I should give some thanks to the guy that programmed me to play these great games. Boy, I have to run them, he only pounded my keys and thought of it! But thanks anyway to so much, Mr. George A. Leggett.

```
1 REM CLOCK PATIENCE  NAME "CLKPT"
2 REM GEORGE A. LEGGETT 20562 WOODWARD MT. CLEMENS, MICH 48043
3 REM JAN. 10, 1983
5 CLS: COLOR0,1,2,7: CLEAR(250): DIMA$(52), CK(13)
6 DIMCD(13)
7 DIMCL(13)
10 POKE19215,25: POKE19473,9: POKE19474,74: PK=18953: POKEPK,1: POKEPK+1,16
15 POKEPK+2,74: POKEPK+3,205: POKEPK+4,162: POKEPK+5,5: POKEPK+6,201
20 POKEPK+13,205: POKEPK+14,231: POKEPK+15,7: POKEPK+16,50: POKEPK+17,21
30 DATA7,7,108,254,254,124,56,16,0,16,56,124,254,124,56,16
35 DATA16,56,84,254,84,16,56,16,56,124,254,84,16,56
40 DATA1848,168,168,168,184,0,0
42 DATAAH,2H,3H,4H,5H,6H,7H,8H,9H,TH,JH,QH,KH
44 DATAAD,2D,3D,4D,5D,6D,7D,8D,9D,TD,JD,QD,KD
46 DATAAC,2C,3C,4C,5C,6C,7C,8C,9C,TC,JC,QC,KC
48 DATAAS,2S,3S,4S,5S,6S,7S,8S,9S,TS,JS,QS,KS
50 FORX=PK+20TOPK+56: READA: POKEX,A: NEXT: POKE24545,29: POKE24546,74
60 DATA6668,6374,8468,9263,8448,9243,8428,9223,6628,6310,4828,4510
70 DATA3028,2710,1228,23,1248,43,1268,63,3068,2774,4868,4574
80 DATA3048,4543
100 GOSUB61000: GOSUB5000: FORX=1TO13: READCK(X), CD(X): CL(X)=4:NEXT
110 FORX=1TO13: GOSUB5100:NEXT
120 GOSUB61100:XA=66:YA=48: GOSUB61200: GOSUB61300: GOSUB5200
130 XA=66:YA=48: CO=0:XL=12:YL=16: GOSUB60000:X=CV: GOSUB5100: GOSUB61300
140 OUTPUTCL(X),XC,YC,0: CL(X)=CL(X)-1: OUTPUTCL(X),XC,YC,1
150 IFX=13ANDCL(13)=0GOTO200
160 GOTO120
200 CL=0: FORX=1TO13: CL=CL+CL(X):NEXT
205 FORX=1TO6: SOUND1,20: FORT=1TO50:NEXT: SOUND1,21: FORT=1TO200:NEXT:NEXT
210 XA=30:YA=50: CO=2:XL=52:YL=20: GOSUB60000
220 OUTPUT"YOU LEFT",33,49,3: OUTPUTCL,46,42,3: OUTPUT"CARDS",40,36,3
230 TL=TL+CL: TD=TD+1: IFCL=0THENTR=TR+1
240 REM SONG LOOP
250 FORX=250TO1STEP-1: TONEX,10:NEXT
300 CLS: COLOR3,4,5,0: OUTPUT"TOTAL DEALS=",6,70,1: OUTPUTTD,72,70,3
310 OUTPUT"TOTAL CARDS LEFT",6,60,2: OUTPUTTL,48,54,3
320 XA=0:YA=40: CO=3:XL=112:YL=34: GOSUB60000
330 OUTPUT"YOU HAVE RUN OUT THE DECK",6,38,0: OUTPUTTR,54,32,2
```

```
340 OUTPUT"TIMES",78,32,0:OUTPUT"YOU LEFT AN",26,24,2
350 OUTPUT"AVERAGE OF",27,18,2:OUTPUTTL/TD,6,12,0
360 OUTPUT"PER DEAL",60,12,2
370 REM SONG PLAY
380 FORX=1TO250:SONEX,15:NEXT
390 CLS:COLOR0,1,2,7:GOTO100
4999 I$=INSTR$(1):END
5000 RESTORE:FORX=1TO39:READA$:NEXT:RETURN
5100 XA=INT(CK(X)/100):YA=CK(X)-XA*100:XC=INT(CD(X)/100):YC=CD(X)-XC*100
5110 GOSUB61200:OUTPUTCL(X),XC,YC,1:RETURN
5200 CV=VAL(T$)
5210 IFT$=CHR$(5)THENCV=10
5220 IFT$="J"THENCV=11
5230 IFT$="Q"THENCV=12
5240 IFT$="K"THENCV=13
5250 IFT$="A"THENCV=1
5260 RETURN
59999 END
60000 POKE18960,YL:POKE18961,XL:POKE13962,CO:POKE18963,77-YA
60010 POKE18964,XA:US=USR(0):RETURN
61000 CL=52:RESTORE:FORX=1TO37:READA$:NEXT:FORX=1TO52:READA$(X):NEXT
61010 RETURN
61100 R=INT(52*RND(1)+1):IFA$(R)="0"GOTO61100
61110 A$=A$(R):A$(R)="0":CL=CL-1:RETURN
61200 XL=12:YL=16:CO=3:GOSUB60000:OUTPUT"3",XA+2,YA-2,2
61210 OUTPUT"L",XA+5,YA-9,2:TONE5,15:RETURN
61300 XL=12:YL=16:CO=3:C=0:AR$=RIGHT$(A$,1):AN$=LEFT$(A$,1):T$=CHR$(5)
61310 IFAR$="H"THENC=1:CH$=CHR$(1):GOTO61400
61320 IFAR$="D"THENC=1:CH$=CHR$(2):GOTO61400
61330 IFAR$="C"THENCH$=CHR$(3):GOTO61400
61340 IFAR$="S"THENCH$=CHR$(4)
61400 IFAN$="T"GOTO61500
61410 IFAN$="J"THENT$="J":GOTO61500
61420 IFAN$="Q"THENT$="Q":GOTO61500
61430 IFAN$="K"THENT$="K":GOTO61500
61440 T$=AN$
61500 GOSUB60000:OUTPUTT$,XA+1,YA-1,C:OUTPUTCH$,XA+4,YA-8,C:TONE5,15
61510 RETURN
OK
```

```
1 REM COUNTING CARD GAME FOR INTERACT  NAME "COUNT"
2 REM GEORGE A. LEGGETT 20562 WOODWARD MT. CLEMENS, MICH 48043
3 REM WRITTEN DECEMBER 12, 1982
5 CLEAR(250):DIMA$(52):COLOR0,1,2,7:CLS
10 POKE19215,25:POKE19473,9:POKE19474,74:PK=18953:POKEPK,1:POKEPK+1,16
15 POKEPK+2,74:POKEPK+3,205:POKEPK+4,162:POKEPK+5,5:POKEPK+6,201
20 POKEPK+13,205:POKEPK+14,231:POKEPK+15,7:POKEPK+16,50:POKEPK+17,21
25 POKEPK+18,74:POKEPK+19,201:POKE24650,22:POKE24651,74:GK=18965
30 DATA7,7,108,254,254,124,56,16,0,16,56,124,254,124,56,16
35 DATA16,56,84,254,84,16,56,16,56,124,254,84,16,56
40 DATA184,168,168,168,184,0,0
42 DATAAH,2H,3H,4H,5H,6H,7H,8H,9H,TH,JH,QH,KH
44 DATAAD,2D,3D,4D,5D,6D,7D,8D,9D,TD,JD,QD,KD
46 DATAAC,2C,3C,4C,6C,6C,7C,8C,9C,TC,JC,QC,KC
48 DATAAS,2S,3S,4S,5S,6S,7S,8S,9S,TS,JS,QS,KS
50 FORX=PK+20TOPK+56:READA:POKEX,A:NEXT:POKE24545,29:POKE24546,74
55 PRINTCHR$(7)
100 CLS:CT=0:CC=0:CN=0:GOSUB61000:OUTPUT"DEALS",76,72,2:DC=DC+1
105 OUTPUTDC,70,66,3
107 OUTPUT"WON",82,42,1:OUTPUTGW,82,36,3
110 XA=8:YA=35:GOSUB61200:OUTPUT"CARDS",6,72,2:OUTPUT"LEFT IN",6,66,2
120 OUTPUT"PILE",6,60,2:IFCL=0GOTO200
125 GOSUB61100:OUTPUTCL,2,42,3
130 XA=50:YA=35:GOSUB61200:GOSUB61300:CC=CC+1:IFCC=11THENCC=1
140 OUTPUTCC,50,42,3
150 IFT$="K"ORT$="Q"ORT$="J"ORT$=CHR$(5)THENCN=10:GOTO170
155 IFT$="A"THENCN=1:GOTO170
160 CN=VAL(T$)
170 IFCN<>CCTHENXA=0:YA=42:CC=0:XL=70:YL=6:GOSUB60000:GOTO110
180 FORX=100TO1STEP-1:TONEX,5:NEXT:GOTO100
200 GW=GW+1:SOUND3,332:OUTPUT"I AM A WINNER!!!",6,12,3
210 FORX=1TO2000:NEXT:SOUND7,4096:GOTO100
60000 POKE18960,YL:POKE18961,XL:POKE18962,CC:POKE18963,77-YA
60010 POKE18964,XA:US=USR(0):RETURN
61000 CL=52:RESTORE:FORX=1TO37:READA$:NEXT:FORX=1TO52:READA$(X):NEXT
61010 RETURN
61100 R=INT(52*RND(1)+1):IFA$(R)="0"GOTO61100
61110 A$=A$(R):A$(R)="0":CL=CL-1:RETURN
61200 XL=12:YL=16:CC=3:GOSUB60000:OUTPUT"G",XA+2,YA-2,2
61210 OUTPUT"L",XA+5,YA-9,2:TONE5,15:RETURN
61300 XL=12:YL=16:CC=3:C=0:AR$=RIGHT$(A$,1):AN$=LEFT$(A$,1):T$=CHR$(5)
61310 IFAR$="H"THENC=1:CH$=CHR$(1):GOTO61400
61320 IFAR$="D"THENC=1:CH$=CHR$(2):GOTO61400
61330 IFAR$="C"THENCH$=CHR$(3):GOTO61400
61340 IFAR$="S"THENCH$=CHR$(4)
61400 IFAN$="T"GOTO61500
61410 IFAN$="J"THENT$="J":GOTO61500
61420 IFAN$="Q"THENT$="Q":GOTO61500
61430 IFAN$="K"THENT$="K":GOTO61500
61440 T$=AN$
61500 GOSUB60000:OUTPUTT$,XA+1,YA-1,C:OUTPUTCH$,XA+4,YA-8,C:TONE5,15
61510 RETURN
OK
```

MACHINE SHOP TALK

COLOR MY WORLD

By George A. Leggett, 20562 Woodward, Mt. Clemens, MI 48043

Welcome to another MACHINE SHOP TALK and in this issue I've got a great bombshell to lay on all you Interactors out there. What I'm about to show you may sound unbelievable at first and at second. But it is not only the truth but it is remarkable. I will hold you in suspense no longer. The fact of the matter is your Interact can display eight colors. Big Deal. You knew that all along. No, I don't mean one at a time. Not four at a time, that's no big deal either. But how about eight colors at one time on the screen? No, I have not gone off my rocker and I'm not making this up. I have now loaded the program which follows this article and am staring at my color TV screen which is outputting four columns of eight different colors in each column at one time. Black, red, green, yellow, blue, magenta, cyan, and white all at once! Not only those eight at once but they're in four areas at once giving a combination that appears to be 32 colors at one time. Talk about Color My World! You never thought the Interact had it in it! Well, it does. I'm sure when we all got our Interact, we experimented with the COLOR statement making different colors, different prints on different backgrounds and we all had a ball at this. Then we began to put a COLOR statement in a FOR loop for weird effects. But we always wished we could have more than four colors at one time. Or at least I did. If there's eight colors, why can't we look at them all at once?

The hardware limitations do not allow this. But through software especially Machine code, we can have eight colors at one time. I myself have outputted eight colors on one letter before, but never on one whole screen. I wanted to do this, but I never could get it together until I paid a visit to Micro Video in September, 1981. I talked with Tom Matulevich and he showed me an interesting program that displayed eight colors at one time. Needless to say, I was shocked. What a brilliant idea! And after being shown how it was done...of course everything is simple when you know how to do it. It looked like a magic act, I must say. Well, I have been bothered for some time now of how to publish this concept without infringing on a promise I made Tom which was not to let anyone know. And I am keeping my promise to him although it may not appear that way and I do not wish to offend anyone. I had this developed over 8 months ago and feel the time is right now to let you know how to do it. (An instant update to this as final transcribing of this article is being completed: In checking a reference for the correct spelling of Tom's last name I came across even another way to achieve eight colors. This can be found in Micro Video RAM Pages Vol. 2 no. 1 Spring 1982 Page 12 "Small Bytes" by Anthony Watson.)
My program is nothing at all like Tom's. Tom's took several hundred bytes as I recall. I could be wrong and I appologize. But it was quite lengthy. This program can be done in BASIC using a USR code. The subroutine for making the eight colors is stored in DATA in Line 30 and Line 40 where Line 50 reads it into the memory. That is all the memory needed to get your eight colors.

The CPU and ROM 1 insert new color information and as the sync circuit at that particular second grabs the information it will then update it with the new color. Before it gets down the screen you can update it thirty or forty times. Therefore, you have a rainbow effect of colors. This is how we output 8 colors at one time.

That brings us to another problem. What if we want to output different groups of colors at different parts of the screen? There are limitations but we can get around them. If, in our subroutine, we increment every color by 1 in this way: Assuming we start with the statement COLOR 0, 1, 2, 3. Incrementing them now makes them become COLOR 1, 2, 3, 4. Increment them again to make COLOR 2, 3, 4, 5. If we output four colors on the screen, as normal, let them be rectangles. Each color will be 1 greater than the other color. The left color would be 1 color behind that of the second column and so on. We are first defining our color statement as Black, Red, Green, Yellow. We update it to Red, Green, Yellow, Blue. We keep changing that information before our rectangles have time to be plotted. If this sounds confusing, it may very well be. All I can say at this time is load in the program, type RUN, and seeing is believing.....

Now that you have run the program you should be a little bit shocked at the results. Why are we doing this in BASIC if this is supposed to be the MACHINE SHOP TALK? Good question! The reason for my decision is to capture the whole audience. I felt that if I gave it to you in all Hexadecimal as loading H's and L's and incrementing B's and C's I would lose 90% of the crowd. But this way, I want everyone to take full advantage of what Machine Language can do for you with a little help from BASIC. You do not have to become a wizzard at the 8080 to get a needed result. You can combine the forces to use the 8080's Language and BASIC as one. That is the goal with which I began the year and it is the way I intend to keep on going. I do not believe any longer that you can have one without the other. The binding of the two languages will give you that much more power than you had before.

I'm sure that if you have run the program and have had the chance to see the results for yourself, you can come up with an enormous amount of of ideas of your own. For those of you who just want to use it as it is, fine! Have a ball with it. Output boxes and letters at different areas of the screen. You're going to come up with some very impressive work. For those of you who want a more in depth insight on what is going on, we shall now look at the machine language listing of the program. There is much versatility in doing this kind of work in Machine Language. Because 1/60 of a second is approximately .016 of a second we have a lot of time. In BASIC that is not not enough time to do anything. But in Machine code, where the average instruction is 8 Machine Cycles taking about 4 millionths of a second, you can cram 4,166 instructions into that time before a screen update. If we divide that number by 8 for the 8 colors, we can still get about 520 instructions while we update our 8 colors. That's half a K of 8080. If you program in 8080 you know how much you can do in that much memory.

Another drastic difference between Tom's meghod and mine was the trick, which I did not know until he pointed it out to me. Note: all one color must be on one horizontal line. Meaning, starting at the top of the screen, let's say, 3 pixels are black, then 3 are red, 3 are green, 3 are yellow, ... and so on, repeating the pattern down the screen. Well, after I thought about that for some time I said there had to be a much better way so we can have one line be of multiple colors. So, here's what we have. When you enter the program listing, you will see four columns, each column having a different sequence of eight colors. Going left to right, a typical row can be green, red, yellow, blue or black, white, red, green. Of course there are four times eight combinations or thirty-two different row combinations. I have outputted the sentence "HOW DO YOU LIKE MY 8 COLORS" and as you see that will change colors

While all of this makes for one heck of a colorful display and it is really exciting to see the Interact do so much there are limitations. It is true that while Tom's program would not allow this, mine will put out four columns of colors. Nevertheless, it is only four columns. The limitation is this: You can have as many groups of eight going from up to down but you can only have four different colors going from left to right in any one row. Let me explain why this is and how the program works. First of all, when you load in the program, Lines 10, 15, 60000 and 60010 are my Fast Graphics Routine. This is used to output rectangles. If you have the Microsoft 8K FAST GRAPHICS BASIC, you may want to substitute this data. Lines 30, 40, and 50 put in the Machine Code for the eight colors. Lines 100 through 170 simply draws three boxes in column form and outputs the sentence "How do you like my 8 colors". From there, the two Pokes are given for the USR and the USR takes over. When you type RUN what you will see is four columns with eight colors in a column rolling down the screen. They do not stay stationary however you can make it stay stationary. We will get into that later. I programmed it to roll down the screen. It looks like your vertical hold has gone haywire but because the sentence always remains in the center of the screen there's nothing wrong with your vertical hold. Here's how it works.

We all know about the statement COLOR 0, 1, 2, 3 which is in BASIC. Well, in Machine Code, it works the same way. You load B and C with the address of where your Color Bytes are. In BASIC they're at 4A00, 01, 02, 03, and 04 When you load B and C with the address of where to find your colors, then you simply CALL 0636 in ROM 1 which sets the colors automatically. We can trick the machine so to speak because of the hardware and the way the machine was designed. The process of setting the colors takes 1/60 of a second. Now, to you and me 1/60 of a second may not seem like much time to worry about. But to the 8080 it can have a field day in 1/60 of a second! We are tricking the machine by incrementing Color 0 calling the Color routine and then incrementing Color 0 again, calling the routine again...in a loop. What happens is we get line after line of the color we're changing before the 1/60 of a second to update the whole screen can occur. Now, our TV scans at a rate of 30 frames a second. Half of that time, it is blanking out and going back to the top of the screen. It is during this time that we can keep telling it ...I don't want black, I want red..no, green,..and we can give it many many of these instructions before it gets down to the bottom of the screen.

Although we're addressing this in BASIC it is definitely a Machine Language usage to get the most out of it. Here is a look at the Machine Language routine. First, we load H and L with where the color bytes are stored which is 4A00. The next several bytes simply increase each memory address by 1. We load B and C with 4A00 then call up our Color routine in ROM 1. Now, the next few bytes load B and C and call up the Delay routine in ROM 1. The delay is there to get the colors in sync so they won't rnn fast. The next four No Op's are not there because I forgot to do something with memory, they're there for timing purposes. Thus, this is the first time in our articles that we are using a No Op for something other than No Operation. It is performing an enormous operation! It is delaying the machine two millionths of a second to get us in sync so we have a nice steady picture. We then call 07E7 which is the keyboard routine in Rom 1. If no key is pressed, we tell it to jump back to 49A0 and do the whole process all over again. If we press a key, C9 RETURN. Return where? To BASIC. Thus, ends our program. I suggest that you vary the delay loops for fun to get wider or narrower color strips. The variety is unlimited.

I hope that you will enjoy and use this program. If you find any great ways to use it, please submit it. Let's share it with our fellow Interactors. I hope I have enlightened you a little bit more, as is always my goal in the MACHINE SHOP TALK and in INTERACTION INTERNATIONAL.

Knowledge is not mere black and white but it is a rainbow of color. And that is why I give you a rainbow of color. To open your world and color all our worlds.

Sincerely,

George A. Leggett

```
1 REM MACHINE SHOP TALK FOR 8 COLORS NAME"8COLR"
2 REM GEORGE A. LEGGETT 20562 WOODWARD MT. CLEMENS, MICH 48043
3 REM JAN. 22, 1983
10 POKE19215,25:POKE19473,9:POKE19474,74:PK=18953:POKEPK,1:POKEPK+1,16
15 POKEPK+2,74:POKEPK+3,205:POKEPK+4,162:POKEPK+5,5:POKEPK+6,201
30 DATA33,0,74,52,35,52,35,52,35,52,1,0,74,205,54,6,1,48,0,205,246,7,0,0
40 DATA0,0,205,231,7,202,160,73,201
50 FORX=18848TO18880:READA:POKEX,A:NEXT
100 CLS:COLOR3,1,2,4
110 XL=30:YL=77:XA=25:YA=77:CO=1:GOSUB60000
130 XA=0:CO=2:YL=77:YA=77:XL=25:GOSUB60000
140 XA=85:CO=3:XL=27:GOSUB60000
150 OUTPUT"HOW DO YOU",25,50,3
160 OUTPUT"LIKE MY 8",28,44,3
170 OUTPUT"COLORS",37,38,3
180 POKE19473,160:POKE19474,73
190 U=USR(0)
59999 END
60000 POKE18960,YL:POKE18961,XL:POKE18962,CO:POKE18963,77-YA
60010 POKE18964,XA:US=USR(0):RETURN
OK
```

HEXADECIMAL LISTING			DECIMAL LISTING	
49A0	21	LXI H	18843	33
49A1	00		18849	0
49A2	4A		18850	74
49A3	34	INR M	18851	52
49A4	23	INX H	18852	35
49A5	34	INR M	18853	52
49A6	23	INX H	18854	35
49A7	34	INR M	18855	52
49A8	23	INX H	18856	35
49A9	34	INR M	18857	52
49AA	01	LXI B	18858	1
49AB	00		18859	0
49AC	4A		18860	74
49AD	CD	CALL	18861	205
49AE	36		18862	54
49AF	06		18863	6
49B0	01	LXI B	18864	1
49B1	30		18865	48
49B2	00		18866	0
49B3	CD	CALL	18867	205
49B4	F6		18868	246
49B5	07		18869	7
49B6	00	NOP	18870	0
49B7	00	NOP	18871	0
49B8	00	NOP	18872	0
49B9	00	NOP	18873	0
49BA	CD	CALL	18874	205
49BB	E7		18875	231
49BC	07		18876	7
49BD	CA	JZ	18877	202
49BE	A0		18878	160
49BF	49		18879	73
49C0	C9	RET	18880	201

4A00	03	COLOR 0			
4A01	01	COLOR 1	18944	3	YELLOW
4A02	02	COLOR 2	18945	1	RED
4A03	04	COLOR 3	18946	2	GREEN
			18947	4	BLUE

BY VARYING THE VALUE IN REGISTERS BC AT 49B1 AND 49B2 YOU WILL BE ABLE TO VARY THE ROLL AND WIDTH OF THE COLORS. IF YOU ONLY WANT ONE OF THE FOUR COLORS TO VARY SIMPLY PUT NOP'S IN 49A4 TO 49A9. IF YOU WANT TO INCORPORATE OTHER ROUTINES WITH THIS ONE TRY TO INSERT YOUR ROUTINE AT 49BA, 49BB, AND 49BC. ALL AND ALL I HOPE YOU HAVE A MOST COLORFUL EXPERIENCE AND A RAINBOW OF SOFTWARE IDEAS.

THE HARD FACTS OF LIFE

BY

GEORGE A. LEGGETT 20562 WOODWARD MT. CLEMENS, MICH. 48043

<=><=><=><=><=><=><=><=><=><=><=><=><=><=><=><=><=><=><=><=><=><=><=>

INPUT HAS GONE TO POT

LAST ISSUE YOU GOT A CHANCE TO KNOW ME AND BEGIN A NEW ADVENTURE IN ELECTRONICS. WE TALKED ABOUT HOW TO HAVE YOUR INTERACT TALK TO THE OUTSIDE WORLD WITHOUT WIRING A SINGLE WIRE TO YOUR COMPUTER. THIS ISSUE WE ARE GOING TO GIVE THE INTERACT A SET OF EARS. THIS TIME WE WILL HAVE TO CONECT A CABLE TO INTERACT BUT HAVE NO FEAR THIS IS EASIER DONE THAN SAID. WE ARE GOING TO USE OUR OLD FRIENDS JOY, POT, AND FIRE INPUT JACKS.

DON'T CUT UP YOUR JOYSTICK CABLE YET OR FOR THAT MATTER EVER. THERE IS A MUCH EASIER WAY TO GET A JOYSTICK CABLE AND THAT WAY IS TO MAKE IT. BEFORE I TELL YOU HOW YOU CAN DO THIS LET ME TELL YOU WHAT YOU CAN DO WITH IT.

OVER THE YEARS THAT I HAVE BEEN ASSOCIATED WITH INTERACTIONS MANY PEOPLE HAVE ASK ISN'T THERE A WAY TO HOOK UP OTHER DEVICES TO THE INTERACT. WELL THERE ARE AND IT IS FAIRLY EASY TO DO IT. I WILL TELL YOU ABOUT TWO DEVICES THAT I HAVE HOOKED UP TO THE INTERACT WITH GREAT SUCCESS. BOTH DEVICES ARE HOOKED TO THE POT INPUT.

THE POT INPUT IS A SORT OF OHM METER IN ITS OWN WAY. IT INTERPRETS RESISTANCE AND ASSIGNS A DIGITAL NUMBER TO IT. THIS IS CALLED AN A TO D CONVERTER. A STANDS FOR ANALOG WHICH IS ANY DEVICE THAT PUTS OUT A VARIED PATTERN. D IS FOR DIGITAL AND A DIGITAL SIGNAL GOES UP OR DOWN IN SET INCREMENTS.
AN EXAMPLE OF THIS IS THE FOLLOWING; GIVEN A 8 BIT DIGITAL NUMBER THERE ARE 256 STEPS THAT MAY BE ACHEVED (0-255) AS YOU ALL KNOW ALREADY. WITH 8 BITS YOU CAN NEVER HAVE THE NUMBER 156.25 IT WILL EITHER BE 156 OR 157. IN ANALOG HOWEVER YOU CAN HAVE ANY NUMBER OF DIVISIONS YOU WANT BETWEEN 156 AND 157 SUCH AS 156.0625 WHICH IS 156 AND 1/16 IN FRACTIONAL FORM. HAVE YOU EVER TRIED TO GET ONLY A 16TH OF A BIT TO TURN ON LATELY. ITS HARD NOT TO SAY IMPOSSIBLE AT BEST. SO WHY USE DIGITAL WHY NOT USE AN ANALOG FORMAT IN OUR COMPUTER? WELL SOME OF THE EARLIER COMPUTERS WERE. THE TROUBLE WITH THIS IS TO GET THE ACCURACY OF A 8 DIGIT NUMBER THE DEVICE WOULD BE EXTREMLY LARGE. ALL WE HAVE TO DO IN A DIGITAL COMPUTER IS JOINED TOGETHER MORE 8 BIT BYTES AND WE CAN HAVE ANY SIZE NUMBER WE WANT. CONSIDER THAT OUR BASIC USES FOUR BYTES PER NUMBER FOR A FLOATING DECIMAL ARRANGEMENT. WHAT IF WE WANTED A VERY LARGE INTEGER NUMBER. ALL WE HAVE TO DO IS ASSIGN MORE BYTES FOR OUR NUMBER LET'S SAY 10 BYTES NOW WE CAN HAVE A NUMBER FROM 0 TO 2↑80 AND THAT'S ONLY 10 BYTES COULD YOU IMAGINE 8K OR 16K OR EVEN 32K FOR ONE NUMBER, DR. CARL SAGAN EAT YOUR HEART OUT OR BETTER YET YOUR GOOGOL PLEXES*.

ALL THIS IS IMPORTANT TO KNOW IF YOU ARE GOING TO HOOK ANALOG DEVICES TO THE INTERACT AND EXPECT TO GET USABLE RESULTS.

* CARL SAGAN TALKS ABOUT VERY LARGE NUMBERS IN PART 9 THE LIVES OF STARS IN THE SERIES COSMOS. A GOOGOL PLEX IS 10↑10↑100.

INTERACT SEES THE LIGHT

I HOOKED A CADMIUM SULFIDE PHOTO CELL TO THE POT INPUT AND WAS MORE THAN HAPPY WITH THE RESULTS. THE CADMIUM SULFIDE CELL CDS FOR SHORT IS FOUND ON PAGE 114 OF THE 1983 RADIO SHACK CATALOG PART NO. 276-116. BASICALLY WHAT IT DOES IS CONVERT LIGHT TO RESISTANCE AND THAT IS EXACTLY WHAT THE POT INPUT UNDERSTANDS, RESISTANCE. FOR ONLY $1.29 YOUR INTERACT CAN SEE!

THE POT INPUT ON THE INTERACT IS APPROXIMATELY 0 TO 10,000 OHMS. THE CDS RANGE IS 3 MILLION OHMS IN TOTAL DARKNESS TO 100 OHMS IN BRIGHT LIGHT. AT FIRST IT MAY NOT SEEM COMPATIBLE BUT IT WORKS QUITE WELL. DEPENDING ON THE RANGE OF THE A/D CONVERTER IN YOUR INTERACT YOUR UPPER AND LOWER LIMITS MAY VARY. (3 to 255)

THE APPLICATIONS OF A DEVICE LIKE THIS ARE MUCH TOO NUMEROUS TO DESCRIBE HERE BUT I SHALL TRY AND GIVE YOU A FEW EXAMPLES.

HOME SECURITY SYSTEM: PUT ONE CDS AND LIGHT SOURCE BY EACH DOOR AND WINDOW.

AUTOMATIC REFRIGERATOR OR FREEZER DOOR DETECTOR: WHEN THE DOOR IS LEFT OPEN THE LIGHT COMES ON TRIPPING THE DEVICE.

MEASURING DEVICE: SUCH AS A WIND SPEED INDICATOR OR SPEEDOMETER FOR EXERCISING BIKE.

THESE ARE JUST A FEW OF THE MANY THINGS YOU CAN DO. ONE OF MY WRITERS HAS ASKED FOR SOMETHING FOR HIS DARKROOM. TRY THIS OUT AND HOOK UP THE CIRCUIT IN VOL. IV NO. 1 AND NOW YOU HAVE A WAY TO SENSE LIGHT AND THEN TRIGGER ANOTHER CIRCUIT BASED UPON A RESULT. THIS CAN BE A VERY POWERFUL COMBINATION IF YOU LET YOUR IMAGINATION RUN WILD.

LET US NOT FORGET AT THIS TIME THAT BEHIND EVERY GOOD HARDWARE DEVICE IS THE SOFTWARE THAT MAKES IT ALL COME TOGETHER. IT IS VERY HARD HERE TO LIST ONE SUCH PROGRAM FOR ANY PARTICULAR DEVICE. THE REASON IS OBVIOUS, IT MAY BE ONLY ONE DEVICE BUT IT COULD HAVE AN INFINITE NUMBER OF USES. THAT'S THE GREAT THING ABOUT SOFTWARE.

I SHALL GIVE YOU ONE QUICK EXAMPLE OF A PROGRAM TO MEASURE THE WIND SPEED. AFTER YOU BUILD YOUR STRUCTURE TO CATCH THE WIND MOUNT A LIGHT SOURCE ABOVE THE ROD WHICH WILL ROTATE AROUND AND THE CDS BELOW RIGHT IN LINE WITH THE LIGHT SOURCE. YOU MAY HAVE TO MAKE A LIGHT SHIELD FOR THE CDS SO NO OTHER EXTERNAL LIGHT CAN EFFECT THE RESULTS. NOW IF YOU ARE USING FOUR CUPS TO CATCH THE WIND YOU MUST DIVIDE ONE ROTATION BY FOUR. ONCE YOU KNOW YOUR UPPER AND LOWER LIMITS YOU CAN WRITE YOUR PROGRAM. FIGURE OUT HOW MANY REVOLUTIONS EQUAL ONE MILE PER HOUR AND YOUR IN BUSINESS.

THIS IS ONLY A START OF WHAT COULD BE A GREAT PROGRAM. FOR EXAMPLE YOU COULD INCORPORATE A TIME/DATE ROUTINE INTO YOUR PROGRAM AND RECORD THE WIND AVERAGES OVER MANY DAYS OR WEEKS AND SAVE THE RESULTS ON TAPE. ANOTHER PROGRAM COULD TURN THIS DATA INTO A BARGRAPH AND GO ON YOUR PRINTER.

AS YOU CAN SEE ANYTHING THE MIND CAN IMAGINE YOU AND YOUR INTERACT CAN DO!

INTERACT CAN GET BURNED AND FEEL IT

AFTER HAVING SUCH GREAT SUCCESS WITH THE PHOTOCELL I ASKED MYSELF WHAT OTHER VARYING RESISTOR DEVICE COULD I HOOK TO THE POT WITH EASE. THE ANSWER WAS SIMPLE; A THERMISTOR. A THERMISTOR IS RESISTOR WHICH VARIES WHEN THE TEMPETURE CHANGES. THIS PART IS NOT ATTAINABLE AT RADIO SHACK WHICH COULD BE A PROBLEM FOR SOME OF YOU. I BOUGHT MINE AT RS ELECTRONICS IN ROSEVILLE MI. YOU CAN USE A WIDE VARIETY OF VALUES BUT THE ONE I SELECTED IS 10,000 OHMS AT 25 DEGREES CELCIUS. THIS IS BY WORKMAN ELECTRONIC PRODUCTS INC. SARASOTA, FL. MODEL FR1007. I GIVE THIS INFORMATION TO YOU SO IF YOU HAVE ANOTHER ELECTRONIC OUTLET WHERE YOU LIVE YOU MAY INQUIRE FOR THAT PART OR THE EQUIVALENT. THE MAIN THING IS THE 10,000 OHM VALUE OR 10K (K=1000).

THE FIRST THING I DID ONCE I HOOKED IT TO INTERACT VIA THE POT, JOY, FIRE CABLE WE WILL BE MAKING UP SOON, IS TO DUMP THE THERMISTOR INTO A BUCKET OF ICE WATER AND THEN INTO BOILING WATER. I THEN HAD SOME IDEA OF ITS RANGE. THE PROGRAM I USED IN BOTH EXPERIMENTS (PHOTOCELL AND THERMISTOR) IS THIS;

```
10 PRINTPOT(0):GOTO10
```

IT PROVIDES A CONSTANT VIEW OF YOUR INPUT.

THE THERMISTOR LIKE THE PHOTOCELL HAS ONLY TWO WIRES. IT DOES NOT MATTER WHICH WIRE GOES WHERE AS IN ANY RESISTOR. I THEN SOLDERED SOME WIRE TO BOTH ENDS AND WRAPPED ELECTRICAL TAPE AROUND THE CONNECTIONS TO INSULATE THE TWO FROM TOUCHING. THIS IS THE SAME FORMAT I USED IN WIRING THE PHOTOCELL. THE EXTRA WIRE IS NEEDED AS THE LEADS ON THESE COMPOENTS ARE QUITE SHORT (1 TO 2 INCHES). THE EXTRA WIRE GIVES YOU THE FREEDOM TO MOVE ABOUT WITH THE PROBE.

NOW WE HAVE A DEVICE THAT CAN TELL HOW COLD OR HOT SOMETHING IS. IF YOU RECALL ONE USE FOR OUR PHOTOCELL WAS A FREEZER OR REFRIGERATER ALERT. WELL IF THE DOOR IS SLIGHTLY OPENED THE LIGHT MAY NOT COME ON AND THERE MAY NOT BE ENOUGH EXTERNAL LIGHT TO EFFECT THE DEVICE. THE THERMISTOR HOWEVER WOULD SENSE THE TEMPETURE RISE AND SET OF YOUR WARNING SYSTEM VIA YOUR COMPUTER. THIS AGAIN IS ONE OF COUNTLESS APPLICATIONS FOR THE THERMISTOR.

AT THIS TIME I FEEL IT IS IMPORTANT TO TALK ABOUT CERTAIN SOFTWARE TECHNIQUES. BECAUSE EVERY INTERACT HAS A DIFFERENT UPPER AND LOWER LIMIT IN ITS A/D CONVERTER THIS PRESENTS US WITH A SMALL PROBLEM, BUT SOFTWARE ALWAYS HAS A WAY AROUND IT. THE ONE SURE WAY TO GET AROUND THIS I FOUND IS TO MAKE UP A TABLE AND STORE IT IN DATA STATEMENTS. LET US ASSUME THE FOLLOWING SEQUENCE.

```
DATA 66,67,68,70,72,74,75,77,79,82,84,85,87,88,91,94,95,96,99,102
```

WHEN YOU USE THE DATA IN THE FOLLOWING WAY YOU HAVE A NICE WAY TO CONVERT OR INTERPRET THE POT READINGS.

```
100 X=0:RESTORE
110 X=X+1:READA
120 IFX<>POT(0)GOTO110
130 PRINT A;"DEGREES IS THE TEMP"
```

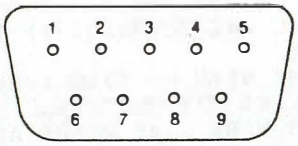
YOU WILL HAVE TO MAKE UP YOU OWN TABLE OF DATA FOR YOUR INTERACT BUT THAT COULD BE HALF THE FUN

MAKING A POT, JOY, FIRE CABLE

MAKING THE CABLE IS A VERY SIMPLE BUT TEDIOUS PROCESS. FOR ME LUCKILY I ACQUIRED AN END OF A JOYSTICK CABLE WHILE INTERACT WAS STILL IN BUISNESS. BUT HAVE NO FEAR YOU CAN BUY THE JACK AT RS ELECTRONICS. THEN JUST HOOK UP WIRE TO EACH PIN TO ANY DESIRED LENGTH YOU WISH. THEN WRAP SOME ELECTRICAL TAPE AROUND THE JACK END TO PROTECT THE CONNECTIONS. RS ELECTRONICS SALES A COVER FOR THE 9 PIN JACK BUT IT IS NEVER IN STOCK AND DOES TAKE 4 TO 8 WEEKS TO ORDER. THESE TWO ITEMS COST APPROXIMATELY $7.00 PLUS YOUR WIRE FOR HOOK UP. I HAVE TRIED TO FIND A PLACE THAT WILL SELL THE CABLE ALREADY MADE BUT HAVE HAD NO SUCCESS YET.

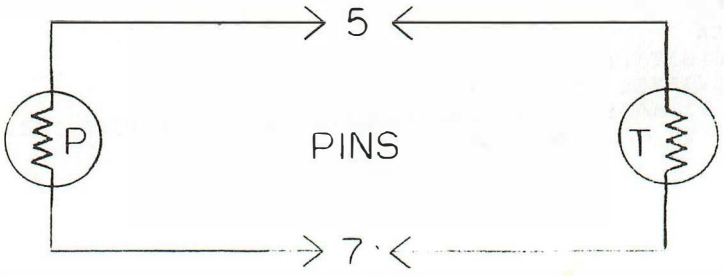
THE FOLLOWING IS A LIST OF PIN CONNECTIONS AND WHAT THEY DO WHEN HOOKED TO THE INPUT OF THE INTERACT. REFER TO THE DRAWING OF THE JACK. KEEP IN MIND THAT THE PICTURE IS THE VIEW OF THE JACK AND NOT THE INTERACT INPUT.

PIN	FUNCTION
1	RIGHT JOY
2	LEFT JOY
3	UP JOY
4	DOWN JOY
5	POT
6	FIRE BUTTON
7	POT
8	GROUND
9	NOT USED



IN THIS ISSUE WE ARE ONLY USING PINS 5 AND 7 ON THE JACK BUT IT IS A GOOD IDEA TO WIRE UP THE WHOLE JACK NOW BECAUSE IN FUTURE ISSUES OF THE HARD FACTS OF LIFE WE WILL BE USING THEM. PIN 5 AND 7 IS WHERE THE PHOTOCELL AND THERMISTOR ARE CONNECT TO GIVE THE RIGHT RESULTS. FOR THOSE OF YOU WHO WANT TO GO ON AND EXPERIMENT ON YOUR OWN. I SHALL QUICKLY TELL YOU HOW THE OTHER PINS WORK.

TO USE ANY OTHER FUNCTION OTHER THEN THE POT SIMPLY TOUCH THE GROUND WIRE TO ANY OTHER PIN. EXAMPLE TOUCHING PIN 4 (DOWN ON JOY) WILL RETURN A 8 IF YOU TYPE PRINTJOY(0) AND PRESS THE CR KEY. BUT FOR NOW LET'S JUST WORK ON THE POT WIRES 5 AND 7. THE PICTURES BELOW SHOW THE SCHEMATIC EQUIVALENT OF THE PHOTOCELL AND THERMISTOR. AS ALWAYS I WISH YOU THE BEST IN YOU ENDEAVORS WITH THIS NEW FIELD. IF YOU HAVE ANY PROBLEMS ACQUIRING THE PARTS I SHALL SEND YOU A COMPLETE KIT INCLUDING: 1 THERMISTOR, 1 PHOTOCELL, 1 9 PIN FEMALE JACK WITHOUT COVER AND 100 FEET OF 22 GAUGE HOOK UP WIRE ALL SENT 1ST CLASS. MAKE CHECKS OR MONEY ORDERS TO GEORGE LEGGETT FOR $15.00 TO THIS ADDRESS.



ADVERTISEMENT

CRAZY CARDS: A new twist on the old favorite CRAZY 8's. You can play against the computer either CRAZY 8's or all-new CRAZY CARDS, where a new card is wild in every game! Guaranteed to be an exciting and enjoyable card game.

The game was designed to play and score correctly according to Hoyle's Rules for Crazy 8's, unlike the game sold by another video game system which does not.

Comes Complete with rules of CRAZY 8's by Hoyle, instructions on CRAZY CARDS, and Program Listing. Send $9.00.

UNO CARD GAME. Challenge the computer as you would a friend to the popular game of UNO. Still Only $5.00.

COLOR BARS: This represents my first 8080 free standing program. Turn your Interact into a color bar generator. Still only $5.00.

THE EIGHTY EIGHT CONSTELLATIONS: Learn about the stars and have fun at the same time. This Machine Language program features a unique musical presentation lasting 4 minutes as its "double Header". Also includes a free Planet Locater Chart while supplies last. Send $8.00.

8080 PROGRAMMER PACKET: Includes the book, 8080 FOR EVERYONE, the tape and book, MACHINE LANGUAGE RAM ROUTINES, and the X - Y PLOTTING MONITOR with 2-page Instructions. Price, $22.00.

8080 FOR EVERYONE a simplified English book of Machine Language Instruction sold separately $15.00

MACHINE LANGUAGE RAM ROUTINES Includes tape and 10-page book of all routines published in 8080 FOR EVERYONE with updates and several new routines. Price $10.00.

X - Y PLOTTING MONITOR enables you to plot a picture on the screen then store it in Machine Language for use in your program. Price $5.00.

COMMODORE VIC-20 SOFTWARE: Three new offering for the VIC-20

HEXI MONITOR: If you love working in Hexadecimal on the good old Interact, this is one of the only ways to do it on your VIC. This Monitor features 11 functions such as dumping registers, Decimal to Hex and vice versa, Go with Break and many other conveninet features. Includes 7-page documentation. Requires 8K or more memory expansion for your VIC to run. Send $7.00

MATH ADVENTURE: A 4-Function math tutoring program which works with or without a Type 'n Talk Speech synthesizer. Send $7.00. Also Requires 8K memory expansion for VIC-20.

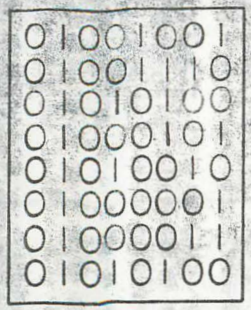
ALPHA HILOW: Play the old number HILOW or new Alphabet HILOW. Needs only the memory that comes with VIC. Send $7.00.

FOR ALL THREE VIC PROGRAMS: $18.00.

SEND TO: George Leggett, 20562 Woodward, Mt. Clemens, MI 48043

INTERACTION INTERNATIONAL
GEORGE A. LEGGETT
20562 Woodward
Mt. Clemens, Mi. 48043

$ 3.00 A COPY
$15.00 FOR 6
ISSUES A YR.



INTERACTION INTERNATIONAL

A MAGAZINE FOR INTERACTORS AND FOR ALL PEOPLE WHO WANT THE COMPUTER KNOWLEDGE OF TOMORROW, FOR TODAY

MAY - JUNE 1983 VOL. IV NO. 3

TABLE OF CONTENTS

CREDITS

Editor/Publisher	George A. Leggett
Features Editor/Transcriptionist	Lora A. Leggett
Programmers:	George A. Leggett, Warren J. Moore, Jerry Goerz
Production Assistance	Olga A. Leggett
Draftsman	Daryl Waananen
Printer	Ted McAdam

INTERACT: THE NEW FRONTIER or, HOW TO TEACH AN OLD COMPUTER NEW TRICKS

Publisher's Statement by George A. Leggett

I am really excited about this issue for several reasons. 1. For the first time, we get to incorporate material from other Interactors which makes me happy and should make all of you happy to get some other input. 2. Not only do we have Interactors in the U.S.A., but we also get a special treat from across the sea. Our contribution this time is from France. Not only have they given us good food and and great loving but now they're getting into the computer market. What won't they think of next? Finally, it is the biggest issue ever and my smallest publisher's statement ever because there is only this page to say what I have to this time. My many thanks to all the contributors who helped to put this magazine together. It also gives me great pleasure to introduce something I have spent a great deal of time working on: my CMD Overlay for BASIC. This will give you great graphics with the ease of BASIC. Just read all about it and enjoy. I also urge you not to pass by the letter from Mr. Moore about our old friend RESTORE and how it has a new form. I didn't know it existed which is amazing. I am still finding out new things about a machine and its language that I spend an average of eight hours a day on seven days a week. This is about the most intriguing thing about computers to me; the never ending possibilities that can come out of them. My thanks also to Alfred E. Jackson and Dean Anschultz for sending me the great 8080 listings. Mine were worn to a frazzle. Oh, I memorized most of them but let's face it, we all forget a command or two sooner or later. It must be toe old age or the kids! And we have our largest HARD FACTS ever, 10 pages, which finally lets our Interact mouth off. My thanks to Jerry Goerz who submitted his speech synthesizer article. Finally, we end up with our largest section of advertisement ever with ads from other Interactors. So, sit back, enjoy, and please address any comments to me, George A. Leggett, at this address. As always, I will try to answer all questions. Keep submitting the great work!

I'd like to close this by thanking two people. First, to Ted, our printer. He does a tremendous job and is such a tremendous help in all our endeavors. I did not tell you about our new format because it happened when I delivered the magazine to be printed. I along with every other Interactor I've talked to am more than happy with the results. Thank you, Ted, for the great idea and the work you put into it. Last but not least, there's a person that I really couldn't do this without and that's my wife. She deserves a round of applause in doing this issue.

Thanks for all your letters, calls and compliments. Here's to a great issue--at least I think so. What do you think? Here it comes!

Sincerely,
George A. Leggett

RANDOM REMS
By Lora A. Leggett

Mercy! Mercy! MERCI BEAUCOUP!!!

Although a home, husband, two small children and helping out with INTERACTION INTERNATIONAL can be challenging to say it nicely at times, it does have its dividends and strange things happen sometimes. You can imagine my shock when the phone rang one morning in February and a voice on the other end with a French accent and the most beautiful English informed me that he was calling from France! Well to say the least I was excited. I had never spoken to anyone in any other country before except the U.S.A. and Canada, which to those of us who have grown up in the Detroit area is not really thought about as another country since one merely has to get into a car and take a short half hour or so drive to Canada.

The person who not only called me once but called me again the very next morning was none other than Michel Henric-Coll of a company you will be reading about in this magazine called MICRONIQUE. He wanted to know whether we could find a book about the FORTH language. Happily, we found the book at a store less than a mile from our home, reported this informatinformation to him the next morning and a few weeks after that, the book went its merry way across the Atlantic! At INTERACTION INTERNATIONAL we will do our best to help out in any way we can. We thank and appreciate all of the support and information we have received from our French Interactors, Patrick Duguet and Michel Henric-Coll. One is immediately impressed with their superb command of the English language. We who live in America are somewhat behind in our ability to use any other language besides English. But after all, computer languages could be thought of as a second language and as this century draws to a close the society as a whole will be forced increasingly to become what could be called "computer literate". Sometimes, I have been told that I am very computer literate. I don't think I possess enough logical thinking in my entire being to put two lines together, but when you live with a person who eats, drinks, thinks, dreams, sleeps and what ever else one could do with time on computers, some of it is bound to rub off or else a marriage could develop communication problems! Otherwise I would be lost in doing transcription, calling stores to chase down parts, asking questions, answering correspondence, etc. It's almost hard to believe that when I went to school an address was merely the place where you lived and RAM was what you did when you bumped into each other and an apple was a piece of fruit and basic was just an adjective for simple things.

Nostalgia is nice but does nothing to move us forward. We salute the leaps taken by French and Japanese as well as our own developers to help bring us all into the computer age. Perhaps as time goes on, it will be difficult to distinguish whether computers are speaking our language or whether we are speaking theirs.

THE NEW FRENCH REVOLUTION
Or... THE INTERACT IS ALIVE AND WELL
AND LIVING IN FRANCE

On the following pages is a fascinating story about our very own Interact and the changes the machine has undergone in France. It is transcribed directly from a personal letter from one of our French Interactors, Patrick Duguet. The first paragraph of the letter has been omitted as it did not pertain to this subject. All of the flavor and text of Patrick Duguet's letter has been retained; it was merely read onto tape by George A. Leggett and typewritten by Lora A. Leggett. Let's get our passports ready for an exciting voyage with Patrick Duguet into the world of the Interact computers in France.

Patrick Duguet
"Cante-Coucut"
Bostens
40090 Mont De Marsan
France

I am going to tell you all the things I know about Interact in France. In fact French don't know Interact. The computer's name is Victor Lambda. By the way from the outside it's the same computer it's the same design. Only the colors are changing. VICTOR LAMBDA is gray the keys are professional and very well.

VICTOR LAMBDA is built by MICRONIQUE. They are producing many computers every month. Rom 1 is the same as Interact for full capability with American software. Memory map too is the same. Joy sticks are the same. When I buy a tape from the U.S.A. I can load it without problem. On the screen colors are wonderful thanks to Periceleotim French system which turns the TV into a monitor.

In fact there are three Victor Lambda. Victor I is a 16K computer with an 8080 clock at 1.75 mhz. and resolution 113 by 77 (text 12 by 17). As you see it is exactly the same computer as Interact.

VICTOR LAMBDA II has the same design but inside the card is different. It's a 48K computer clock 1.7 MHZ. but the Microprocessor is Z80. Resolution is the same as Victor I. Between 0 and 800 is always the same old Interact ROM. thanks to compatibility between Z80 and 8080.

VICTOR LAMBDA II H.R. (for High Resolution) is too 48K with a Z80 but the clock is 5.1 MHZ. ROM 1 is always the same but ROM 2 between 800 and FFF is there for High Resolution Routines. High Resolution is 241 by 231 (the draws are very fine) for the text. It's fantastic with 22 by 37. The new character set is in ROM 2. It has the draw pixel by pixel like the ROM 1 character set. There are upper and lower characters. The screen addresses for High Resolution are between C000 F9C0. When you switch the computer you can choose to load a tape in High Resolution or come back to the 113 by 77 resolution to load a tape from MICRO VIDEO. for example.

For printers theese computers have parallel ports inside. Soon they will get to RS232 port. MICRONIQUE is working on a Victor IIIwith with two Floppy Disks inside it. The disks will be fully different.

Characteristics will be 12K ROM (between 0 and 7FF always Interact ROM. Between 800 and FFF High Resolution Routines. And between 1000 and 3FFF BASIC in ROM). compatible with C.P.M. Microprocessor will be Z80 In fact there will be only a part of BASIC in ROM. This new BASIC is very powerful and has been written by MICRONIQUE Software Department. It's 12K long. It's designed for Victor III and Victor II H.R. For Victor II H.R. it's in RAM between 4000 and 5F9F and between 6000 and 6FFF. User program begins at 7000 to BFFF. Above there is High Resolution screen. The name of this fantastic new BASIC is BASIC III. I got it since two weeks (I've not said to you that I get a Victor Lambda I and a Victor Lambda II H.R. with a SEIKOSHAA GP80 printer. I hope to buy soon an Epson MX80).

Now the characteristics of BASIC III.
First the Cursor. Yes it's a real cursor flashing upon the letter. You can insert delete. Backspace don't erase the letter but drives the cursor back. Tab Key drives the cursor forward. With CONTROL R you delete the letter under the cursor. With CONTROL T you insert a letter. With CONTROL D you come back to the beginning of the line. With CONTROL F you go to the end. With EDIT and the number line you edit the line with the cursor as in professional computers. CONTROL O is for Lower Case CONTROL P for Upper. In CONTROL O Mode with SHIFT you get upper case. In CONTROL P Mode with SHIFT you get lower case. With "SHIFT" you get # With "SHIFT" you get & and with "SHIFT/" you get @ LOCK is now a Repeat Key for all the characters. In EDIT Mode you can modify syntax of the lines too number of the lines. You get too CONTROL C to cancel the line you are editing and CONTROL S to go out of editing Mode and find again the time before the addition. Buffer of the line is designed for 256 characters. You can shorten the token (GOS for GOSUB LP for LPRINT ...e). You can use Hexadecimal & ex. USR&FB00 or POKE&FF42.

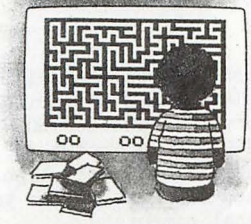
PRINT: For the screen this BASIC III allows the user to do fantastcc things. You get first the old Print but this is too PRINT/M where M is 0 for nothing 1 for screen only 2 for printer only 3 for screen and printer.
PLOT: has a different syntax from MICRO VODEO BASIC. Color is at the end. Example PLOT10,230,232,230,0 (to clear High Resolution screen but it's not the better way) Coordinates X Y are at the left of the square you want to draw.
LINE: No change (same characteristics as MICRO VIDEO BASIC)
PEN: You can select color of print or input. Example COLOR 0, 1, 2, 3 with PEN1 print and input will be in red. With PEN0 they will be in black and you will see nothing.
BRIGHT: To get half intensity of Color 2. BRIGHT0 = full intensity. BRIGHT1 = half intensity.
FLASH: Easy to understand. With FLASH3 you flash Color 3. FLASH3,12 flash Color 3 12 times. The default value of the second parameter is 16.

Victor a plus d'un tour dans son sac.

1

les jeux électroniques...

Jouer, c'est la moindre des choses lorsqu'on est un ordinateur astucieux qui sou-



haite devenir l'ami de toute la famille. Victor propose des jeux de réflexes, de stratégie ou bien d'intelligence – par exemple les terribles gloutons affamés qui dévorent tout sur leur passage ou bien les étranges envahisseurs venus d'une autre galaxie, il vous faut un rayon laser pour les combattre. Ou encore les échecs

car Victor sait être à volonté un partenaire facile, sérieux ou redoutable. Il suffit de changer de cassette et vous changez d'univers...

2

mais aussi la programmation...

Les ordinateurs qui ne savent que jouer, Victor les trouve un peu bêtes ; avec Victor,

vous pouvez apprendre à parler avec un ordinateur ou plutôt à lui donner des ordres par écrit : c'est la programmation. Avec Victor, c'est facile d'apprendre le B-A-BA d'un langage universel et simple : le Basic. Vous pourrez écrire vos propres programmes et surtout vous apprendrez à "conduire" un ordinateur. Bientôt, ce sera aussi important que de savoir se servir d'une voiture.

3

et la fonction domestique :

Victor sait aussi [illegible] de bons petits plats. Chargez une cassette "recettes" et Victor met son tablier. Indiquez le nombre de convives, il calcule les proportions exactes sans rien oublier, ni le sel, ni les épices.
Changez de cassette, Victor quitte ses fourneaux et devient le financier habile qui va tenir les comptes de la famille.

Ce que vous écrivez sur Victor me donne envie d'en savoir plus. Sans engagement de ma part, adressez-moi votre documentation :

Nom _____

Prénom _____

Adresse complète _____

_____ Age _____

Profession _____

O.I.

Retournez ce bon à V.L.D.
61 rue Fernand Laguide
91100 Corbeil-Essonnes

LAMBDA

victor

the friendly computers

Victor I 16 k : 2.950 F TTC
(prix généralement constaté)

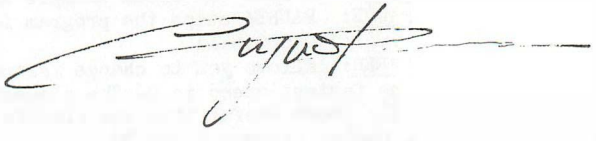
SCREEN: Really fabulous. You can define your screen with scrolling anywhere on the video screen. Example SCREEN50,200,100,100 is X = 50. Y = 100. (coordinate at top left) Width of screen is 50+100. Height of screen is 200-100. After that, all the prints are scrolled inside the screen. If you want to write outside the screen you can use the old OUTPUT. It's really fantastic to see text scrolling in a little square at the right top or in the middle of the screen.
To clear the screen you can use CLS or CONTROL L CLS only clears the screen you have defined. If you want another color for your screen you can write CLSN where N is the position of the color in the last COLOR instruction. If you want to clear the full screen you can get WIPE which too holds the curser at the left top. CONTROL K produces the same effect.
SCROLL: With that you can do some mad things. Scrolling from left to right, right to left, scrolling slow or scrolling very fast.
CURSOR(x,y) To put the cursor on the next Print or Input anywhere on the screen.
POS: POS0 To know where the last character was printed on the screen.
POS1 To know column where the last character was printed on the screen.
POS2 To know horizontal position (in pixel) of the cursor.
POS3 To know vertical position (in pixel) of cursor.
You can do very useful combination of POS2 and POS3 with the cursor.
ELSE: Yes, there is an ELSE at last with GOTO and GOSUB. You can use a variable. Example I=2000:GOTO I
USR: Approximately the same as MICRO VIDEO 32K BASIC but you can use Hexadecimal with &
CLEAR: With the second parameter you can define the RAM spot variable to reserve a place for Machine Language routines.
SWAP: To exchange two variables. Example SWAPA,B (is the same as C=AA, A=B, B=C but much quicker) If you want to give the same value to many variables you can write for example A,B,F,E=0 A$,B$,F$(I,J)= "HELLO" Variables too are MAX and MIN. Example A=MAX(X,10) A can't be inferior to 10.) B=MIN(X,10) (B can't be greater than 10)
TISET: Puts the clock to 0
TIME: To read the clock Example A=TIME(50) gets the clock for a second.
PAUSE: PAUSE4 stops the program for 4 secondsMore useful than FORI=1TO1000:NEXT
SPEED: allows you to change fastness of execution of the program. The fastest speed is 0. The slowest speed is 65,536
RND: More useful than the classic RND Example A=RND(8,50) gives you a number between 8 and 50.
SEED: is a sort of randomizer
INKEY$: Input a key but does not stop the program (CALL 7E7)
INSTR$: Input a string but stops the program (CALL 7E0)
TAPE: TAPE1 is Tape On and TAPE0 is Tape Off.
LOAD: Like CLOAD but you can load a language machine program.Be careful.
INP OUT: To receive and send data from/to the port.

AUTO RUN: is not an instruction but if you do POKE FF38,1 before a save when you load the program it will be auto running.
ERROR: To control errors example 10ERROR!))) 1000IF EC=2THEN EDIT EL EC and EL are system variables. In EC you find Error Code (from 1 to 21) and in EL the number of line where is the error.
ASC: You can get second parameter example ASC(A$,4) gives to you ASCII code of the 4th character of A$.
REM: No change but you can use Slash. In the listing REM never appears. It's always replaced by /
PI: is a new function like SIN, COS, LOG, EXP, etc.
EDIT: without the number of line drives you into the Editor. Yes, there is too an editor. This editor is very powerful. With AUTO, RENUMBER, LOCATE, MERGE, APPEND, FREE, LIST, LAST, EXTRACT, DELETE, and of course BASIC to come back to BASIC. It's more powerful than EZ EDIT. For example RENUMBER is good for multiple statement lines. LOCATE no. You can do RUN from the EDIT Mode.

Now the software. Yes I know BASIC is software but it's very special. For the other programs nothing to say. You know them. They are exactly the same with French names of course. In the screen the sentences are in French too. Since one year we do not get new programs in France because there are some negotiations between MICRONIQUE and MICRO VIDEO. We get all the old Interact programs and some old MICRO VIDEO programs.
By now the new MICRONIQUE Software Department has written BASIC # and High Resolution FORTH. With they have written a very good High resolution Cat Race.

That's all. In the letter you'll find two ads. An ad from a big store in Paris with many computers among them them the Victor LambdaI . The second ad is from Victor Lambda Diffusion (V.L.D.) Every time I get new functions I'll send them to you.

Sincerely,



From all of the Interactors and from INTERACTION INTERNATIONAL a sincere Thank You and MERCI BEAUCOUP, M. Duguet, for a most enjoyable and most informative letter.

CMD: NEW BASIC COMMANDS OVERLAY
By George A. Leggett

I don't know where to begin this long story except to say that I am very excited about theese new commands for BASIC. Of course I should be since I programmed them! But that's beside the point. It is hard to be unbiased and objective but I shall try. But as the song says, it's hard to be humble! All kidding aside, I must say that this is a very powerful tool for all of us to use with our BASIC Language and I shall take some time explaining what this is all about and the history of how it happened and the changes it has already undergone before you even had the chance to see and use my new commands for BASIC.

By using the new CMD commands you will truly have for the first time in your Interact a BASIC with fast graphics. It will give you the ability to make boxes, draw lines, make triangles and fill in shapes. Furthermore, by using the subroutine supplied in the Demo program which follows the overlay, you can make circles; both unfilled and filled. This is accomplished as fast as Machine Language will allow and yet with the ease and simplicity of the normal BASIC commands. No messy Pokes or Peeks. Just nice X and Y BASIC commands which we are all used to using.

This whole thing got started while I was doing some commercial work with 32K BASIC. As you know, 32K BASIC has the LINE command as well as the extended PLOT statement as Microsoft 8K FAST GRAPHICS BASIC does. This was great for 32K BASIC and my commercial work. But for LEVEL II BASIC I was in the dark. I do not like being in the dark and without power in my machine for long. So I began serious study of all of the documentation of the BASIC language and other things I could find or had known before. I found the work of Walt Hendrickson and Harry Holloway in INTERACTIONS Vol. 3 no. 1 to be extremely helpful and I also studied the work of Dave Schwab. All of this plus my knowledge of Machine Language finally came together in a new command which began my first overlay. This first command and overlay you will never see. All of the routines work fine so I will briefly explain them here. It went out of date before it ever went to press, which was only a matter of three weeks. Ah, the wonders of technology! Here today and gone by midnight! Nevertheless, I will explain that the commands were: SET, which is totally not needed and shall join the ranks of Get Key, and shall be explained later, TRI for triangle, BOX for Box, FILL for fill, SKETCH for Lines, and SET# for changing Print Color where the number sign was for the color you wanted your print. Nothing wrong with the routines...but there was something very wrong with them and for that matter with all overlays I have been familiar with so far. They took away some of the BASIC words. Obviously, to put in words you must take away words. For the record, the words that got the proverbial ax in my first version were: WINDOW, DEF, FN, TAB, and LET. In my first writing of this article, I spent a great deal of time explaining the way to do without each command and how to get the same function in BASIC. Since they are still here, I need not go through that now.

How could I take away nothing and add five new commands? We do

have the word DUMMY and one opening. I may be a dummy but I did not use it in my first overlay nor my second. You still have that word.

The one word I did take away to do all of this is the word LET. If anyone out there still uses LET... well, I'm sorry I took your word away. But let's be realistic. I don't know of a single soul who would object to a word that does absolutely nothing. If there are any objections, please let me know and I'll tell you what I'll do about it; absolutely nothing! I will print your letter as to why you use it. I do not mean this as a sarcasm because recently I worked with a Timex Sinclair computer which really does use the LET command. Amazing! But that's another story.

So how did I get five commands out of one word? Not meaning to be one I must say I thought my way of doing it was damn clever. The key word I use in all my commands is CMD for Command followed by a letter with which you are commanding. The following are the new command words of this new BASIC overlay to give you an example of the format you will use to enter them.
CMDB The B stands for Box. CMDF The F stands for Fill.
CMDL The L stands for Line. CMDP The P stands for Print Color.
CMDT The T stands for Triangle. Therefore, you can get by with the one key word CMD and any number of routines such as CMD followed by A through Z, 0 through 9 or any shift of the numbers. Note: The Plus, Minus, Times or Divide By signs will not work in this manner as they are arithmetic functions but anything else on the keyboard will. This gives you more command words than RAM will possibly allow. At least on our 16K machine. Maybe when we get our 48K machine like the one you will be reading about in this issue, who knows? But there are any number of command words you can use and never alter any of the original commands. I thought this was terrific if I do say so myself.

There is one other thing that you did lose from BASIC, not inthe words but in the memory. Because the routines are so lengthy and immense, and because I did not want to take any RAM away from your BASIC I did take the RAM used for the CSAVE* and CLOAD* functions. That is where the main routines for the lines and triangles sit. You can still use CSAVE* and CLOAD* but their use will destroy the Line and Triangle commands. Even though for those of you who really dig down deep, there are another 140 or so bytes free in BASIC, and theese will be used for other commands coming up soon, the 140 bytes were not enough for the Line and Triangle routines. They take 265 bytes of memory. So there is still room for more routines. The area from 49A0 to 49FF is still free so you may put any routines you want in there. The other opening is in BASIC itself. I shall not take the time now to go into the memory location of each routine and the Hexadecimal information. I feel that for the majority of you this would not be of interest. However, if there are enough letters or calls, I'll be happy to publish the Machine Language listing. The following program which loads the overlay is done in BASIC and this in itself is very unique. It is worth several moments of thought and discussion. I am a Hexadecimal man. I think, dream and program in Hexadecimal. Of course I am the most at home when I'm doing 8080. Therefore, as you will notice on the following pages, everything is stored in Data. But when you look at the data it's in

Hexadecimal 8080 commands. At Line 64000 of the program, the Hexadecimal two bytes are converted to a decimal number whereby it is then poked into a specified address. Although it may look like a lot of numbers and alphabet, any Machine Language buffs out there can sit down and start from the first Data statement and read it in Machine Code. If you want to break it all apart, everything is there for you. I have found this to be the best way of entering Machine code in BASIC. It was easier to make a routine to handle all of the converting from Hex to decimal than to use the mind to do it all the time. This should work well for your USR routines or any Machine Language needs you may have.

I shall now discuss the loading and use of the new CMD commands. Load the program as you would any normal LEVEL II BASIC program. Type RUN and wait until it is finished. I have put in a couple of Print statements to let you know it is not locked up. It does take a while to run it and Poke in all that data. Several hundred bytes are being poked and by the time the Hexadecimal string is converted to numeric decimal it does take time. After the program is loaded, run and finished, it will clear the screen, tell you it's done and then execute a NEW command. The NEW command lies in Line 500. If you wish to examine the program, just don't type RUN first but delete Line 500 and then you can play around with the program. Why the NEW? Well, it's simple. When I load it up I want it running and load up with the commands. Although in the future I will just be loading one tape to do all this, for now and for this magazine, this is the way I thought would be best. Now that is done and you are ready to use your five new commands.

At this time, I recommend loading in the Demo program which follows the CMD overlay. This will illustrate for you the real power you have in your BASIC now with the new commands. I took a variety of examples. Though they may not be the best, they do demonstrate everything you can do with the commands including at the end I might say one dog gone impressive pie circle graph. I have seen this on other computers, but never on the Interact. It is to my knowledge the first pie circle graph done on the Interact but now you have it for your use.

Now that you have run both programs, I shall go through each new command, what it does, and how it does it.

NOTE: There are a couple of changes worth noting before you learn the commands. The first thing you will notice is that the colors used for BASIC are now 0, 1, 2, 3 which is yellow on black with red and green. The reason I changed it is I have read recent studies that say yellow is the easiest color on the eyes and the black background prevents glare.

Another change is that we may now bid farewell to our olf friend POKE 19215, 25. As in the newer versions of BASIC, the overlay takes care of the problem for you so there is no more need for the initial poke.

The other difference is in my reference to the Y axis on the screen. In all of my work with Rom 1 and machine code, 1 on the Y axis is at the upper left of the screen and 77 at the bottom, thus coordinate 1, 77 would be at the lower left. If it is difficult for you to reverse this thinking process, simply enter your X coordinate, then for your Y coordinate, say 77YA where YA is the number for Y in the BASIC way of like if you wanted 67 for the Y then 77-67 = 10 which would be your Y coordinate for Machine Language or for theese command routines.

It is also important to note that there is no protection on the Y axis for going beyond the screen memory into other RAM If you do not give a Y coordinate beyond 77, you should not have a problem with it. When I developed my first overlay I did not have the memory to change that. Now we might, but I did not take the time to put in that protection.

CMDB makes boxes, squares, rectangles etc. It takes the following form: CMDBXS,YS,XL,YL,C whereby XS is the starting X coordinate, YS is the starting Y coordinate, XL is the length of the X sides of the box and YL is the length of the Y sides of the box. C is for the color. This color can be any of the four colors you wish. This is exactly what the extended Plot statement in FAST GRAPHICS BASIC does.

CMDL enables you to draw a line from any X-Y point to any X-Y point in any of the four colors. It takes the form of:
CMDLXS,YS,XE,YE,C where XS is the X starting coordinate, YS is the starting Y coordinate, XE is the X ending coordinate, YE is the Y ending coordinate, and C is the color of the line.

CMDT will make a Triangle in any of the four colors given any three X-Y points. It takes the form of:
CMDTX1,Y1,X2,Y2,X3,Y3,C whereby X1, Y1, X2, Y2, X3, Y3 represent three points where the triangle is drawn. Please note that because it must fill in every point in those coordinates it does take a little longer than the other subroutines.

CMDP is to changing the Print Color. It takes the form of:
CMDP# whereby the # may be 0, 1, 2 or 3 which is the color you want to print. You may change this at any time before or after a Print statement and is designed to save you much time and memory over using an Output statement. You may simply want to change the color of your text and keep outputting print.
CMDF This command will fill in any bordered area you wish in any color you wish 0, 1, 2 or 3. I originally saw this on the fast graphics cartridge for the VIC-20 where it is called a PAINT command. I thought it was unique how you could make a shape and color it in. You can see when you're doing pictures how important this can be. You can make boxes and triangles without filling them in in this way but what if you want an upside down diagonal U shape? How are you going to fill in that shape without outputting every X and Y coordinate within that shape? It is obviously a time-consuming effort not to mention the RAM it would take. with the CMDF command you can do that. There are some exceptions, which I shall discuss. The command takes the form of:
CMDFXS,YS,C,BO where XS and YS are the starting coordinates, C is the color and BO is the border color. The YS point should be the highest point in the Y axis in your image as the the Fill command works from left to right, right to left, and downward. As soon as it bumps into a border color on the Down routine, it will stop. Assume you have an upside down V in the middle of the screen taking the whole screen. You simply tell it the X and Y point in the center top of that V and the

the color you wish to fill the V in with, and the border color of that V. It will instantly go down that screen filling up that image. However, if you now have a big V right-side-up you're in trouble. Every time it goes down a notch in the Y axis it bumps into a border color and thinks it's all done! For that, refer to the Demo program again and you'll see how I get out of that. You will notice in my demo that by using a FOR loop to increase the Y down the screen you can then make any Fill you wish. Now you can get any shape but it is a bit slower than without the FOR LOOP it will fill in anything you need to fill in.
This concludes the brief explanation. The following chart is a brief recap for your reference.

CMDBXS,YS,XL,YL,C

CMDFXS,YS,C,BO

CMDLXS,YS,XE,YE,C

CMDP# (0, 1, 2 or 3)

CMDTX1,Y1,X2,Y2,X3,Y3,C

When I described my original overlay at the beginning of this story, I mentioned one other command which you will notice is left out of this the CMD overlay for new commands. I called this command SET. and it was my proudest achievement of them all. However, you will find, as I did, that the Interact had this command all along. I thought this command was going to make history and was thinking of contacting the companies who design and put out BASIC languages about this revolutionary discovery and what a great help it would be. In Interactors Input in this issue you will read a very interesting and informative letter from Mr. Warren J. Moore. How he ever knew I was working on such a command I'll never know. I urge you to read Mr. Moore's letter and learn about a command that you never knew your BASIC had. Or at least I never knew. People with whom I talk are still shocked about it. It is not mentioned in any of the BASIC books of Interact or Micro Video. In fact, in Basically Speaking, which is an excellent guide to have they say to use a dummy Read loop to increment your Data counter. Now, we find out, all you have to do is say the command RESTORE Line Number. Quite an oversight! I honestly don't think anyone knew about it. But now, through the pages of INTERACTION INTERNATIONAL, you know about it!

I hope you will like theese five new commands. There is no reason unless you are using CSAVE* or CLOAD* why you should not load up and use them right away. I will be using them from now on. I think they are a necessity in theese days of graphics and colors and shapes.
By changing the formalas in the Demo it is possible to make pentagons, hexagons, octagons, and any gone you want! Anything beyond twelve sides comes out looking somewhat like a circle however. I hope you will benefit from them as much as I have enjoyed bringing them to you and that they will take the place of any other graphics overlays you may have. After all, it didn't cost you $19.95! It is a service to you from your Editor and from INTERACTION INTERNATIONAL! Thank you.

```
1 REM NEW BASIC COMMANDS NAME"NBC"
5 CLS:PRINT"THIS PROGRAM.      TAKES TIME TO RUN.DO NOT WORRY WHEN IT IS DONE
"
7 PRINT"IT WILL TELL YOU."
10 POKE19215,25
20 FORCT=25127TO25135:READHD$:GOSUB64000:POKECT,B:NEXT
30 DATA31,8D,4C,CD,FA,66,C3,0C,66
40 POKE24928,0:POKE24929,1:POKE24930,2:POKE24931,3
50 POKE19199,3
100 CLS:COLOR3,4,2,1:PRINT:PRINT"NOW PROGRAMMING  PLEASE WAIT"
110 FORCT=18953TO19187:READHD$:GOSUB64000:POKECT,B:NEXT
120 DATACD,13,4A,CD,28,4A,CD,13,4A,C9,2A,F9,4A,EB,2A,FB,4A,22,F9,4A,2A,FD
122 DATA4A,22,FB,4A
130 DATAEB,22,FD,4A,C9,3A,F4,4A,CD,2F,06,2A,FB,4A,22,F5,4A,2A,FD,4A,22,F7
140 DATA4A,21,44,4A,22,DC,4A,CD,77,4A,C9,2A,F9,4A,22,F5,4A,6A,63,22,F7,4A
150 DATA21,DF,04,22,DC,4A,CD,77,4A,2A,FB,4A,22,F5,4A,2A,FD,4A,22,F7,4A
160 DATA21,44,4A,22,DC,4A,C9,3A,F4,4A,CD,2F,06,21,DF,04,22,DC,4A,AF,57,5F
170 DATA6F,3A,F5,4A,47,3A,F7,4A,90,F28A,4A,2F,3C,2E,01,47,3A,F6,4A,4F
180 DATA3A,F8,4A,91,F2,9E,4A,2F,3C,67,7D,F6,02,6F,7C,4F,90,FA,AA,4A,78,41
190 DATA4F,7D,F6,04,6F,78,1F,E6,7F,2F,3C,67,E5,D5,C5,3A,F5,4A,47,3A,F6,4A
192 DATA4F,7D,E6,04
200 DATACA,C5,4A,7A,53,5F,7D,E6,02,7B,CA,CE,4A,2F,3C,81,5F,7D,E6,01,7A,CA
210 DATAD9,4A,2F,3C,80,57,CD,DF,04,C1,D1,E1,14,78,92,F8,7C,81,67,CA,B1,4A
220 DATAFA,B1,4A,90,67,1C,C3,B1,4A
225 CLS:PRINT"STILL PROGRAMMING"
230 FORCT=19200TO19211:READHD$:GOSUB64000:POKECT,B:NEXT
240 DATACD,28,4A,CD,13,4A,CD,28,4A,C3,09,4A
250 POKE25433,195:POKE25434,77:POKE25435,68
260 POKE25704,48:POKE25705,98
270 FORCT=25136TO25346:READHD$:GOSUB64000:POKECT,B:NEXT
280 DATAFE,42,CA,56,62,FE,46,CA,84,62,FE,4C,CA,CE,62,FE,50,CA,F9,62,FE,54
290 DATACA,D9,64,FE,96,CA,35,68,FE,96,CA,35,68,C3,35,68
310 DATA23,CD,BF,75,32,F8,4A,CD,92,77,32,F7,4A,CD,92,77,32,F5,4A,CD,92,77
320 DATA32,F4,4A,CD,92,77,32,F6,4A,E5,01,F4,4A,CD,A2,05,E1,3A,FF,4A,CD,2F
330 DATA06,C9,23,CD,BF,75,57,CD,92,77,5F,CD,92,77,4F,C5,CD,92,77,C1,47,E5
340 DATACD,AE,62,15,CD,C0,62,14,1C,C5,D5,C1,CD,10,06,C1,B8,C2,93,62,E1,C9
350 DATACD,00,06,14,C5,D5,C1,CD,10,06,C5,D1,C1,B8,C2,AE,62,C9,3E,15,32,B1
360 DATA62,CD,AE,62,3E,14,32,B1,62,C9,23,CD,BF,75,32,F5,4A,CD,92,77,32,F6
370 DATA4A,CD,92,77,32,F7,4A,CD,92,77,32,F8,4A,CD,92,77,32,F4,4A,E5,CD,6B
380 DATA4A,E1,3A,FF,4A,CD,2F,06,C9,23,CD,BF,75,32,FF,4A,C3,F2,62
390 CLS:PRINT"I AM ALMOST DONE."
400 FORCT=25817TO25866:READHD$:GOSUB64000:POKECT,B:NEXT
410 DATA23,CD,BF,75,32,F9,4A,CD,92,77,32,FA,4A,CD,92,77,32,FB,4A,CD,92,77
420 DATA32,FC,4A,CD,92,77,32,FD,4A,CD,92,77,32,FE,4A,CD,92,77,32,F4,4A
430 DATAE5,CD,00,4B,C3,F1,62
450 CLS:COLOR0,1,2,3
499 NEW
999 END
64000 B=0:C=16:FORX=1TO2:B$=MID$(HD$,X,1):IFB$>"9"THENGOSUB64100
64010 B=VAL(B$)*C+B:C=C/16:NEXT:RETURN
64100 B$=STR$(ASC(B$)-55)
64110 RETURN
OK
```

```
1 REM NEW BASIC COMMAND OVERLAY DEMO
2 REM GEORGE A. LEGGETT 20562 WOODWARD MI. 48043
3 REM MARCH 31, 1983
5 CLS: COLOR0, 1, 2, 3: PRINT"WHEN YOU ARE THRUVIEWING EACH      SCREEN PRESS
ANY"
7 PRINT"KEY TO CONTINUE.": I$=INSTR$(1)
10 CLS: PRINT"WITH THE NEW CMD COMANDS YOU CAN  DO MANY GRAPHIC  PROGRAMS
 WITH
20 PRINT"LEVEL II BASIC.  THE CMDB COMMAND ALLOWS YOU TO     MAKE BOXES."
30 I$=INSTR$(1): CLS
40 CMDB10, 30, 50, 10, 1: CMDB70, 40, 15, 20, 2
50 CMDB20, 60, 30, 20, 3: I$=INSTR$(1)
60 CLS: PRINT"WITH THE CMDL     COMMAND YOU CAN  MAKE A LINE       BETWEEN
ANY 2
70 PRINT"X-Y POINTS ON THESCREEN IN ANY     COLOR YOU WANT": I$=INSTR$(1)
75 CLS: COLOR7, 0, 1, 4
80 FORCT=1TO100: GOSUB1000
90 CMDL56, 38, X, Y, C: NEXT: I$=INSTR$(1)
100 CLS: COLOR0, 4, 3, 7: XS=1: YS=1: FORCT=1TO100
110 GOSUB1000: CMDLXS, YS, X, Y, C: XS=X: YS=Y: NEXT: I$=INSTR$(1): CLS
120 COLOR3, 1, 2, 4: CMDP3: PRINT"WITH  THE CMDP": CMDP1: PRINT"COMMAND YOU CAN"
: CMDP2
130 PRINT"CHANGE THE PRINT": CMDP3: PRINT"COLOR WHENEVER": CMDP1: PRINT"YOU
WANT WITHOUT"
140 CMDP2: PRINT"USING THE OUTPUT": CMDP3: PRINT"COMMAND TO DO IT.": I$=INST
R$(1): CLS
150 COLOR2, 0, 3, 7: PRINT"WITH THE CMDT     COMMAND YOU CAN  MAKE FILLED IN
"
160 PRINT"TRIANGLES OF ANY SIZE IN ANY COLOR": I$=INSTR$(1): CLS
170 CMDT10, 10, 20, 10, 15, 20, 1: CMDT20, 60, 60, 20, 100, 50, 2
180 CMDT80, 20, 105, 15, 90, 40, 3: I$=INSTR$(1): CLS
190 COLOR6, 4, 3, 0: PRINT"BY USING THE      FOLLOWING ROUTINEEARLY IN YOUR
"
200 PRINT"PROGRAM TO STORE 12 X-Y VALUES YOUCAN THEN MAKE A  CIRCLE OUT
OF 12
210 PRINT"TRIANGLES": I$=INSTR$(1): CLS
220 DIMX(12), Y(12): RD=57.2958: A=0: FORCT=0TO360STEP30: Y(A)=SIN((CT+.1)/RD
)
230 X(A)=COS((CT+.01)/RD): A=A+1: NEXT
240 CLS: PRINT"WHEN YOU WANT TO MAKE A CIRCLE     SIMPLY USE THE   FOLLOWI
NG"
250 PRINT"FORMULA:  WHERE    R=RADIUS OF YOUR CIRCLE AND XA=    X AXIS OF C
ENTER
260 PRINT"YA=Y AXIS OF THE CIRCLE CENTER     C=COLOR THEN      GOSUB 2000"
265 I$=INSTR$(1): CLS
270 R=30: XA=56: YA=38: C=1: GOSUB2000: I$=INSTR$(1): CLS
280 R=10: XA=15: YA=15: C=2: GOSUB2000: R=25: XA=80: YA=40: C=3: GOSUB2000: C=1
290 R=15: XA=30: YA=45: GOSUB2000: I$=INSTR$(1): CLS
```

```
300 COLOR1,0,3,7:PRINT"IF YOU WANT TO   MAKE AN UNFILLED CIRCLE SIMPLY U
SE"
310 PRINT"THE SUBROUTINE AT3000 TO ACHIEVE  THE RESULTS. USE "
320 PRINT"THE SAME          VARIABLES AS YOU DID FOR THE      FILLED CIRC
LE."
330 I$=INSTR$(1):CLS:R=10:XA=15:YA=15:C=3:GOSUB3000:C=2:R=25:XA=75:YA=40
340 GOSUB3000:R=20:XA=20:YA=45:C=1:GOSUB3000:I$=INSTR$(1):CLS
350 COLOR0,1,4,3:PRINT"WITH THE CMDF     COMMAND YOU CAN  FILL IN MOST "
360 PRINT"SHAPES THAT YOU  HAVE. ":I$=INSTR$(1):CLS
370 R=30:XA=56:YA=38:C=3:GOSUB3000
380 CMDF56,9,1,3:FORY=38TO66:CMDF56,Y,2,3:NEXT:I$=INSTR$(1)
390 CMDL56,38,56,8,3:CMDL56,38,86,38,3
400 FORY=9TO36:CMDF57,Y,3,3:NEXT:I$=INSTR$(1)
410 CMDL56,38,30,50,3:CMDF54,9,0,3
420 FORY=38TO49:CMDF31,Y,0,3:NEXT
430 CMDL56,38,74,59,3:FORY=39TO58:CMDF74,Y,1,3:NEXT:I$=INSTR$(1):CLS
440 PRINT"I HOPE YOU WILL  MAKE THIS OVERLAYAPART OF YOUR      BASIC LANGU
AGE."
450 PRINT:PRINT"I  BELIEVE NOW WE    TRULY HAVE A":CMDP2:PRINT"  FAST GRAP
HICS"
460 CMDP1:PRINT"          BASIC":FORCT=1TO3000:NEXT:FORCT=1TO12:PRINTCHR$(7)
:NEXT
470 CLS
999 END
1000 X=INT(112*RND(1)):Y=INT(76*RND(1)):C=INT(3*RND(1)+1):RETURN
2000 FORCT=0TO11:X=X(CT)*R+XA:Y=Y(CT)*R+YA:X1=X(CT+1)*R+XA
2010 Y1=Y(CT+1)*R+YA:CMDTXA,YA,X,Y,X1,Y1,C:NEXT:RETURN
3000 FORCT=0TO11:X=X(CT)*R+XA:Y=Y(CT)*R+YA:X1=X(CT+1)*R+XA
3010 Y1=Y(CT+1)*R+YA:CMDLX,Y,X1,Y1,C:NEXT:RETURN
OK
```

ADJUSTMENTS FOR Vol. IV no. 2

There are three small mistakes that we know of in the magazine. We appologize for theese oversights and thank John Peters of Wayland, MI for bringing the first two errors to our attention.

In the DECK OF CARDS routine, Page 12, Line 46 should read:

```
46DATAAC,2C,3C,4C,5C,6C,7C,8C,9C,JC,QC,KC
```

This line should also be corrected in all other card games but CLOCK PATIENCE.

On page 15 in the STUD POKER game, Line 600 at the end says GOTO 2080. In some magazines it appears as 2030, but it is Two Thousand Eighty. This is the fault of the Teletype printer which makes the listings.

On Page 30 in the discussion of the Joy Stick input, the top row is labeled correctly as 1, 2, 3, 4, 5. However, the bottom row should read from left to right: 9, 8, 7, 6.

We do try our best, but the problem is, we are human, not computers!

MARCH 12, 1983

Mr. John H. Peters
774-135th St. Lot 46
Wayland, Mi. 49348
(616) 792-9825

Mr. George A. Leggett
20562 Woodward
Mt. Clemens, Mi. 48043

RE: MICRO VIDEO KEYBOARD INSTALLATION INSTRUCTIONS

Sir;

The professional keyboard has proved to be an excellent improvement over the old one.

After reading the instructions however, I found that step #6 was unnecessary.

I didn't drop the keyboard into place, I angled it into place, without removing the five keys.

To unsnap the keycaps just didn't sound right and in my case it proved to be correct.

I sent a letter to Mr. Ross of MICRO VIDEO also, concerning this matter.

To avoid damage to any keycaps, I would advise not to unsnap them from the keyboard, if at all possible.

John H Peters

Mr. John H. Peters

EDITOR'S NOTE: Thank you, Mr. Peters, for your letter. I would like to add to it. I felt exactly as you did when I got my first keyboard and it worked exactly that way. I did not remove the keys as Micro Video recommended. Over five years ago I purchased a keyboard and knew the hassles of removing keys and the dangers of bending contacts. So I did not remove the keys and it went in perfectly. But wait! When I ordered my second keyboard for my new 32K machine, this technique did not work at all. I struggled and pushed and pulled for over a half hour to no avail. I had no choice but to take off the keys. After closer examination I found that my 16K which was a Model 1 8K originally way back when they first came out and it has a bigger opening. Thus, it worked in the older machine and not in the new 32K machine. So, my advice is this: Try it without removing the keys. If it won't go in within two minutes, remove the keys. But do it gently!

March 5, 1983

W.J. Moore
571 So. Broadway
Pittsburg, Ca. 94565

Interaction International
George A. Leggett
20562 Woodward
Mt. Clemens, Mi. 48043

I just received my copy of INTERACTION INTERNATIONAL Vol.IV No.1. It is well organized and very interesting to read. There is no doubt that a substantial amount of information will be presented in future issues. To this end, I have enclosed an old mailing list purchased from Steve Cook. You might want to send out some kind of flyer of change in publishers. I got my information from another Interactor, Al Jackson - Simi Valley, Ca.

I previously sent you a program with listing on Feb 18, 1983 which has a program error (what program doesn't?). The program name is SCREEN GRAPHICS CALCULATOR and the error is in LINE-630 (H=1) should read (H=0) therefor the corrected should read:

```
630 IF AR='D' THEN H=0: V=V-1: GOSUB 300: GOSUB 310
```

In your MACHINE SHOP TALK on page-16 last paragraph, you might not be aware of Basic's RESTORE command being able to accept an argument. The argument is LINE number and resets a pointer so that the next READ will start at the LINE.
Example:

```
10 Restore 50
20 Read T$
30 Print T$
40 Data It does not work.
50 Data It does work!
```

This is much faster than using a FOR loop to step up to the N'th data item. It also means data can be placed anywhere as long as data blocks are not on same LINE. Hope this will be passed on to the readers.

I am very thankful for the explanation for PEEKing and POKEing above Basic's limits. I was very frustrated since my Interact is MV 32K and all programs are above 32,767!

Will be looking for next great newsletter so for now. GOOD COMPUTING.

W. Jim Moore

february 18, 1983

W.J. Moore
571 So. Broadway
Pittsburg, Ca. 94565

Mr. George Legget
20562 Woodward
Mt. Clemens, Mich. 48043

SCREEN GRAPHICS CALCULATOR was developed for the purpose of converting a character into its component numerical values and as an aid to developing the image. The program lets you draw your spaceship or monster or whatever that you want to move around the screen. The working area is blown-up from a single pixel to a 3x3 pixel block in order to help with the fine details.

The working area is 32 pixels (4 bytes) across by 17 pixels down. This arrangement provides either 8 pokes per line for programs in BASIC or 4 bytes per line for Rplot routines in machine language. The start of each 4 pixels and the start of each byte is color coded above the working area. There are also (X) and (Y) cursors indicated by (+). All controls are keyboard operated (joystick not used).

there are two modes of operation. The first one permits drawing your image or making changes. The second one calculates numerical (hex) values for each 4 pixels at a time. For poking, this value will be a decimal number. For Rplot, this value will be in hex-decimal. (Remember it takes two hex-decimal numbers for one byte.)

FIRST MODE CONTROLS (FOR DRAWING)
R - move (X) cursor to right
L - move (X) cursor to left
U - move (Y) cursor up
D - move (Y) cursor down
0 - plot at XY in color-0
1 - plot at XY in color-1
2 - plot at XY in color-2
3 - plot at XY in color-3
P - go to PRINT VALUES MODE

SECOND MODE CONTROLS (PRINT VALUES)
R - return to DRAWIND MODE
D - drop one line and reset (X) cursor to left
C - change from POKE to RPLOT or vice-versa all othr keys - continue calculations for each 4 pixels

well there you have it. If your screen character is larger than working area, then do a section at a time but am sure that the area is large enough for most purposes. Of course this program only aids in defining your image and not where it will be displayed. Pokes to the screen control the colors of each pixel. Rplot routine defines character shape in one color only so it makes no difference which color you use in this mode, only if pixel is turned on or off counts. If I am wrong about the use of Rplot, please comment.

GOOD GRAPHICS

```
10 REM (SCREEN GRAPHICS CALCULATOR) 8K BY W.J. MOORE
20 DIMP(32,17):Z=1:COLOR0,1,2,7
25 H$="0123456789ABCDEF"
30 CLS:OUTPUT"P-POKE SCREEN",6,59,1
40 OUTPUT"R-RPLOT (BINARY)",6,47,1
50 A$=INSTR$(1):IFA$="R"THENZ=2
60 CLS:FORI=11TO104STEP3:PLOTI,71,3:NEXT
70 FORI=11TO104STEP24:PLOTI,71,2:NEXT
80 FORI=23TO104STEP24:PLOTI,71,1:NEXT
90 PLOT8,11,3,1,54:PLOT8,65,3,100,1
100 PLOT107,11,3,1,54:PLOT8,11,3,99,1
110 H=1:GOSUB300:V=17:GOSUB310
120 E$=INSTR$(1)
130 IFE$="R"THEN200
140 IFE$="L"THEN220
150 IFE$="U"THEN240
160 IFE$="D"THEN260
170 IFE$="P"THEN350
180 IFE$>"0"ANDE$<"4"THEN320
190 GOTO120
200 IFH+1>32THEN120
210 H=H+1:GOSUB300:GOTO120
220 IFH-1<1THEN120
230 H=H-1:GOSUB300:GOTO120
240 IFV+1>17THEN120
250 V=V+1:GOSUB310:GOTO120
260 IFV-1<1THEN120
270 V=V-1:GOSUB310:GOTO120
300 PLOT6,67,0,100,3:OUTPUT"+",H*3+6,70,3:RETURN
310 PLOT4,13,0,3,52:OUTPUT"+",3,V*3+13,3:RETURN
320 P(H,V)=VAL(E$):OUTPUT"+",H*3+6,V*3+13,P(H,V)
330 OUTPUT"=",H*3+6,V*3+13,P(H,V):GOTO120
350 H=1:GOSUB300:V=17:GOSUB310:H=0
360 FORI=1TO4:H=H+1
370 IFH>32THENH=1:V=V-1:GOSUB310:IFV=0THEN110
380 IFZ=1THENP=P*4
390 IFZ=2ANDI=1THENP=16
400 IFZ=1ANDI=1THENP=1
410 IFZ=2THENP=P/2
420 IFZ=1THENX=P(H,V)*P:T=T+X:GOSUB300
425 IFZ=2THENX=(P(H,V)>0)*P:T=T-X:GOSUB300
430 NEXT
440 IFZ=2THEN460
450 PLOT20,5,0,80,5:OUTPUTT,32,9,3:GOTO600
460 T$=MID$(H$,T+1,1)
470 PLOT20,5,0,80,5:OUTPUTT$,32,9,3:OUTPUT"HEX",62,9,3
600 A$=INSTR$(1)
610 T=0:P=0:T$=""
620 IFA$="R"THEN110
630 IFA$="D"THENH=1:V=V-1:GOSUB300:GOSUB310
640 IFA$="C"THEN660
650 GOTO360
660 IFZ=1THENZ=2:GOTO680
670 IFZ=2THENZ=1
680 GOTO350
```

MACHINE SHOP TALK

WELCOME TO ANOTHER MACHINE SHOP TALK. IN THIS ISSUE WE HAVE OUR FIRST GUEST CONTRIBUTOR, MR. ALFRED E. JACKSON WHO GAVE USE THE WONDERFUL 8080 LISTINGS. I FOUND THESE WOULD BE A GREAT ASSET TO ANYONE WHO WANTS TO LEARN 8080 MACHINE LANGUAGE. FOR YEARS I USED THE LISTINGS FOUND IN THE 8080A BUGBOOK. BUT THESE NEW LISTINGS WILL TAKE THE PLACE VERY NICELY.

I AM VERY HAPPY TO SAY THAT INTERACTIONS INTERNATIONAL HAS ANOTHER FIRST TO MY KNOWLEDGE IS PRINTING A COMPLETE 8080 LISTING. NOT ONLY IN NUMERIC ORDER BUT ALSO IN ALPHABETIC AND FUNCTION ORDER AS WELL. I BOAST ABOUT THIS SIMPLY BECAUSE THE MACHINE AS BEEN AROUND 5 YEARS AND NO ONE THAT I KNOW OF AS PUBLISHED ONE SIMPLE 8080 INSTRUCTION SET YET IN ANY MAGAZINE OR PUBLICATION UNTIL NOW. WHILE IT MAY BE THE OPINION OF SOME THAT THERE MAY NOT BE ENOUGH INTREST IN THIS SUBJECT I HAVE FOUND THAT TO BE THE FARTHEST THING FROM THE TRUTH BY THE OVERWHELMING RESPONCE IN 8080 AND MY PRODUCTS FOR 8080 MACHINE LANGUAGE. THANK YOU FOR THE SUPPORT. MY THANKS ALSO GO TO MR. DEAN ANSCHULTZ WHO ALSO SUBMITTED A COPY OF THESE LISTINGS.

AT THIS TIME I LIKE TO TELL YOU HOW IMPORTANT THESE LISTINGS ARE TO YOU. AS I AM SURE YOU KNOW BY NOW I AM A 8080 NUT AND THINK IT IS ONE OF THE FINEST LANGAUGES TO WORK IN. I HAVE STUDY 6502, 6510, AND Z80. THE Z80 CAN DO EVERYTHING THE 8080 CAN DO PLUS A WHOLE LOT MORE BY PRECEEDING THE INSTRUCTION WITH A CB OR ED THE INSTRUCTIONS TAKE ON A WHOLE NEW MEANING GIVING YOU AROUND 500 INSTRUCTIONS INCLUDING ALL THE 8080 INSTRUCTIONS. THIS IS A VAST CONTRAST TO THE 6502 OR 6510 WHICH ONLY HAVE 152 INSTRUCTIONS. A TRUELY INFERIOR CPU IF YOU WANT CALL IT A CPU A ALL. NOW I POINT THIS OUT TO YOU BECAUSE THE VIC-20 AND THE COMMODORE 64 HAVE A 6502 AND 6510 RESPECTIVELY AND IF YOU HAVE ANY PLANS TO GET ONE OF THESE MACHINES IN THE FUTURE YOU SHOULD BE AWARE OF THIS FACT. THIS IS WHY I BOUGHT MY VIC-20 IN THE FIRST PLACE SO I COULD LEARN 6502.

I LIKE TO TELL YOU VERY BRIEFLY ON EXAMPLE IN MY EXPERIENCE WITH THE 6502. I WAS TRYING TO WRITE A ROUTINE TO MOVE A CHARACTER AROUND THE SCREEN. BASIC IS JUST TO SLOW FOR SOMETHING LIKE THIS AND BEING THE VIC-20 RUNS AT ONLY 1 MEGACYCLE IT IS SLOWER YET. ANYWAY ONCE I GOT THE ROTINE FINISHED I FOUND IT TOOK 4 TIMES THE AMOUNT OF MEMORY AS DOING IT IN 8080 NOT TO MENTION THE SPEED.

FINALLY I LIKE TO TELL YOU THAT OUT OF ALL THE CPU INSTRUCTION SETS WRITTEN I BELIEVE THE 8080 IS THE EASIEST TO UNDERSTAND. IT WASN'T UNTLI A FEW WEEKS AGO THAT I FULLY UNSTOOD HOW THE Z80 SET LAYED OUT AND THANKS TO ONE OF THE APPENDIX IN THE TIMEX/SINCLAIRE MANUAL THAT I GOT THE CONNECTION. AS I HAVE SAID BEFORE AND IN MY BOOK THAT I WROTE YOU CAN HAVE THE GREATEST IDEA IN THE WORLD BUT IF YOU CAN NOT COMMUNICATE IT TO OTHER PEOPLE IT IS TOTALLY USELESS TO THE PUBLIC. SO I HOPE THAT ALL YOU MAKE GOOD USE WITH THESE LISTINGS AND MANY HOURS OF GREAT 8080 PROGRAMMING.

PLEASE SEND ALL QUESTIONS AND IDEAS THAT YOU WOULD LIKE TO SEE IN MACHINE SHOP TALK TO GEORGE A. LEGGET 20562 WOODWARD MT. CLEMENS, MI. 43043 THANK YOU.

8080 CPU INSTRUCTIONS IN NUMERICAL SEQUENCE

Mnemonic	Hex	Mnemonic	Hex	Mnemonic	Hex	Mnemonic	Hex
NOP	00	MOV B,B	40	ADD B	80	RNZ	C0
LXI B	01	MOV B,C	41	ADD C	81	POP B	C1
STAX B	02	MOV B,D	42	ADD D	82	JNZ	C2
INX B	03	MOV B,E	43	ADD E	83	JMP	C3
INR B	04	MOV B,H	44	ADD H	84	CNZ	C4
DCR B	05	MOV B,L	45	ADD L	85	PUSH B	C5
MVI B	06	MOV B,M	46	ADD M	86	ADI	C6
RLC	07	MOV B,A	47	ADD A	87	RST 0	C7
DAD B	09	MOV C,B	48	ADC B	88	RZ	C8
LDAX B	0A	MOV C,C	49	ADC C	89	RET	C9
DCX B	0B	MOV C,D	4A	ADC D	8A	JZ	CA
INR C	0C	MOV C,E	4B	ADC E	8B	CZ	CC
DCR C	0D	MOV C,H	4C	ADC H	8C	CALL	CD
MVI C	0E	MOV C,L	4D	ADC L	8D	ACI	CE
RRC	0F	MOV C,M	4E	ADC M	8E	RST 1	CF
LXI D	11	MOV C,A	4F	ADC A	8F	RNC	D0
STAX D	12	MOV D,B	50	SUB B	90	POP D	D1
INX D	13	MOV D,C	51	SUB C	91	JNC	D2
INR D	14	MOV D,D	52	SUB D	92	OUT	D3
DCR D	15	MOV D,E	53	SUB E	93	CNC	D4
MVI D	16	MOV D,H	54	SUB H	94	PUSH D	D5
RAL	17	MOV D,L	55	SUB L	95	SUI	D6
DAD D	19	MOV D,M	56	SUB M	96	RST 2	D7
LDAX D	1A	MOV D,A	57	SUB A	97	RC	D8
DCX D	1B	MOV E,B	58	SBB B	98	JC	DA
INR E	1C	MOV E,C	59	SBB C	99	IN	DB
DCR E	1D	MOV E,D	5A	SBB D	9A	CC	DC
MVI E	1E	MOV E,E	5B	SBB E	9B	SBI	DE
RAR	1F	MOV E,H	5C	SBB H	9C	RST 3	DF
LXI H	21	MOV E,L	5D	SBB L	9D	RPO	E0
SHLD	22	MOV E,M	5E	SBB M	9E	POP H	E1
INX H	23	MOV E,A	5F	SBB A	9F	JPO	E2
INR H	24	MOV H,B	60	ANA B	A0	XTHL	E3
DCR H	25	MOV H,C	61	ANA C	A1	CPO	E4
MVI H	26	MOV H,D	62	ANA D	A2	PUSH H	E5
DAA	27	MOV H,E	63	ANA E	A3	ANI	E6
DAD H	29	MOV H,H	64	ANA H	A4	RST 4	E7
LHLD	2A	MOV H,L	65	ANA L	A5	RPE	E8
DCX H	2B	MOV H,M	66	ANA M	A6	PCHL	E9
INR L	2C	MOV H,A	67	ANA A	A7	JPE	EA
DCR L	2D	MOV L,B	68	XRA B	A8	XCHG	EB
MVI L	2E	MOV L,C	69	XRA C	A9	CPE	EC
CMA	2F	MOV L,D	6A	XRA D	AA	XRI	EE
LXI SP	31	MOV L,E	6B	XRA E	AB	RST 5	EF
STA	32	MOV L,H	6C	XRA H	AC	RP	F0
INX SP	33	MOV L,L	6D	XRA L	AD	POP PSW	F1
INR M	34	MOV L,M	6E	XRA M	AE	JP	F2
DCR M	35	MOV L,A	6F	XRA A	AF	DI	F3
MVI M	36	MOV M,B	70	ORA B	B0	CP	F4
STC	37	MOV M,C	71	ORA C	B1	PUSH PSW	F5
DAD SP	39	MOV M,D	72	ORA D	B2	ORI	F6
LDA	3A	MOV M,E	73	ORA E	B3	RST 6	F7
DCX SP	3B	MOV M,H	74	ORA H	B4	RM	F8
INR A	3C	MOV M,L	75	ORA L	B5	SPHL	F9
DCR A	3D	HLT	76	ORA M	B6	JM	FA
MVI A	3E	MOV M,A	77	ORA A	B7	EI	FB
CMC	3F	MOV A,B	78	CMP B	B8	CM	FC
		MOV A,C	79	CMP C	B9	CPI	FE
		MOV A,D	7A	CMP D	BA	RST 7	FF
		MOV A,E	7B	CMP E	BB		
		MOV A,H	7C	CMP H	BC		
		MOV A,L	7D	CMP L	BD		
		MOV A,M	7E	CMP M	BE		
		MOV A,A	7F	CMP A	BF		

8080 CPU INSTRUCTIONS IN ALPHABETICAL SEQUENCE

Mnemonic	Hex	Mnemonic	Hex	Mnemonic	Hex	Mnemonic	Hex	Mnemonic	Hex
ACI	CE	DAD B	09	MOV A,A	7F	MOV L,A	6F	RNC	D0
ADC A	8F	DAD D	19	MOV A,B	78	MOV L,B	68	RNZ	C0
ADC B	88	DAD H	29	MOV A,C	79	MOV L,C	69	RP	F0
ADC C	89	DAD SP	39	MOV A,D	7A	MOV L,D	6A	RPE	E8
ADC D	8A	DCR A	3D	MOV A,E	7B	MOV L,E	6B	RPO	E0
ADC E	8B	DCR B	05	MOV A,H	7C	MOV L,H	6C	RRC	0F
ADC H	8C	DCR C	0D	MOV A,L	7D	MOV L,L	6D	RST 0	C7
ADC L	8D	DCR D	15	MOV A,M	7E	MOV L,M	6E	RST 1	CF
ADC M	8E	DCR E	1D	MOV B,A	47	MOV M,A	77	RST 2	D7
ADD A	87	DCR H	25	MOV B,B	40	MOV M,B	70	RST 3	DF
ADD B	80	DCR L	2D	MOV B,C	41	MOV M,C	71	RST 4	E7
ADD C	81	DCR M	35	MOV B,D	42	MOV M,D	72	RST 5	EF
ADD D	82	DCX B	0B	MOV B,E	43	MOV M,E	73	RST 6	F7
ADD E	83	DCX D	1B	MOV B,H	44	MOV M,H	74	RST 7	FF
ADD H	84	DCX H	2B	MOV B,L	45	MOV M,L	75	RZ	C8
ADD L	85	DCX SP	3B	MOV B,M	46	MVI A	3E	SBB A	9F
ADD M	86	DI	F3	MOV C,A	4F	MVI B	06	SBB B	98
ADI	C6	EI	FB	MOV C,B	48	MVI C	0E	SBB C	99
ANA A	A7	HLT	76	MOV C,C	49	MVI D	16	SBB D	9A
ANA B	A0	IN	DB	MOV C,D	4A	MVI E	1E	SBB E	9B
ANA C	A1	INR A	3C	MOV C,E	4B	MVI H	26	SBB H	9C
ANA D	A2	INR B	04	MOV C,H	4C	MVI L	2E	SBB L	9D
ANA E	A3	INR C	0C	MOV C,L	4D	MVI M	36	SBB M	9E
ANA H	A4	INR D	14	MOV C,M	4E	NOP	00	SBI	DE
ANA L	A5	INR E	1C	MOV D,A	57	ORA A	B7	SHLD	22
ANA M	A6	INR H	24	MOV D,B	50	ORA B	B0	SPHL	F9
ANI	E6	INR L	2C	MOV D,C	51	ORA C	B1	STA	32
CALL	CD	INR M	34	MOV D,D	52	ORA D	B2	STAX B	02
CC	DC	INX B	03	MOV D,E	53	ORA E	B3	STAX D	12
CM	FC	INX D	13	MOV D,H	54	ORA H	B4	STC	37
CMA	2F	INX H	23	MOV D,L	55	ORA L	B5	SUB A	97
CMC	3F	INX SP	33	MOV D,M	56	ORA M	B6	SUB B	90
CMP A	BF	JC	DA	MOV E,A	5F	ORI	F6	SUB C	91
CMP B	B8	JM	FA	MOV E,B	58	OUT	D3	SUB D	92
CMP C	B9	JMP	C3	MOV E,C	59	PCHL	E9	SUB E	93
CMP D	BA	JNC	D2	MOV E,D	5A	POP B	C1	SUB H	94
CMP E	BB	JNZ	C2	MOV E,E	5B	POP D	D1	SUB L	95
CMP H	BC	JP	F2	MOV E,H	5C	POP H	E1	SUB M	96
CMP L	BD	JPE	EA	MOV E,L	5D	POP PSW	F1	SUI	D6
CMP M	BE	JPO	E2	MOV E,M	5E	PUSH B	C5	XCHG	EB
CNC	04	JZ	CA	MOV H,A	67	PUSH D	D5	XRA A	AF
CNZ	C4	LDA	3A	MOV H,B	60	PUSH H	E5	XRA B	A8
CP	F4	LDAX B	0A	MOV H,C	61	PUSH PSW	F5	XRA C	A9
CPE	EC	LDAX D	1A	MOV H,D	62	RAL	17	XRA D	AA
CPI	FE	LHLD	2A	MOV H,E	63	RAR	1F	XRA E	AB
CPO	E4	LXI B	01	MOV H,H	64	RC	D8	XRA H	AC
CZ	CC	LXI D	11	MOV H,L	65	RET	C9	XRA L	AD
DAA	27	LXI H	21	MOV H,M	66	RLC	07	XRA M	AE
		LXI SP	31			RM	F8	XRI	EE
								XTHL	E3

8080 CPU INSTRUCTION SET

	A	B	C	D	E	H	L	(HL) M	Imm.[1]	comment
ADC	8F	88	89	8A	8B	8C	8D	8E	CE	add to A with carry
ADD	87	80	81	82	83	84	85	86	C6	add to A
ANA	A7	A0	A1	A2	A3	A4	A5	A6	E6	AND with A
CMP	BF	B8	B9	BA	BB	BC	BD	BE	FE	compare with A
DCR	3D	05	0D	15	1D	25	2D	35	—	decrement
INR	3C	04	0C	14	1C	24	2C	34	—	increment
MOV A	7F	78	79	7A	7B	7C	7D	7E	3E	move to A
MOV B	47	40	41	42	43	44	45	46	06	move to B
MOV C	4F	48	49	4A	4B	4C	4D	4E	0E	move to C
MOV D	57	50	51	52	53	54	55	56	16	move to D
MOV E	5F	58	59	5A	5B	5C	5D	5E	1E	move to E
MOV H	67	60	61	62	63	64	65	66	26	move to H
MOV L	6F	68	69	6A	6B	6C	6D	6E	2E	move to L
MOV M	77	70	71	72	73	74	75	—	36	move to (HL) address
ORA	B7	B0	B1	B2	B3	B4	B5	B6	F6	OR with A
SBB	9F	98	99	9A	9B	9C	9D	9E	DE	subtract from A w/borrow
SUB	97	90	91	92	93	94	95	96	D6	subtract from A
XRA	AF	A8	A9	AA	AB	AC	AD	AE	EE	exclusive OR with A

	(PSW, A)	(B,C)	(D,E)	(H,L)	SP	comment
DAD	—	09	19	29	39	add pair to HL
DCX	—	0B	1B	2B	3B	decrement pair
INX	—	03	13	23	33	increment pair
LDAX	—	0A	1A	—	—	load A indirect
LXI[2]	—	01	11	21	31	load pair immediate
POP	F1	C1	D1	E1	—	pop pair from stack
PUSH	F5	C5	D5	E5	—	push pair on stack
STAX	—	02	12	—	—	store A indirect

	i= 0	1	2	3	4	5	6	7	comment
RST	C7	CF	D7	0F	E7	EF	F7	FF	restart call to location i*8

	unc	zero/ not zero	carry/ no carry	plus/ minus	even parity/ odd parity	comment
CALL[2]	CD	CC/C4	DC/D4	F4/FC	EC/E4	call subroutine if true
JMP[2]	CB	CA/C2	DA/D2	F2/FA	EA/E2	jump if true
RET	C9	C8/C0	D8/D0	F0/F8	E8/E0	return if true

	code	comment
CMA	2F	complement A (1's)
CMC	3F	complement carry
DAA	27	decimal adjust A
DI	F3	disable interrupts
EI	FB	enable interrupts
HLT	76	halt
IN[1]	DB	input to A
LDA[2]	3A	load A from (n)
LHLD	2A	load HL from (n)
NOP	00	no operation
OUT[1]	D3	output from A
PCHL	E9	jump to (HL)
RAL	17	rotate A left through carry
RAR	1F	rotate A right through carry
RLC	07	rotate A left circular
RRC	0F	rotate A right circular
SHLD[2]	22	store HL in (n)
SPHL	F9	load SP from HL
STA[2]	32	store A in (n)
STC	37	set carry flag
XCHG	EB	exchange (DE) & (HL)
XTHL	E3	exchange (HL) & top of stack

1. 2 byte instruction
2. 3 byte instruction

THE HARD FACTS OF LIFE
BY

GEORGE A. LEGGETT 20562 WOODWARD MT. CLEMENS MICH. 48043

<=><=><=><=><=><=><=><=><=><=><=><=><=><=><=><=><=><=><=><=><=>

INTERACT GETS A MOUTH

IN THIS ISSUE OF HARD FACTS I AM PROUD TO ANNOUNCE THAT WE HAVE OUR FIRST GUEST CONTRIBUTOR FOR HARD FACTS, MR.JERRY GOERZ. MR. GOERZ GIVES US A GREAT HARDWARE PROJECT OF SOME COMPLEXITY. HE HAS NOW GAVE US THE ABILLITY TO MAKE THE INTERACT TALK VIA A RS-232 PORT. I FELT HIS PROJECT WAS WORTH BEING SEEN BY ALL BECAUSE OF THE MOST EXTENSIVE WORK AND WRITE UP DONE ON THE PROJECT OF A SPEECH SYNTHESIZER. BEFORE WE TAKE A LOOK AT THIS I LIKE TO INFORM YOU ALSO ON THE TYPE 'N' TALK SPEECH SYNTHESIZER (TNT) WHICH I OWN AND BEEN USING FOR TWO YEARS NOW.

BREIFLY THE TNT DEFERS FROM MR. GOERZ SYNTHESIZER IN THAT YOU MAY TYPE IN THE WHOLE WORD OR WORDS YOU WISH TO HEAR. THIS HAS A GREAT ADVANTAGE OVER OTHER SYNTHESIZERS IN THE SOFTWARE DEPARTMENT. FOR EXAMPLE IF YOU WERE USING A MICRO VIDEO PORT AND WANTED TO SAY, I AM THE INTERACT, YOU WOULD TYPE:

LPRINTI AM THE INTERACT

YOU MAY USE PHONETICS IF YOU WISH BY I HAVE NEVER FOUND A NEED. FOR EXAMPLE TO SAY THE WORD COMPUTERYOU MUST SPELL IT COMPEUTER SO WITH VERY LITTLE REWRITING YOU WOULD NEVER NEED ANY PHONETICS IN YOUR PROGRAMMING.

HOWEVER THE ABOVE STATEMENTS ARE BY NO MEANS TO PUT DOWN MR. GOERZ SYNTHESIZER. THERE ARE DEFINATE ADVANTAGES TO JERRY'S UNIT. THE BIGGEST OF THESE IS THE COST FACTOR. MY TNT WITH CABLE WAS NEARLY $400 WHILE YOU CAN BUILD THIS SYNTHESIZER FOR $150. ALSO I WOULD NOT WORRY TOO MUCH ABOUT THE DIRECT ENGLISH TRANSLATOR IN THE TNT BECAUSE WITH A LITTLE 8080 PROGRAMMING YOU CAN MAKE YOUR OWN TRANSLATOR. BOTH UNITS WORK ON THE SC-01 SPEECH CHIP.

NO MATTER WHICH UNIT YOU DECIDE TO GET I AM SURE YOUR HAVE GREAT FUN WITH ONE OF THESE UNITS. THE SOFTWARE APPLICATIONS ARE ENORMOUS. IT IS A GREAT ASSET FOR EDUCATIONAL GAME, ESPECIALLY FOR WORD GAMES. WE HAVE TAKEN A FEW INTERACT BASIC GAMES AND SPEECH TO THEM FOR MY WIFE WHO IS BLIND. THIS ALLOWS HER THE LUXURY TO HEAR WHAT IS OUTPUTTED ON THE SCREEN.

A SPEECH SYNTHESIZER MAY NOT BE FOR EVERYONE AND IS STILL IS A VERY EXPENSIVE TOY BUT THIS IS ONE PERSON THAT WAITED OVER 15 YEARS FOR THIS TO BECOME A REALITY. WELL I GUESS THAT'S MY SCIENCE FICTION UPBRINGING THAT EVERY MACHINE COULD TALK, WELL NOW THEY CAN!!! MY NEXT DREAM IS FOR A SPEECH RECONIZER. WHILE THERE ARE A FEW ON THE MARKET I HAVE NOT SEEN ONE HOOKWD UP TO THE INTERACT YET BUT WHO KNOWS MAYBE SOMEONE OUT THERE CAN COME UP WITH ONE IN THE NEAR FUTURE. I DO HOPE SO. AS MY MOTHER HAS SAID MANY MANY TIMES EVERYTHING IN TIME I JUST HOPE IT DON'T TAKE ANOTHER 15 YEARS.

NOW U SHALL TURN YOU OVER TO THE VERY COMPETENT HANDS OF MR. JERRY GOERZ. AND THANKS AGAIN JERRY KEEP UP THE GREAT WORK!

PRODUCT REVIEW: Netronics Speak-Easy

By Jerry Goerz, 2394 Larochelle, Lexington, KY 40504

The Speak-Easy, described in the December 1982 Computers and Electronics, is an easily-assembled speech synthesizer kit at an affordable price ($149.95 + $3.00 postage and handling). It is very versatile, accepting serial, parallel, ASCII or binary inputs, and can operate in a stand-alone mode with binary codes input with eight toggle switches and a pushbutton. Text is easily entered from any computer equipped with a printer port, by using print statements in BASIC, or by using a word processor program capable of outputting printer control codes. Entire canned words may be called from the Speak-Easy's EPROM, or any word in any language may be constructed with phonemes.

The Speak-Easy kit is sold by Netronics, 333 Litchfield Rd., New Milford, CT 06776. Your money buys a PC board, parts, and instructions almost identical to the magazine article. You must provide a cabinet, AC transformer, switches and hardware. The transformer specified is rated at 8.5 volts, 200 milliamps, but with minor modifications you can substitute a 12-volt filament transformer. I squeezed my synthesizer into a slope-front cabinet with top-mounted toggle switches and pitch and volume controls. A DIP-switch, used to reconfigure the inputs, may be mounted facing rearward so that it may be accessed through a notch in the cabinet. I added the following parts, all from Radio Shack:

qty. 1: 270-265 slope-front cabinet, $8.49
qty. 2: 271-1721 10-K potentiometer, $1.09 each
qty. 1: 274-414 pkg. of 2 knobs, $1.39
qty. 8: 275-324 SPST toggle switch, $1.99 each
qty. 1: 275-1549 SPDT N.O. push switch (SPST will do), $2.29
qty. 1: 270-739 pkg. of 2 clip-in fuseholders (only one needed), $0.79
qty. 1: 270-1270 pkg of 3 1/4 amp fuse (only one needed), $0.69
qty. 1: 278-1255 AC line cord, $0.79
qty. 1: 273-1385 12-v 300-ma transformer, $3.29
qty.1 : 272-1028 100-uf 35-v PC-mount electrolytic capacitor, $0.79
Also, you need a metal nibbling tool such as 64-823, $9.49

The construction instructions are clear and accurate. However, if you duplicate my construction, keep these points in mind:

a) Save the slide switch for cabinet-mounting, and skip the power input jack. Wire the switch in series with the AC line cord and fuse.

b) Mount the DIP-switch at a 90-degree angle to face the rear of the PC board, using short lengths of stiff wire to connect the top row of switch pins to the PC board.

c) Substitute the 35-volt capacitor at C9, because 26.7 volts will be present at C9 if a 12-volt transformer is used. C9 filters a voltage-doubling circuit.

d) Mount regulator VR3 with heat-sink compound, as it will get warm using a 12-volt transformer.

e) Install a small-value (.003 uf or so) capacitor from the junction of R38-R39 to ground. This will eliminate oscillations which otherwise occur when relocating the pitch and volume controls.

f) Insert a 1.5 K-ohm resistor between the high side of R38 and the +9-volt supply. This will allow the pitch control to be useful over its entire range. Some experimentation may be required.

I am driving the Speak-Easy with an Interact computer equipped with an RAF IOS/2.3 operating system and the Sabre ML4A printer port. My synthesizer must therefore be configured for the computer-driven serial ASCII mode. At this point, both the instructions and the magazine article become confusing. I will attempt to overcome the confusion:

There are eight DIP-switches at the rear of the board. Let us assume that you have oriented the switch assembly so that the switch nearest the DB25 connector P2 is switch number 1. This switch controls the BUSY function, and must be closed to let the synthesizer tell the computer when it is ready to accept text. The next switch, with the number 2 stamped on it, controls pin 3 of the DB25 connector. Switch 3 controls pin 2, naturally! Whichever pin you are using needs to have a closed switch. I close them both.

Switches 4 and 5 both have the designation C-bar, meaning negative clock. Of these, only switch 4 must be open, and switch 5 doesn't seem to matter. Switch 6, labeled B, selects binary or ASCII, and should be open for ASCII, but mine works either way! Switch 7, labeled P, selects parallel or serial, and must be open for serial data. Switch 8, designated C, meaning clock, should be open, since clock pulses will not be input through DIP-header P1. For simplicity: the first three DIP-switches should be closed, and the rest either must or may be open, for connection to a serial ASBII port.

Now, an important observation: when you turn the Speak-Easy on in the serial ASCII configuration, the data pin (2 or 3) **must** be negative with respect to the ground pin (7), or else the synthesizer will button its lip! With my Interact, the data pin is negative only when the computer is turned on and the IOS/2.3 operating system is not being used. The data pin will stay at about -4 volts while loading and running programs, until data is output to the port, at which time the data pin carries +5 volt pulses of data on a -4 volt baseline. When the computer is off, the data pin is at 0 volts, and when the IOS logo is showing the data pin is at +5 volts, and if the Speak-Easy is turned on under these conditions, you cannot later coax speech out of it, as it apparently fails to recognize the subsequent carraige return as the first symbol sent to it. The Speak-Easy expects a CR as the first symbol, and uses it to determine the baud rate in the serial ASCII configuration. In fact, I've found that two initial CR's are needed to reliably set the baud rate.

Now, another important observation: if you turn the Speak-Easy off while it is connected to an Interact computer which has a program in memory, the program has a better-than-even chance of being damaged. A noise pulse exits the Speak-Easy on the data pin as it is turned off. You can see the noise pulse zip across your TV screen. You can watch the program bomb. So unplug the DB25 connector before turning the Speak-Easy off, if you want to protect the program. To turn the Speak-Easy back on, first reconnect the DB25, or else the synthesizer won't see a negative data line and will not speak. Now, the catch: there is a slight chance that the program will bomb when the Speak/Easy is turned back on! Well, you must admit, the price is reasonable!

If some genius will figure out how to solve the turn-on and turn-off problems, **please** send your solutions to INTERACTION! Perhaps an opto-isolator on the data line...

Let's suppose you have turned the computer on and loaded BASIC and (if necessary) a printer overlay. Let's suppose you then properly set the DIP-switches, plug in the cables, and turn the Speak-Easy on, and that your program continues to run. Now, with a couple of empty print statements, send carraige-returns out the port so the synthesizer can determine the baud rate. Then, print a CHR$(1) to send a control-A out the port, which instructs the synthesizer to begin loading data. Next, select one of four pitch controls by printing a CHR$(35,36,37, or 38). You may substitute $ for CHR$(36), or % for CHR$(37), but the Interact has no keys to substitute for the CHR$(35 or 38). Finally, output the phrase you wish to hear spoken, constructing it from alphameric phoneme codes from the phoneme table (with a space between each code) and/or hex codes (from 00 to 7E) from the ASCII column of the word table (spaces are not needed between word codes). End the phrase with a space and a period. When the Speak-Easy sees a number from zero through seven coming down the cable, it thinks "aha! That's not a phoneme, it's the first digit of a two-digit code calling a canned word from my EPROM." When the Speak-Easy sees a period, it thinks "stop listening, tell the computer to shut up, and start talking!" Then, when it is finished talking, it thinks "tell the computer that it can talk again, and wait for a control-A (CHR$(1)) to signal the start of another phrase."

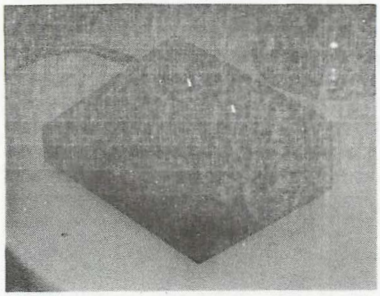
An example BASIC speech program will follow this review. The POKE statement is needed to enable the SABRE port. If you have other ports, omit the POKE and substitute LPRINT for PRINT. Notice the use of the H phoneme following words which end in "fricative stops" such as K or P. The H gives a more natural sound to the speech.

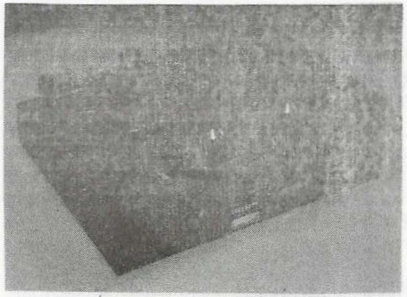
To operate the Speak-Easy as a stand-alone device, the eight toggle switches and the clock pushbutton are used to load data. Close the DIPswitches numbered 4,5,6,and 7, and open switches 1,2,3, and 8 (using my numbering convention). Then turn the Speak-Easy on, and set the toggle switches for hex F0. On my unit, the leftmost switch is Switch 8, representing the most significant (2^7) bit of the 8-bit address. The rightmost switch is Switch 1, representing the least-significant (2^0) bit. A closed switch grounds the respective pin on the DIP-header P1, and represents a logical zero. My switches are closed when flipped down. So to set an F0, I flip the left four switches up (off for logical one) and the right four switches down. Press the clock pushbutton to load the F0. Then, for an example, load "hello" by setting the toggle switches to hex 88 (switches 8 and 4 up, others down) and pressing the clock button. Then, initiate speech by loading a hex F5 (4 and 2 down, others up), and pressing the clock. The Speak-Easy should say "hello."

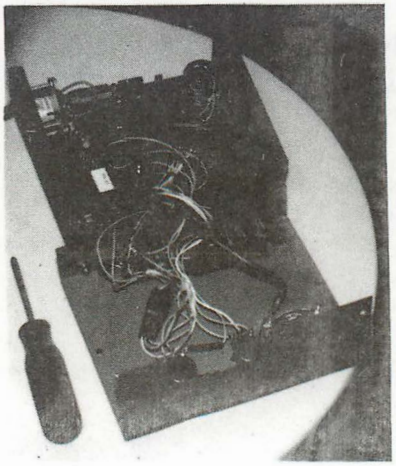
Note that the word table provided with the kit and in the article is incomplete. To complete the binary column, add hex 40 to corresponding entries from the ASCII column. Entries below hex 40 represent phonemes from the phoneme chart. You can program the same phrases in the stand-alone mode that you could in the computer-driven mode, albeit with more difficulty because of the hex-to-binary conversions and switch-flipping.

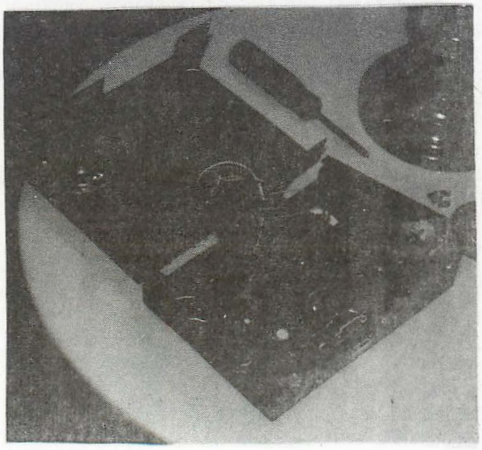
I've added an extra SPDT toggle switch which is wired with four germanium diodes to other switches, and serves to lock in an F5 code regardless of the other toggle positions, preventing accidental erasure of the phrase in memory. A schematic of this modification will follow. Another schematic shows how you might wire two 16-position rotary switches through a diode array to allow hex codes to be entered directly, without the mental conversion to binary. One of you geniuses might show us how to rig up a hex keypad!

I hope this review will help you enjoy your Speak-Easy as much as I'm enjoying mine!





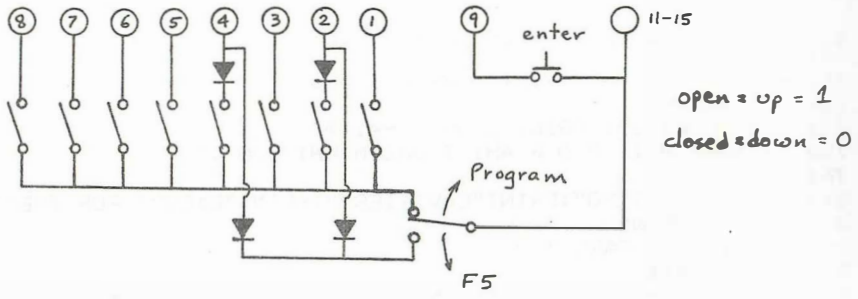


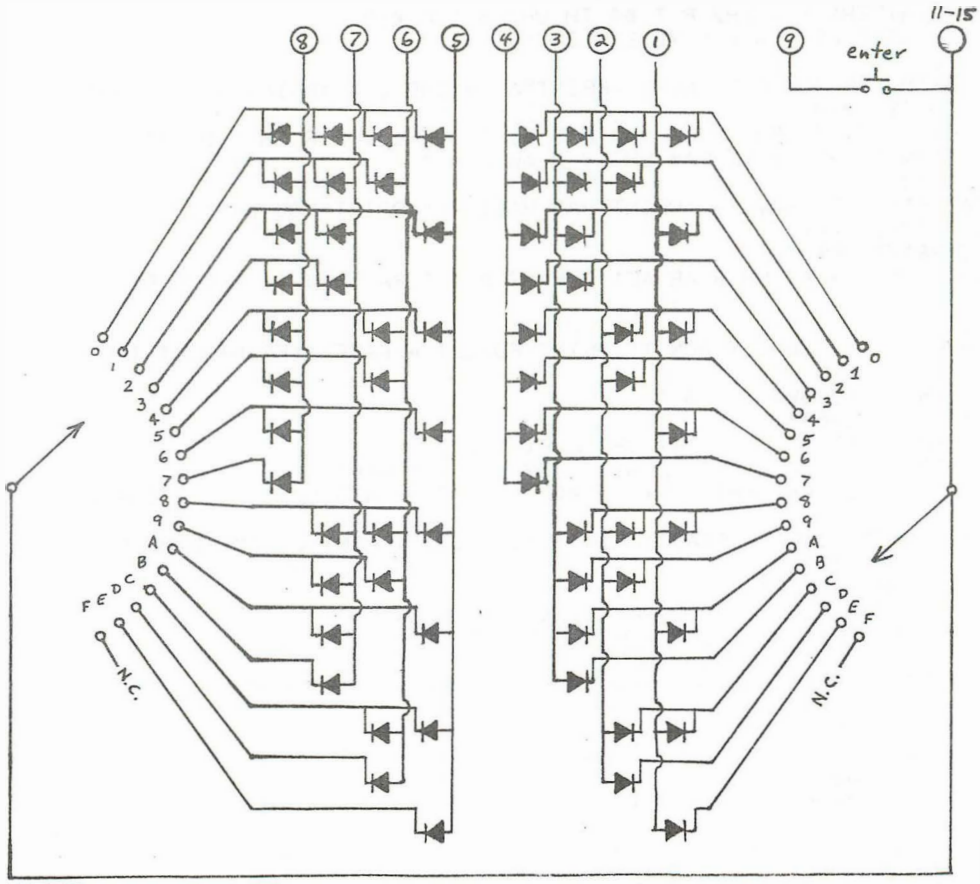


```
5 N=0
9 WINDOW 77: POKE 19541,0
10 CLS:PRINT"  GREEBLER'S":PRINT"  INSULTS":PRINT:PRINT:PRINT:PRI
NT:PRINT:PRINT:PRINT:PRINT
20 WINDOW 54
40 N=N+100
50 FOR T=1 TO 500:NEXT
70 IF N=100 GOTO 100
71 IF N=200 GOTO 200
73 IF N=300 GOTO 300
74 IF N=400 GOTO 400
75 IF N=500 GOTO 500
76 IF N=600 GOTO 600
77 IF N=700 GOTO 700
78 IF N=800 GOTO 800
79 IF N=900 GOTO 900
80 IF N=1000 GOTO 1000
81 IF N=1100 GOTO 1100
82 IF N=1200 GOTO 1200
83 IF N=1300 GOTO 1300
99 GOTO 5
100 PRINT"SMILE, OR ELSE":PRINT"YOUR FACE WILL":PRINT"BREAK!"
110 GOSUB 2000
120 PRINT"S M AH2 I2 L PA1 62 38 7D F A1 AY S H W I L B R A2 AY K
 H ."
130 GOTO 9
200 PRINT"YOUR BREATH COULD":PRINT"STOP A FREIGHT":PRINT"TRAIN"
210 GOSUB 2000
220 PRINTCHR$(38)
230 PRINT"7D B R EH TH K 00 D S T AH1 P A F R AY T PA0 T R A2 AY
N ."
240 GOTO 9
300 PRINT"GO AWAY!":PRINT"YOU STINK!"
310 GOSUB 2000
320 PRINTCHR$(38)
325 PRINT"42 PA1 UH2 W A AY AY PA1"
330 PRINT"Y1 Y1 IU U1 U1 PA0 S S T I I2 NG K H ."
340 GOTO 9
400 PRINT"LET ME HELP":PRINT"YOU OUT -"
405 PRINT"WHICH WAY DID":PRINT"YOU COME IN?"
410 GOSUB 2000
420 PRINT"L EH2 T H M E H EH2 L P 74 63 PA1"
425 PRINT"W I2 T CH W A Y D I1 D 74 K UH M":PRINTCHR$(35):PRINT"I
1 $I2 N ."
430 GOTO 9
500 PRINT"HEY! OLD-TIMER!":PRINT"DID YOU VOTE":PRINT"FOR WOODROW"
:PRINT"WILSON?"
510 GOSUB 2000
520 PRINTCHR$(35)
530 PRINT:PRINT"H A AY PA0 $0 L D T AH2 EH3 E1 M E5 "
535 PRINT"PA1 % D I D 74 V O T H PA0 04 W 001 D R 01 W PA0 W I2 L
 S $UH1 N ."
540 GOTO 9
600 PRINT"I'LL BET YOU":PRINT"WERE THE POSTER":PRINT"CHILD FOR BI
RTH":PRINT"CONTROL!"
610 GOSUB 2000
620 PRINT"AH2 I2 L B EH T H PA0 74 W E5 [illegible]
2 I2 L D "
```

```
630 PRINT"04 B ER TH K UH2 N T R O L ."
640 GOTO 9
700 PRINT"LOOK! A":PRINT"HIPPOPOTAMUS!"
710 GOSUB 2000
720 PRINTCHR$(35):PRINT"L OO K PA1 $"
730 PRINT"A H I1 P O P AH1 T UH2 M UH1 S S ."
740 GOTO 9
800 PRINT"WHAT? NO":PRINT"CAVITIES?":PRINT"EXCEPT FOR THE":PRINT"
BIG ONE BETWEEN"
805 PRINT"YOUR EARS!"
810 GOSUB 2000
820 PRINT"W UH3 $AH2 T PA1 %5B K AE V $I3 T E S %PA1 PA1"
830 PRINT"EH1 K S EH2 P T 04 TH UH2 B I G 01"
835 PRINT"B E1 T W E N 7D E R Z ."
840 GOTO 9
900 PRINT"IF YOU EVER NEED":PRINT"A FRIEND ...":PRINT"BUY A DOG!"
910 GOSUB 2000
920 PRINT"4E 74 EH2 V ER PA0 N E E1 D H EH1 Y F R EH N D PA1"
930 PRINT"B AH1 Y Y1 PA0 EH2 Y D AW G H."
940 GOTO 9
1000 PRINT"YOU HAVE A":PRINT"BAD HABIT -":PRINT"YOU BREATH!"
1010 GOSUB 2000
1015 PRINTCHR$(38)
1020 PRINT"74 47 UH B AE AE1 D H AE B I T PA1 PA0 74 B R E E1 TH
H ."
1030 GOTO 9
1100 PRINT"I USUALLY DON'T":PRINT"FORGET A FACE,":PRINT"BUT I'LL
TRY TO"
1105 PRINT"FORGET YOURS!"
1110 GOSUB 2000
1120 PRINT"4D Y1 U1 ZH U1 UH3 L E1 Y D O N T"
1125 PRINT"F 02 R G EH T UH1 F A Y S H PA1 PA0"
1130 PRINT"2C AH1 EH3 I3 Y L PA0 73 02 F 02 R G EH T Y O R Z H ."
1140 GOTO 9
1200 PRINT"YOU'RE A REAL":PRINT"SMOOTHIE,":PRINT"CHROME-DOME!"
1210 GOSUB 2000
1215 PRINTCHR$(37)
1220 PRINT"7D UH1 $R E L PA0 %S M U TH E PA1"
1225 PRINTCHR$(38):PRINT"K R O M % D O M ."
1230 GOTO 9
1300 PRINT"IF YOU'RE SO":PRINT"SMART, WHY AIN'T":PRINT"YOU RICH?"
1310 GOSUB 2000
1315 PRINTCHR$(37)
1320 PRINT"4E 7D S O S M $AH R T PA1"
1325 PRINTCHR$(38)
1330 PRINT"7B %A N T 74 R $I %T CH ."
1340 GOTO 9
2000 POKE 19541,255:PRINT:PRINT:PRINTCHR$(1)+CHR$(36)
2010 RETURN
```

USING TOGGLE SWITCHES, WITH LOCK-IN SWITCH





USING ROTARY SWITCHES

ADVERTISEMENT

All products described in Vol. IV no. 2 Page 31 are still Available from George A. Leggett, 20562 Woodward, Mt. Clemens, MI 48043.
Back Issues of INTERACTIONS NEWSLETTER are available. The 1981 and 1982 issues remain $2.50 each. For the 1980 issues, send $3.50 for single issues or $19.00 for the entire first year. If you are interested in back issues, contact George A. Leggett.

INTERACTION INTERNATIONAL Programs On Tape: Are you tired of typing in long listings, getting SYNTAX ERRORS because of a bad mistype or have you just simply run out of time to enter so many programs that you've wanted to try?
INTERACTION INTERNATIONAL Programs are still on tape and still at the low price of only $6.00 a tape or $30.00 per year. The year's Subscription includes a free Radio Shack Cassette Storage Album. Look at it this way: The BASIC Overlay alone is worth $19.95 to anyone! (Well, that's what I think) Not to mention all the other good stuff that you'll be getting! Subscribe now! The cassette album offer Expires June 30, 1983. All tapes guaranteed to work or you will receive a speedy tape replacement.

WANTED: We Want Your Ads Now! Your BASIC Programs, your Hardware, Or whatever you have, your ads are a vital source for INTERACTION INTERNATIONAL. Many people look forward to the products. Five Lines of advertisement are totally free for Software Products. For more information, contact George A. Leggett. We Need Your Hardware and Software Ads Now!!!

NAME LIST--A list of 85'Interact'Owners, Great for pen pals Or mailing lists. The list does not come on tape. $3.00 per copy. Wait for free catalog. M.G.H. Software, Box 645, Bayfield, WI 54814

WANTED 7 or more INTERACT OWNERS TO JOIN ME IN PURCHASING INTERWORD word processing program at 19.50 each. Supports Micro Video and Slagh RS-232 ports, has editing commands, cursor-up, down, left and right, carriage return line feed, backspace, add a space, erase a character, add a line, delete a line, tab set and tab clear. I really want to own this program and with a total of 10 or more people it's ours very cheap. Please send Name, address and check to Barbara bridges, p.o.box 42, Badger, CA. 93603.

I have had 5 replies to a similar ad in HUG newsletter so 4 more will be added.

INTERACT EXPANSION PRODUCTS

Features:

Does not require major surgery to the Interact - no soldering of wires to the printed circuit board or cutting of circuit traces - The Interact can be quickly restored to original condition
Expansion not restricted by space inside the Interact
Allows for memory expansion up to nearly **155K**
32K RAM card (one such card brings total Interact memory to 48K) has totally hidden refresh - no wait states as the case for the resident 16K or other internal expansions - this means that programs written to this block execute faster
Expansion frees memory space wasted because of the original Interact design

Supports up to four 5 1/4 inch floppy disc drives

Supports conversion to S-100

The expansion consists of an expansion interface board (IE) which buffers and brings out from the Interact enclosure the necessary signals on a 40 conductor ribbon cable. The ribbon cable is in turn plugged into a motherboard (IMB-2) which is housed inside an enclosure external to the Interact. Expansion products such as the IMEM-1 32K RAM card are then plugged into the motherboard.

Prices of bare boards for products available now:

IE Expansion Interface Board	$25.50
IMB-2 Motherboard	$24.50
IMEM-1 32K Memory Board	$44.50
IEN-1 Enclosure	$19.95

The above products are also available in complete kit and assembled forms. See below for address to send for literature.

The floppy disc and S-100 hardware are presently in existence in prototype form. If you are already on our mailing list, you will be notified when the finished product is available. Otherwise send a business-sized self addressed stamped envelope to:

Walter H. Jopke Jr. 5016 West 99th St. Bloomington, Minn. 55437

Also available INTERWORD Word Processing Software

1-$59.50, 2-$49.50 each, 3-$39.50 each, 4-$30.50 each,
5-9 $25.50 each, 10 or more - $19.50 each

To take advantage of the quantity price advantage for Interword, you must supply the names and addresses for all persons involved.

QUALITY PROGRAMS FOR THE INTERACT
from: David J. Schwab
10 Jay Lee Court
Ann Arbor, Michigan 48104

SKETCH PAD - Basic program with extensive machine language subroutines for creating, modifying, and saving screen displays. Draws open and filled circles (round ones!), triangles, rectangles, lines, and letters with super-fast joystick positioning. Saves screen on tape with or without stop code (to create program banners). Hours of fun for all ages.....$8.00

EDU-BASIC OVERLAY - Allows for PEEK, POKE, and USR type facilities in EDU-BASIC. Also for use with Slash U80 port to direct output or listings to port. Use this powerful language to its full potential.....$8.00

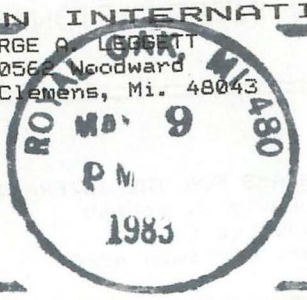
QUEST in EDU-BASIC - An 8k adventure program. You must retrieve a treasure from an underground maze. Descriptions are given of each room and you have 6 directions in which to try to proceed. A pirate lurks in the maze and may steal the treasure back.....$5.00

8080 DISASSEMBLER in BASIC - This Basic program lists adresses, contents, corresponding ASCII character, and standard 8080 mnemonic assembler language op codes and registers for any memory locations. Includes complete instructions and sample output listing form.....$5.00

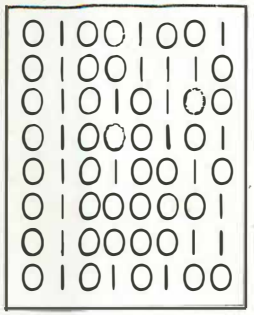
```
10 REM                 QUEST IN EDU-BASIC BY D.J.SCHWAB
20 REM                  FROM A PROGRAM BY ROGER CHAFFEE
30 REM
40 REM    THE PROGRAM TAPE FOR QUEST CONTAINS TWO FILES.  THE FIRST FILE
50 REM    IS THE EDU-BASIC PROGRAM AND THE SECOND IS A DATA FILE CONTAI-
60 REM    NING THE NODE INTERCONNECTIONS.  TO RUN THE PROGRAM:
70 REM
80 REM         1. LOAD EDU-BASIC.
90 REM         2. LOAD QUEST PROGRAM WITH LOAD COMMAND.
100 REM        3. START PROGRAM WITH RUN COMMAND BUT LEAVE READ BUTTON
110 REM           IN DOWN POSITION.  QUEST PROGRAM WILL READ DATA FILE.
120 REM
130 REM     IF YOU WANT TO START OVER WITHOUT RELOADING THE PROGRAM AND
140 REM     DATA, DELETE LINE 400 TO INHIBIT DATA FILE LOADING AND RESTART
150 REM     WITH A RUN COMMAND.
160 REM
170 REM     IN QUEST SOME PASSAGEWAYS, INCLUDING DEAD ENDS, OPEN AND CLOSE
180 REM     DEPENDING ON WHETHER YOU ARE CARRYING THE TREASURE FOR THE
190 REM     FIRST OR SECOND TIME SO DON'T BE AFRAID TO TRY THE SAME PASSA-
200 REM     GEWAYS AGAIN.  IF YOU WANT TO KNOW MORE ABOUT THE PROGRAM,
210 REM     CONSULT BYTE MAGAZINE JULY, 1979 OR SEND SPECIFIC QUESTIONS
220 REM     WITH A SELF-ADDRESSED STAMPED ENVELOPE TO ME.
230 REM
240 REM                          HAPPY HUNTING!!!
Ok
```

INTERACTION INTERNATIONAL

GEORGE A. LEGGETT
20562 Woodward
Mt. Clemens, Mi. 48043



$ 3.00 A COPY
$15.00 FOR 6
ISSUES A YR.



INTERACTION INTERNATIONAL

A MAGAZINE FOR INTERACTORS AND FOR ALL PEOPLE WHO WANT THE COMPUTER KNOWLEDGE OF TOMORROW, FOR TODAY

JULY-AUG. 1983 VOL. IV NO. 4

TABLE OF CONTENTS

CREDITS

EDITOR/PUBLISHER	George A. Leggett
FEATURES EDITOR	Lora A. Leggett
PROGRAMMERS/AUTHORS	Dean Anschultz
	Reese Anschultz
	Russell Schnapp
PRODUCTION	George A. Leggett
TRANSCRIPTION ASSISTANCE	Lora A. Leggett
DRAFTSMAN	Daryl Waananen
PRINTER	Ted McAdam

"Congress shall make no law abridging the freedom of speech or of the press or the right of the people..."

Well, Congress shall make no such law and maybe there is no such law. Nevertheless, the freedom of speech and the press now and then does get abused. If it sounds like I'm about to teach you 12th-grade Government Class here, not at all. It this issue's editorial.

PUBLISHER'S STATEMENT
By George A. Leggett

It awes me to think of the great responsibility one has as the publisher or editor of a piece of work that so many people see; how we can influence their lives with ideas and thoughts, how we can share knowledge of others and we can learn so much from the experience of so many other people. If we go back several hundred years before the advent of printer word on paper it is no wonder why the world took so many hundreds of years to develop. The only form that people had of passing on knowledge was through their children and then passing it along a chain of successors. Now and then, a message could be changed or misinterpreted or just plain forgotten. Thus, with the printed page, we for the first time in the history of the world can finally document our work, our thoughts, our ideas, our hopes, our feelings and we can pass it along not to one or two people but to hundreds, thousands, even millions of people! What a marvel this is! I wonder what a man like Shakespeare would have thought. I wonder what Aristotle would have thought. Plato passing out his thoughts and ideas. If they only knew back then what a tremendous power the word has and the printed word has that much more power.

It is I think enormously important that we do not abuse this power. To the point at hand. In putting together this magazine I do enjoy it. I take the compliments with the criticism. It's all part of the job which I enjoy and welcome both. Though compliments are are always better to receive than criticism as anyone knows!

One of my goals in publishing this magazine is to open a wider majority of the audience to make sure that everyone is getting something they wanted whether it be a Machine Language program, a hardware idea, a BASIC program or another language such as FORTH or whatever it may be even the profiles and interviews. The ideas that people give me and share with you so we may publishe them are a vital force of this magazine. I had to do the first two magazines as you know primarily by myself with my wife's help because I wanted the magazine to get off the ground before 1990. Which it did! However, once I had the opportunity to get the great work from you out there I am more than happy to turn the helm over, as it were.

This is why it makes me very angry to see that another such publication of this computer does not entertain ideas and thoughts that we all share I'm sure. I'm annoyed in that no publisher or editor has the right to only put its own views and its own products and ideas. And then sell it as a magazine. As an advertisement they have the perfect right. They are selling a product. In all cases I know, the purpose of a magazine is to inform you and present to you an unbiased forum for ideas. They must freely express no matter how much they would rather not the thoughts and ideas of other people. It is my oppinion and I say this loud and clear that it is my oppinion only, that if we as editors and publishers ever stop listening to the public, well, we might as well be living in another country. I'm sure Interact would display the flag quite well--something with a red background.....But we're not there, we're here and being here we must do our best to make sure that we display red, white and blue on the screen! And that means FREEDOM! Enough of this First Ammendment stuff and let's move on to a great new issue, Vol. IV no. 4--a pair of 4's for us!

First of all, the majority of this issue was written and produced by one man and his son. My son is too young so therefore, it is not me! It is Mr. Dean Anschultz 22304 Gilmore St. Canoga Park, CA 91303 and his son Reese. The three games (EELS, MANHOLE and MOO) were written by Reese Anschultz. I give my undying gratitude to theese two men who have worked so hard on theese programs. Mr. Anschultz also gives us our Machine Shop Talk this time on the Assemblex/Editex Slagh Interface. He has also done the typesetting of our first FORTH material which is the documentation of the version of FORTH by Russell Schnapp.

Further along in this magazine I will be adding some comments and clarification of the materials you will be seeing. This will also be the introduction of my Commodore VIC 1525E Graphics Printer which was added to my system in April. You will also note that the page headings are done in that format for this issue. In the future I will have more available space and an opportunity to demonstrate some of the printer's capabilities.

During the past few months I have done some things with BASIC you would not believe. As you saw in Vol. IV no. 3 there were five new commands. Since then I now have what I call SUPER BASIC III which has those five commands and over 30 more. It also includes an 8080 2K Monitor.

By combining the 8080 and this new BASIC in a 32K Interact you get the speed of Machine Language for graphics and you get the ease of programming math and the bulk of the program in BASIC. Truly a happy marriage and for me the ultimate language. So I will just be spending more time improving my language. It might be a point on your side for those of you using FORTH as I am always defining new words in my vocabulary. I guess this is what FORTH is all about--defining words and making them do a specific job. Please anyone out there correct me if I'm wrong and I welcome anyone who wants to write an article explaining what they use FORTH for how they use it and why. We do need feedback or we can't get FORTH off the ground.

Finally we have a Hard Facts which is a very simple thing but very unique in design. It's a Fire Button Extender for the Interact by using a very interesting technique. The idea was brought to my attention by Mr. Ron Kregoski for whom I do much of my commercial work. In his work we needed to extend the Fire Button to prompt the program from far away. Kind of like a slide projector where you press the little button and the next slide comes up. Read all about it--it's gorgeous. Strangely enough, the article will be written by my wife with Mr. Kregoski. She never believed she would be assigned a Hard Facts. There's a first for everything.

So as always I hope you will read and enjoy this issue. We do need more programs, more articles and more input from all of you! Let's keep the freedom of the press free and open and only use it to better everyone so we can all learn, we can all enjoy and we can all share the knowledge and the experience of so many great people.

Sincerely,
George A. Leggett

RANDOM REMS
By Lora A. Leggett

Let's give credit where credit is due! When you submit material to us to be published in the magazine, please be sure to indicate clearly that the work you are sending is your own and if you have modified the work of someone else, please give that person credit for the initial effort that you are improving upon. In most cases, the other person is not at all going to mind having his work shown, but like those days gone by when we were in first grade, we all still like to have a little gold star on our paper. Now, instead of the star it suffices to have our name mentioned. Recognizing the amount of hard work that has been done, we do want to give the right people a plug--so if we have ever failed to give anyone full credit--most sincere appologies.

Since this magazine focuses so much attention on one machine, the computer, and most specifically the Interact computer, it is amazing how many other machines are put to work each time we put together a magazine. This article is being typed on a Remington Standard typewriter. (I'll process mo own words for a while yet, thank you!) Listings are done on an Interact-driven ASR33 Teletype. George will be using his Commodore 64 computer, the VIC 1525E printer and storing material on

disk for later reference. He uses a portable cassette recorder to dictate articles for me. He uses a paper cutter for some of the layout work.

I use a few unique machines to help put together my parts of the magazine. Perhaps the oldest and least unique of theese is the one invented by Alexander Graham Bell. Though more costly than the mail, it is quicker and enables me to get a better grip on the fact that you people are all very real and I enjoy all of our conversations. Sometimes I scribble notes in Braille using a Perkins Brailler. I have a file of Braille note cards with everyone's addresses and some phone numbers and it serves as the "master" for subscription labels and correspondence. So, since I use Braille, how do I know I have your labels on right-side-up or that they didn't runn all together or something? Or, if I'm typing, how do I know what I wrote last? This is done by using one of my favorite pieces of equipment. It is called an Optacon (Optical To Tactile Converter) and is about the size of the larger portable cassette machines, (not those neat little mini recorders). Using the machine requires a little practice but if you spend enough time at it you can get by for reading mail or doing the work for the magazine. It is a little slow for reading books, but you can giving yourself time. Thirty or forty words per minute is good and common for a reading speed though I have heard of someone who could read 80 words a minute. Your left hand is placed on the unit with the index finger in what is called a Stimulator Array. This is composed of six rows of 24 vibrating pins each and theese stimulators come on and off as a light-sensitive probe or camera is moved across the printed page with your right hand. I suppose you could think of theese as pixels coming on and off under your index finger to form the shapes of the letters going by.

In a future issue, we will explore even more of the machines used to print the magazine. Meanwhile, keep the great material coming so we can keep all theese wonderful machines off the unemployment line! After all, perhaps the most important and most irreplaceable and still the least understood machine of all is the human brain which can be run independently or in series with other machines of its kind and to which the potential has never been fully tested.

Since so much of this magazine has been put together by you, the regular feature Interactors' Input will return next issue with many of your comments and questions. We have not had the opportunity to get a note or a call back to each of you, but we greatly appreciate hearing from you and thank you for sharing your thoughts with us.

ADJUSTMENT FOR Vol. 4 no. 3

On Page 24 in the 8080 listings where the instructions are put together by types of functions, there is an error in the Jumps, Calls and Returns section. Unconditional Jump in the left-hand column should read C3 as there is no such 8080 instruction as CB.

INTERACT FORTH 1.1

Differences from fig-FORTH Model

character replaced by %, @ character replaced by ^.

(LINE)	deleted. Replaced by (SLINE) in EDITOR vocabulary.
+BUF	deleted.
.LINE	deleted.
BLOCK	deleted.
BLOCK-READ	Replaced by GREC.
BLOCK-WRITE	Replaced by PREC.
BUFFER	deleted.
COLD	After initial load, COLD is invoked. Resets cause WARM to be invoked.
DLIST	Not implemented. See VLIST.
DR0 DR1	deleted.
DUMP	Not implemented.
LIST	deleted, see L in EDITOR vocabulary.
LOAD	deleted. Replaced by SLOAD or CLOAD.
MON	replaced by BYE.
R/W	deleted. See GREC & PREC.
UPDATE	Only ensures that you will have your hand slapped if you try to GETSCR before the current screen has been DISCARDed or PUTSCRed.
WIDTH	has been set to 3, to save RAM.
[	replaced by LBR.
[COMPILE]	replaced by BCOMPILE.
]	replaced by RBR.

Added Words to FORTH Vocabulary

?GREC f —

Intended to follow a GREC invocation. If f is true, an error message is generated and the tape is turned off.

CLOAD

Loads a screen from tape and interprets it.

CRC addr count — b

Computes an end-around carry accumulated addition upon the indicated memory block.

DING
Emits a BEL character.

DISCARD
Clear current screen, and reset the UPDATED flag.

GETSCR
If the UPDATED flag is true, then an error message is generated. Otherwise, the screen is loaded from the cassette. A CRC is computed and verified.

GREC addr buffersize — addr+recsiz ff
addr buffersize — tf
Reads a record off the cassette into the address supplied, with a limiting "buffersize". Successful read yields flag=0 (false).

PREC addr recsiz — addr+recsiz
Writes a single record to the cassette. The "recsize" parameter must be 0 to 255. A 0 "recsize" value is treated as 256.

PUTSCR
Writes the current screen to cassette, and clears the UPDATED flag. A CRC is saved with the screen.

READLEAD
Starts the tape moter, and reads until a valid leader tone is detected. Interrupts (and keyboard) are disabled.

SLOAD
Interprets the current screen.

TAPEOFF
Turns off the tape motor. Enables interrupts.

TAPEON
Turns on tape and disables interrupts (and keyboard).

UPDATED — addr
Address of the one byte UPDATED flag is returned.

WRITELEAD
Turn on tape (disable interrupts), and write a ten second leader on cassette.

REWIND
Turn on tape motor until a keystroke is detected.

Words in EDITOR Vocabulary

Most of these words are taken from the "Line editing" command definitions in screens 87-91 of the fig-FORTH model.

%LAG — addr count
Leave address and count of text following the cursor.

%LEAD — addr count
Leave address and count of text up to the cursor.

%LOCATE — offset line
Leave line number and offset to the cursor.

-MOVE sourceaddr destline —
Move one line of text from "sourceaddr" to "destline" in screen.

(SLINE) line — addr count
Converts a line number into TYPE format.

D line —
Delete "line", hold its text at PAD.

E line —
Blanks out (erases) "line".

H line —
Hold line's text at PAD. The line is untouched.

L
List the screen. Waits for a keystroke after each two lines. Typing a CR terminates the listing.

LINE line — addr
Convert a line number to an address of the line.

M n —
Move cursor by signed amount "n". Print the line upon which the cursor resides, displaying the cursor position as an underscore.

P line —
Accepts a line of text terminated by a CR, stores it at PAD, and copies it to "line". Example: `1 P THIS IS LINE ONE OF THE SCREEN.` (CR) (The screen contains 16 lines, numbered 0 through 15.)

R line —
Replaces "line" with text at PAD.

S line —
"Spread." Move "line" through 14 down one line, and erase "line".

T line —
Type "line" to screen.

TEXT c —
Accept a line of text, to be terminated by a character "c" or a CR, and place it at PAD.

I line —
Insert the text in PAD into "line", shift the rest down a line.

WHERE n m —
To be used following an aborted SLOAD or CLOAD. Types out line 0 and the offending line. (ABORTs leave two numbers on the stack for just this purpose.)

" —
Accept a line of text terminated by another " or by a CR, and place it in PAD. Example:
" THIS IS LINE OF TEXT TO BE INSERTED AT LINE 4 " 4 I (CR)

FORTH: A Personal View

First of all my thanks go out to Mr. Russ Schnapp for supplying this word vocabulary from his package of FORTH. Also thanks must go out to Mr. Dean Anschultz for doing the typesetting. I have decided to print this material not for those of you who have FORTH already but for those who have been thinking of starting out in the language. I have spent many hours looking over this and other FORTH material including STARTING FORTH by Brodie. I have come to the conclusion that after spending time with FORTH I needed a fifth!

Perhaps this is not the time nor the place for my opinions about this language but nevertheless here goes anyway. By no means is this to reflect upon the marvelous work done by these and all the other authors and workers of FORTH. They are not to blame. They receive my utmost admiration on how they could put up with such a goofy language. This is how I felt after trying to interpret about 500 pages of FORTH material. **Please note that the following are my own observation on FORTH and if any of these are incorrect I am sorry but shall try my best to remedy this in the future.** The language was designed as a go-between since BASIC is very easy but slow. Machine Language is very fast but harder to learn for many people. I see perfectly what the objective was in creating the language but in no way did they conquer this objective in my opinion. It reminds me of Part 12 of the series COSMOS by Dr. Carl Sagan where he refers to the rosetta stone and

Words in Assembler Vocabulary

Registers: A B C D E H L M PSW SP

Labels: NEXT HPUSH DPUSH

Instructions:

Standard INTEL 8080 mnemonics, suffixed by a comma. Arguments are in standard order, but precede the instruction mnemonic.

Examples: D H MOV, assembles an instruction which moves H to D.

B 5000 LXI, loads 5000 into BC.

In addition to the INTEL mnemonics, LS and GE are synonymous with C and NC respectively, in conditional instructions: JC is therefore equivalent to JLS.

There are no labels, per se. As in colon definitions, control structures reduce the need for them.

Control Structures:

cond IF ELSE ENDIF as per colon definitions.
BEGIN cond UNTIL as per colon definitions.

where "cond" is either 0< or 0= followed optionally by NOT.

Usage:

The assembler is invoked by the FORTH vocabulary word CODE. *See the FORTH documentation for further information.* BC is IP, and must be preserved, and CODE words are terminated by a jump to NEXT, HPUSH, or DPUSH. These, respectively, return; return after pushing HL; return after pushing DE, then HL. Example: CODE + H POP, D POP, D DAD, HPUSH JMP, (Definition of the word +)

Words in GRAPHICS Vocabulary

This vocabulary implements a plotter like abstraction. The carrier begins at 0,0 at the top left with the pen up. Current color is inkwell 3—white.

INKWELL number —
Set pen color to "number" (between 0 and 3).

PENUP
Raise the pen.

PENDOWN
Lower the pen.

CTABLE
A variable whose first four bytes contain the ink colors to go in wells 0 through 3.

COLORSET
Apply CTABLE.

PLOTDOT x y —
Plot a dot at x,y in the current color.

DRAW x y —
If pen is down, draw a line from the carrier to x,y in the current color. Whether the pen is up or down, leave the carrier at x,y.

CLS
Clear screen to inkwell 0 color.

PLOTBLOCK xwid ywid —
Plot a rectangle, "xwid" by "ywid", in current color at current position. The carrier is at the top left of the rectangle.

PLOTCHAR char —
Plot "char" at current position in current color.

SETCHARTABLE addr —
Associate a graphics character generator table. This table holds as long as backspace is not EMITted. The first two bytes of this table hold character width and height. The remainder of this table is character generation masks. This table is used for characters in the range 01 through 1D.

Example:

```
HEX 0606 VARIABLE ACGEN
70 C, 88 C, F8 C, 88 C, 88 C, 00 C,
ACGEN SETCHARTABLE ( associate ACGEN )
1 EMIT 1 PLOTCHAR ( print an "A" at cursor and carrier )
```

BLTABLE
A variable used by PLOTBLOCK.

%IN — n
This word accepts a number from the keyboard in the current base, and leaves it on the stack.

PTC x y —
This word places the cursor (next print position) at the given column (x) and row (y). A carriage return character cancels this, and places the cursor at the bottom left position. The top left position is column 0, row 0. (Please note that x and y are specified in character positions, rather than pixels.)

WINDOW n —
This word sets the number of lines on the bottom of the screen to be scrolled. 1≤n≤11. The remainder of the screen is not touched. (Initial conditions are as if 11 WINDOW had been executed.) The results of this command hold until the next time WINDOW is executed, even after hitting the reset button.

POINTDOT x y — color
Return the color at position x,y.

RND range — number
Returns a random number 0≤number<range.

SEED
A word used as a seed by RND.

hieroglyphics. Until they found the rosetta stone to unlock the key hieroglyphics were totally unrelatable. Once this key was found every thing fell into place. Well maybe this is the case with FORTH and maybe I just have not found the key yet. One of the main things I think is wrong is that it uses Polish notation. Simply and without going into a lengthy explanation of what that is, it means that all your equations in mathematics are backwards. Also for a language that wants to take you away from Machine language it uses quite a bit of it by Pushing and Pulling on and off the stack. I should say Pushing and PoPPing since Pulling is what a 6502 does but it's all the same. At any rate, it seems to me that one needs a good deal of knowledge about what the CPU is doing. Though I am one of the greatest fans of 8080, this is a roundabout way of using it. Why do we need a middle man? I have always been against middle men in life.

I do not like to discourage anyone from [illegible] but I am still investigating it myself. I urge any of you out there who might be more well-versed in the FORTH language to please submit Programs in it and more importantly why you feel this is a good language. Or if you have not been successful at FORTH Please write also and tell us why.

To sum up my feelings on languages I think it is important that if you are going to take the time and trouble to learn a language you should learn two on whichever machine you work on--the Machine Language of that system to unleash the full speed and Power of that system and BASIC because of its simplicity to learn and to operate. When you combine the two, as one of my favorite men, Jackie Gleason says, "And Away we go!"

EELS (for the INTERACT)

EELS is an action game for 1 player. (Requires Level II or 8K BASIC)

OBJECT: Eat energy packets to grow to maximum size.

PLAY: Use left joystick to steer continually moving "eel" to energy packet while avoiding contact with the walls of your tank or yourself. Since the eel and the walls are electric, contact is fatal and causes your body to disintegrate. The energy packets provide two benefits, they provide you with energy for more foraging and contribute to your growth. You have a limited amount of time to reach an energy packet, an increasing pitch as you move about indicates your desperation. Should you fail to reach the energy packet in time, your body withers and dies. Each energy packet has a limited (randomly determined) lifespan and may evaporate before you reach it; however, you may take heart in knowing it will be replaced immediately (alas, elsewhere) by another. There is always one energy packet for you to seek. The energy (foraging time) provided by each packet is the same but the growth value varies from packet to packet; the growth value (scoring value) is indicated at the top right of the screen. When your length reaches 250, your tail falls off to clutter the landscape and you continue on as a new young eel.

After you are killed and your body disappears, use the left fire button to start a new game.

SCORING: Points are scored for each energy packet eaten; the value of the current energy packet is shown in the upper right. Your current score is shown in the center and the highest score attained for the session is shown in the upper left.

```
100 REM EELS          11 AUG 82
110 CLEAR50:PRINTCHR$(3);:DIMX(250),Y(250):HS=0
120 XL=30:YL=30
130 X(0)=INT(115/2):Y(0)=INT(65/2):XO=(115-XL)/2:YO=(65-YL)/2
140 S=2*(XL+YL):F=0:E=0:G=5:SC=0:D=3:CLS:COLOR7,5,0,6
150 FORI=YOTOYO+YL:PLOTXO,I,2:PLOTXO+XL,I,2:NEXT
160 FORI=XOTOXO+XL:PLOTI,YO,2:PLOTI,YO+YL,2:NEXT
170 OUTPUTHS,10,71,1:OUTPUTSC,38,71,1:GOSUB440:OS=S
180 TM=INT(RND(1)*S/2)+3
190 OS=OS-1:IFOS=0THEN500
200 TM=TM-1:IFTM=0THENOD=AD:AD=0:GOSUB410:AD=OD
210 TONEOS+20,3:IFG>0THENG=G-1:GOTO230
220 PLOTX(E),Y(E),0:E=E+1:IFE>250THENE=0
230 GOSUB330:OF=F:F=F+1:IFF>250THENF=0
240 IFF=ETHENF=E+5:IFF>250THENF=F-251
250 Y(F)=Y(OF):X(F)=X(OF)
260 IFD=0THENY(F)=Y(OF)+1
270 IFD=1THENY(F)=Y(OF)-1
280 IFD=2THENX(F)=X(OF)-1
290 IFD=3THENX(F)=X(OF)+1
300 IFPOINT(X(F),Y(F))=2ORPOINT(X(F),Y(F))=1THEN540
310 IFPOINT(X(F),Y(F))=3THENGOSUB400
320 PLOTX(F),Y(F),1:GOTO190
330 J=JOY(0)
340 IFJ=0THENRETURN
350 IFJ=4THEND=0:RETURN
360 IFJ=8THEND=1:RETURN
370 IFJ=1THEND=2:RETURN
380 IFJ=2THEND=3
390 RETURN
400 OUTPUTAD,76,71,0:AD=INT(RND(1)*9)+1:OUTPUTAD,76,71,1:OS=S
410 TM=INT(RND(1)*S/2+S*.25)+3
420 PLOTRX,RY,0:PLOTRX,RY+1,0:PLOTRX+1,RY,0:PLOTRX+1,RY+1,0
430 OUTPUTSC,38,71,0:SC=SC+AD:OUTPUTSC,38,71,1:G=AD
440 RX=INT(RND(1)*XL)+XO:RY=INT(RND(1)*YL)+YO+1
450 IFPOINT(RX,RY)>0THEN440
460 IFPOINT(RX,RY+1)>0THEN440
470 IFPOINT(RX+1,RY)>0THEN440
480 IFPOINT(RX+1,RY+1)>0THEN440
490 PLOTRX,RY,3:PLOTRX,RY+1,3:PLOTRX+1,RY,3:PLOTRX+1,RY+1,3:RETURN
500 SOUND3,50:I=E:SOUND3,51
510 PLOTX(I),Y(I),0:I=I+1:IFI>250THENI=0
520 IFI<>ETHEN510
530 GOTO590
540 SOUND3,150:J=0:FORI=0TO250:IFPOINT(X(I),Y(I))<>1THEN560
550 X(J)=X(I):Y(J)=Y(I):J=J+1:IFI<>TTHEN550
560 NEXT:SOUND3,151
570 T=INT(RND(1)*(J+1)):PLOTX(T),Y(T),0:X(T)=X(J):Y(T)=Y(J):J=J-1
580 IFJ>-1THEN570
590 FORJ=1TO1000:NEXT:IFSC>HSTHENHS=SC
600 TONE40,40:FORJ=1TO100:IFFIRE(0)=0THEN120
610 NEXT:GOTO600
OK
```

MANHOLE (for the INTERACT)

MANHOLE is an action game for 1 player. (Requires 8K Graphics BASIC)

The situation: The city sewer workers have gone on strike at midnight. To insure that their action is noticed, they have removed all the manhole covers. You, as a safety-conscious public-spirited 'sewervisor' were able to locate a single manhole in reserve storage and have now taken upon yourself to use it to keep pedestrians from falling through the open holes into the sewer. You are there early in the morning before the first pedestrian. Traffic is light at first, but picks up quickly.

OBJECT: Escort pedestrians entering at lower right and upper left to their destination at the upper right.

SKILL LEVEL: The skill level determines how rapidly the traffic increases. At the lowest level, you need escort only one pedestrian at a time through the loop five times before having to escort two at a time. Five groups of two pedestrians must now be escorted before moving on to groups of three. At the highest skill level, after you have escorted only one through the loop, you must escort two, and then three, quickly escalating until groups of ten pedestrians must be cared for.

PLAY: Use left joystick to position yourself with manhole cover at manholes where pedestrians are crossing. They only need a little help; if you are there only briefly while they are crossing the hole, they can manage.

After the third pedestrian has drowned in the muck, use the left fire button to start a new game at the same skill level.

SCORING: One point is scored for each pedestrian going offscreen at the upper right. The number of casualties is shown in black at the center. Your current score is shown at the left and the highest score attained for the session is shown in at the right.

```
10 REM MANHOLE        17 JUL 82          (8K BASIC)
20 GOTO360
30 I=T(JOY(0)):IFI<0ORI=0JTHENRETURN
40 C=3:SOUND4,322:GOSUB50:SOUND4,323:Q=I:W(M(I))=1:I=0J:0J=Q:C=0
50 PLOTPX(M(I)),PY(M(I))-RW,C,PW,RW
60 POKED1,C1:POKED2,C2:OUTPUTCHR$(1),PX(M(I))+3,PY(M(I))-3,C:RET
URN
70 PLOTPX(I),PY(I),0,PW,PW-4:RETURN
80 IFI>8THENPOKED1,B1:POKED2,B2:GOTO100
90 POKED1,A1:POKED2,A2
100 OUTPUTCHR$(1),PX(I)+4,PY(I)+9,1:RETURN
110 IFJ>8THENPOKED1,B1:POKED2,B2:A3=B1+B2*256:GOTO130
120 POKED1,A1:POKED2,A2:A3=A1+A2*256
130 FORZY=7TO-12STEP-1:ZZ=ZY+1:C=0:GOSUB200:IFZY<-3THENPOKEA3,PE
EK(A3)-1
140 ZZ=ZY:C=1:GOSUB200:TONE20-2*ZY,70:NEXT
150 C=0:GOSUB200:SOUND1,550:POKEA3,10:CR=CR+1:SOUND1,551
160 OUTPUTCR-1,49,36,0:OUTPUTCR,49,36,3:IFCR<3THEN330
170 IFSC>HSTHENOUTPUTHS,70,36,0:HS=SC:OUTPUTHS,70,36,1
180 GOSUB30:IFFIRE(0)=0THEN610
190 GOTO180
```

```
200 OUTPUTCHR$(1),PX(J)+4,PY(J)+ZZ,0:RETURN
210 FORF2=1TOFQ:T2=TT:S=-1:GOTO240
220 IFT2=0GOTO240
230 IFRND(1)>.2THENS=NOTS
240 E=0:FORJ=NP-1TO0STEP-1:GOSUB30:IFP(J)=0THEN320
250 E=-1:IFJ=NP-1THENOUTPUTSC,5,36,0:SC=SC+1:OUTPUTSC,5,36,2
260 IFPX(J)>=0THENI=J:GOSUB70
270 P(J)=0:IFW(J)=0THEN110
280 P(J+1)=1:W(J+1)=1:IFPX(J+1)<0THEN310
290 IF POINT(PX(J+1),PY(J+1)-1)=0THENW(J+1)=0
300 I=J+1:GOSUB80
310 TONE(J+10)/2,70
320 NEXT:IFSTHENE=-1:P(0)=1:W(0)=1:I=0:GOSUB80:TONE2,100
330 IFSTHENT2=T2-1:IFT2=0THENS=0
340 IFEGOTO220
350 NEXT:TT=TT+1:GOTO210
360 PRINTCHR$(8);
370 CLS:PRINT:PRINT"INSTRUCTIONS?":FQ$=INSTR$(1):IFFQ$="Y"THENGO
SUB640
380 PRINT"ENTER SKILL":PRINT"LEVEL (1-5) ";:FQ$=INSTR$(1):PRINTF
Q$
390 FQ=6-VAL(FQ$):IFFQ<1ORFQ>5THENPRINT"1 THRU 5 PLEASE":GOTO380
400 HS=0:NS=2:SH=77:SW=115:FR=INT(SH/NS):RW=2:RL=SW:RH=INT(FR/2)
410 WL=RL:WW=INT(FR/4):COLOR6,2,4,0:CLS:FORI=0TONS-1
420 PLOT0,RH+I*FR,3,RL,RW:PLOTO,FR*I,2,WL,WW:R=FR*I+WW
430 FORJ=0TOWLSTEP8:PLOTJ,R,2,3,1:PLOTJ+2,R+1,2,1,1:NEXT:NEXT
440 NL=8:NN=1:NT=NN+NL:NP=NS*(NL+NN):NM=4
450 DIMPX(NP),PY(NP),M(NM),T(10),P(NP),W(NP)
460 PX(NP)=-1:FORI=0TO10:READT(I):NEXT
470 DATA-1,-1,-1,-1,-1,2,3,-1,-1,1,0
480 PW=INT(SW/NL):FORI=0TONP-1:PY(I)=INT(I/NT)*FR+RH+RW
490 K=I-INT(I/NT/2)*NT*2
500 IFK>=NTTHENK=2*NT-K-1-NN:IFK<0THENPX(I)=-1:GOTO530
510 IFK>=NLTHENPX(I)=-1:GOTO530
520 PX(I)=INT(PW*(NL-K-1))
530 NEXT:FORI=0TONM:READM(I):NEXT:DATA2,5,11,14,14
540 POKE19670,154:POKE19671,95:POKE19690,154:POKE19691,95
550 POKE19542,105:POKE19543,95:FORJ=24476TO24511:READX:POKEJ,X:N
EXT
560 DATA10,8,24,152,136,124,26,25,56,104,79,193
570 DATA10,8,24,25,17,62,88,152,28,22,226,131
580 DATA10,8,129,153,90,60,24,24,60,60,36,102
590 A2=INT(24476/256):A1=24476-256*A2:B2=INT(24488/256):B1=24488
-256*B2
600 C2=INT(24500/256):C1=24500-256*C2:D1=24545:D2=24546
610 C=0:FORI=0TONM-1:GOSUB50:NEXT:PLOT5,32,0,65,5:OJ=4
620 FORI=0TONP-1:P(I)=0:W(I)=0:IFPX(I)>=0THENGOSUB70
630 NEXT:SC=0:CR=0:TT=1:D=5:GOTO210
640 CLS:PRINT"OBJECT:  TO KEEP PEDESTRIANS FROM FALLING INTO OPE
NMANHOLES."
650 PRINT:PRINT"PLACE MANHOLE UNDER PEOPLE BY  MOVING LEFT JO
Y"
```

```
660 PRINT"STICK IN CORRECT DIRECTION.":GOSUB740
670 PRINT"PEDESTRIANS ENTERFROM THE LOWER"
680 PRINT"RIGHT AND MOVE INA CLOCKWISE ORDERTO EXIT AT THE   UPP
ER RIGHT."
690 PRINT:PRINT"ONE POINT IS":PRINT"SCORED FOR EACH  PERSON SURV
IVING THE TRIP."
700 GOSUB740:PRINT"THE THIRD PERSON TO FALL INTO THE SEWER ENDS
GAME."
710 PRINT:PRINT"HIGH SCORE IS     RETAINED THROUGH SUBSEQUENT GAM
ES."
720 PRINT"TO START A NEW   GAME, PRESS THE  FIRE BUTTON."
730 GOSUB740:PRINT"NOTE:":PRINT"  A BRIEF RESCUE IS ADEQUATE."
740 FORFQ=1TO1000:NEXT:PRINT:RETURN
OK
```

MOO (for the INTERACT)

MOO is a strategy game for 1 player. (Requires Level II or 8K BASIC)

OBJECT: Guess a number selected by the computer. (Similar to the commercial game of Mastermind.)

OPTIONS: You may choose the length of the number (number of digits), the range of values each digit may take (from 0 to n, where n is 9 or less), and whether repeated digits may exist within the number. An error message is produced if you attempt to have more digits in the number than legal digits without repeats. For example: the game of Mastermind is played with 4 positions (length of number) and six color types (digit values of 0-5 allowed) usually allowing repeats.

PLAY: After responding to all the requests to enter option values, a ? prompts you to enter a number. Key in a number with as many digits as you requested followed by CR. The computer responds with B = # C = #. The B number represents 'Bulls' and the C number represents 'Cows' yielding a justification for the somewhat strange name of this game. A bull signifies that you have guessed one of the digits of the computer's number and its correct position. When you have as many bulls as digits in the number, you have guessed the entire number correctly and the game is over. Each cow represents a digit that you have guessed which exists in the computer's number that you have incorrectly positioned. (Ranking cows below bulls seems sexist, but don't blame me, I didn't invent the game.) For example: if the computer's number is 1234 and you guess 1523, you get one bull for the 1 and two cows for the 2 & 3. If you guess 1222, you get 2 bulls and no cows. If you guess 1022, you get one bull and one cow.

After you have guessed the correct number, you are given the option to play another game, and if so, to use the same setup or not.

SCORING: Your score is the number of moves required to guess the number; it is announced at the end of the game.

```
10 REM MOO              01 JUL 82
20 CLS:PRINT"HOW MANY":INPUT"POSITIONS";N:DIMP(N),G(N),C(N):P(0)
=-5
30 INPUT"NUMBERS 0 THRU";T:T=ABS(INT(T))
40 IFT>9THENPRINT"NUMBER MUST BE":PRINT"LESS THEN 10.":GOTO30
50 PRINT"ARE REPEATS":PRINT"ALLOWED?";:A$=INSTR$(1):PRINTA$
60 R=0:IFA$="Y"THENR=1
70 IFR=0ANDA$<>"N"THENPRINT"Y(ES) OR N(O)":PRINT"PLEASE...":GOTO
50
80 IF(N-1>T)ANDR=0THENPRINT"ILLEGAL COMBO.":GOTO30
90 FORJ=1TON
100 P(J)=INT(RND(1)*(T+1)):IFR=0GOTO120
110 NEXT:GOTO140
120 FORK=0TOJ-1:IFP(K)=P(J)THEN100
130 NEXT:GOTO110
140 CLS:GN=0
150 INPUTA$:GN=GN+1
160 IFLEN(A$)<>NTHENPRINTN;"DIGITS":PRINT"PLEASE...":GOTO150
170 FORJ=1TON:G(J)=VAL(MID$(A$,J,1)):NEXT:J=1
180 IFG(J)>TTHENPRINT"NUMBERS 0 THRU";T;"PLEASE...":GN=GN-1:GOTO
150
190 J=J+1:IFJ<=NTHEN180
200 PC=0:PI=0:FORJ=1TON:C(J)=0:NEXT
210 FORJ=1TON:IFP(J)=G(J)THENPC=PC+1:G(J)=-1:C(J)=1
220 NEXTJ
230 FORJ=1TON:IFC(J)=1THEN270
240 FORK=1TON:IFC(J)=1THEN260
250 IFP(J)=G(K)THENPI=PI+1:C(J)=1:G(K)=-1
260 NEXTK
270 NEXTJ
280 PRINT"B =";PC;TAB(10);"C =";PI:IFPC<>NGOTO150
290 PRINT"IT TOOK YOU";PRINTGN;"GUESS";:IFGN>1THENPRINT"ES";
300 PRINT
310 PRINT"PLAY AGAIN?";:A$=INSTR$(1):PRINTA$:IFA$="N"THENEND
320 IFA$<>"Y"THENPRINT"Y(ES) OR N(O)":PRINT"PLEASE...":GOTO310
330 PRINT"SAME SET UP?";:A$=INSTR$(1):PRINTA$:IFA$="Y"THEN90
340 IFA$<>"N"THENPRINT"Y(ES) OR N(O)":PRINT"PLEASE...":GOTO330
350 CLEAR:GOTO20
OK
```

QUILT (for the INTERACT)

QUILT is program which generates all-over patterns by repeating a specified pattern block. After the final pattern is displayed, the colors may be changed and if a printer is available, a reference pattern may be printed. The program was originally developed to produce stitching diagrams for needlepoint, but is suitable for testing of quilting or wallpaper designs. Since the repeat axes need not necessarily be horizontal or vertical, *any* all-over motif may be utilized. The program can also be useful in testing symmetric single figures, since any component may be "flopped" about the horizontal, vertical or both axes.

The pattern is specified by a group of data statements. These specify the colors to be used, the repeat axes steps, the positioning of components within the repeat block and the configuration of each component. The specifications may appear in any order but must be terminated by DATA*. Up to twenty-six different component specifications are permitted, each designated by one letter of the alphabet. Each component is a rectangle of any arbitrarily chosen size. Since the background color is never plotted, components may overlap reasonably without interference. In fact, symmetric figures of odd width will overlap on the mid-line. The coordinate system used has its origin in the upper left at 1,1. Horizontal values increase to the right and vertical values increase downward. The following are the types of DATA statements that are expected:

DATA!nnnn (optional)

Choose colors: the first n refers to the background color, the others refer to colors 1, 2 & 3 respectively as selected in the component specifications. The values for each n range from 0 to 7 where 0 means black, 1 means red, 2 means green, 3 means yellow, 4 means blue, 5 means magenta, 6 means cyan and 7 means white. If this statement does not occur, DATA!0347 is assumed, i.e. black background with yellow, blue and white.

DATA%,xa,ya,xb,yb (one required)

Set repeat step sizes: The 'a' axis is basically horizontal, and xa must be positive. ya is the vertical step associated with xa and may be positive or negative. (remember negative means up) The 'b' axis is basically vertical, and yb must be positive (downgoing) while its associated horizontal step xb may be of either sign.

Examples: DATA%,20,0,0,20 would specify a simple repeat on a square grid.

DATA%,40,0,20,20 or DATA%,20,20,0,40 could be used to specify a checkerboard.

DATA%,20,-10,10,20 would be used with an oblique pattern.

DATAa (up to 26 permitted)
DATAnnn . . . nn
DATAnnn . . . nn
. . . . . .
DATAnnn . . . nn

Specify component: a is a letter naming the component, the description of the component in the form of a color designation map follows immediately. Each n is a digit in the range 0 to 3 specifying which color will occur in that position. The number of digits in each row determines the width of a component. Each row within a component description must contain the same number of digits. The number of rows (starting with a digit) determines the depth of the component.

DATA+a,x,y (as many as memory permits)
Specify component placement within repeat block. x and y, which must be positive, specify a position within the repeat block at which component a will be displayed. (The same component may be displayed in more than one place.) The upper left of the repeat block is x=1 y=1.

DATA-a,x,y (as many as memory permits)
Same as DATA+a,x,y except the component is "flopped" about a vertical axis. i.e. a mirror image.

DATA^a,x,y (as many as memory permits)
Same as DATA+a,x,y except the component is "flopped" about a horizontal axis. i.e. an inverted image.

DATA/a,x,y (as many as memory permits)
Same as DATA+a,x,y except the component is "flopped" about both the horizontal & vertical axes. i.e. a 180 degree rotation.

DATA* (REQUIRED)
Terminate pattern data. (May be inserted before other data to simplify figure in development.)

PROGRAM VARIATIONS

Lines 30, 560 and 580 through 650 as well as the array CC on line 20 are used only for printing and may be omitted if no printer is available. The printer code reflects a Centronics 737 printer on a Slaugh interface. The control codes select 132 character/line mode and half step line advances. Implementation for a particular printer will likely vary. The data statement on line 30 specifies what characters will represent what color in the output. As given, black is ".", red is "r", green is "g", yellow is "y", blue is "b", magenta is "m", cyan is "c" and white is "w". This printer output code can of course be used to output any screen in any program.

If you run out of memory, the dimensions of X$(71),D(26),W(26) may be reduced. X$ need only be as large as the widest component, while D & W need only be dimensioned to encompass the largest component name. (Avoid using Z!)

To speed things up and save the memory required to specify large rectangles of a single color, I have considered adding a DATA$,x,y,c,w,d for this purpose. Implementation in 8K Graphics BASIC would require limiting the w & d values in some cases to avoid going off-screen.

RUNNING THE PROGRAM

I prefer entering the DATA statements to EZEDIT and APPENDing them to a copy of the program, but editing them in BASIC is possible. After entering the data statements, simply say RUN and go get a cup of coffee. The program is not fast! It typically takes about 10 minutes to fill the screen. It is somewhat faster than doing the needlework though.

After the picture is complete, a tone is produced, letting you know that the last little corner that you can't even see has finished. At this point, you have three options, (1) change color by keying 4 digits as specified in the DATA!nnnn description, (2) printout the pattern by keying TYPE (It won't echo and no CR is required.) or (3) return to BASIC to modify your DATA statements by keying QUIT. (This selects reasonable colors for editing before doing END.)

COMMENTS

This program has given me a chance to ameliorate the relationship between my wife and 'that damn box', it is a tool that she can use and enjoy. We have found some interesting psychological aspects of color. For example, in the Fleur-de-lis/Lion pattern in the accompanying listing which was conceived as a carpet for a doll's house; changing the background to white and the figure colors to pastels makes it seem much more suitable for a bedspread. We have experimented with lions facing in opposite ways on alternate rows; having alternate rows inverted so there is no up or down; or removing the border from the lions. Since the changes can be made fairly quickly and easily, we have made the experiments.

THE PRINTED PICTURE:
To Be Or Not To Be
That Was The Question

When I first received the Printed Pages that go along with QUILT and HEXES and so on, they were way to light and my Printer said if we tried to Publish them, they would not come out at all. Now, still in time to make this issue, Mr. Anschultz sent new copies which are dark enough since they were done using a new ribbon. But by this time all the space in this magazine had been allotted. Perhaps, if you write and tell me you would like to see them, I'd be happy to Publish one or two of these Picture. My thanks again to Mr. Dean Anshultz who took the time to send new copies of these Pictures.

```
10 REM "QUILT" 7SEP82
20 CLS:KK=0:A$="0347":GOSUB200:CLEAR150:DIMX$(71),D(26),W(26),CC
3),CC(7)
30 DATA46,114,103,121,98,109,99,119:FORX=0TO7:READY:CC(X)=Y:NEXT
40 KM=INT((FRE(0)-400)/12):DIMCF(KM),CX(KM),CY(KM)
50 READX$:A$=LEFT$(X$,1)
60 IFA$="*"GOTO220
70 IFA$="%"THENREADXA,YA,XB,YB:GOTO50
80 IFA$="!"THENA$=MID$(X$,2,4):GOSUB200:GOTO50
90 IFA$="+"THENQ=0:GOTO180
100 IFA$="-"THENQ=1000:GOTO180
110 IFA$="↑"THENQ=10000:GOTO180
120 IFA$="/"THENQ=11000:GOTO180
130 IFA$>"Z"ORA$<"A"GOTO50
140 I=ASC(A$)-65:IFD(I)<>0THENPRINT"DUPLICATE ";A$:STOP
150 READX$:A$=LEFT$(X$,1):IFA$>"3"ORA$<"0"GOTO60
160 IFD(I)>0THENIFLEN(X$)<>W(I)THENPRINT"BAD WIDTH FOR ";CHR$(I+
64):STOP
170 W(I)=LEN(X$):D(I)=D(I)+1:GOTO150
180 CF(KK)=Q+ASC(MID$(X$,2,1))-65:READQ:CX(KK)=Q-1:READQ:CY(KK)=
Q-1
190 KK=KK+1:GOTO50
200 FORI=0TO3:C(I)=VAL(MID$(A$,I+1,1)):NEXT
210 COLORC(0),C(1),C(2),C(3):RETURN
220 IFXA<=0THENPRINT"NO HORIZ. STEP!":STOP
230 IFYB<=0THENPRINT"NO VERT. STEP!":STOP
240 BW=0:BD=0:FORI=0TOKK-1:Q=CF(I):IFQ>9500THENQ=Q-10000
250 IFQ>500THENQ=Q-1000
260 QQ=CX(I)+W(Q):IFQQ>BWTHENBW=QQ
270 QQ=CY(I)+D(Q):IFQQ>BDTHENBD=QQ
280 NEXT
290 XM=112:YM=76:X0=0:Y0=YM:IFYA<>0THENYO=YO+YB:XO=XO-XB
300 IFYO+YB-BD<YMTHENYO=YO+YB:XO=XO-XB:GOTO300
310 X1=XO:Y1=YO
320 IFX1-XA+BW>0THENX1=X1-XA:Y1=Y1+YA:GOTO320
330 IFX1+BW<=0GOTO500
340 FORK=0TOKK-1:FL=0:FU=0:I=CF(K):IFI>9500THENFU=1:I=I-10000
350 IFI>500THENFL=1:I=I-1000
360 Y2=Y1-CY(K):Y3=Y2-D(I)+1:IFY2<10RY3>YMGOTO490
370 X2=X1+CX(K):X3=X2+W(I)-1:IFX2>XMORX3<0GOTO490
380 RESTORE:A$=CHR$(I+65)
390 READX$:IFLEFT$(X$,1)<>A$GOTO390
400 IFFUTHENFORY=Y3TOY2:GOTO420
410 FORY=Y2TOY3STEP-1
420 READX$:IFY>YMORY<1GOTO480
430 IFFLTHENFORX=X3TOX2STEP-1:C=VAL(MID$(X$,X3-X+1,1)):GOTO450
440 FORX=X2TOX3:C=VAL(MID$(X$,X-X2+1,1))
450 IFC=0GOTO470
460 IFX>=0ANDX<=XMTHENPLOTX,Y,C
470 NEXT
480 NEXT
490 NEXT
500 X1=X1+XA:Y1=Y1-YA
```

```
510 IFY1>YOTHENY1=Y1-YB:X1=X1+XB
520 IFYO-Y1=>YBTHENY1=Y1+YB:X1=X1-XB
530 IFX1<=XM+ABS(XB)GOTO340
540 YO=YO-YB:XO=XO+XB:IFYO>0GOTO310
550 TONE50,500:A$=INSTR$(4):IFA$="QUIT"THENCOLOR0,3,4,7:END
560 IFA$="TYPE"GOTO580
570 GOSUB200:GOTO550
580 A$=PORTS(A$,9):PRINTCHR$(27);CHR$(20);
590 PRINT"          1         2         3         4         5
        6";
600 PRINT"         7         8         9         0         1"
610 PRINT"    123456789012345678901234567890123456789012345678901
1234567890";
620 PRINT"1234567890123456789012345678901234567890123456789012345678901234
630 :FORY=YMTO1STEP-1:PRINTYM-Y+1;TAB(4);:FORX=0TOXM
640 PRINTCHR$(CC(C(POINT(X,Y))));
650 NEXT:PRINTCHR$(27);CHR$(30):NEXT:A$=PORTS(A$,0):GOTO550
990 REM PATTERN FOLLOWS
1000 DATA!!347  FLEUR-DE-LIS AND LION
1010 DATA%,40,0,20,21
1020 DATA+C,1,1,-C,11,1,↑C,1,12,/C,11,12
1030 DATA+C,21,1,-C,31,1,↑C,21,12,/C,31,12
1040 DATA+A,5,5,-A,11,5
1050 DATA+B,24,3
1060 DATAA  HALF FLEUR-DE-LIS
1070 DATA0000002
1080 DATA0303002
1090 DATA0010002
1100 DATA0303022
1110 DATA0000022
1120 DATA0000022
1130 DATA0222022
1140 DATA2222222
1150 DATA2002222
1160 DATA2000222
1170 DATA0200222
1180 DATA0000022
1190 DATA0020222
1200 DATA0022202
1210 DATA0000002
1220 DATAB  LION
1230 DATA000000333000000
1240 DATA000003333300000
1250 DATA000333003330000
1260 DATA000333333330000
1270 DATA000003333330330
1280 DATA330033333303003
1290 DATA330003333003003
1300 DATA033333333003000
1310 DATA003333333000300
1320 DATA000003333300030
```

```
1330 DATA033333333333030
1340 DATA033000333333330
1350 DATA000000333333300
1360 DATA000000333333300
1370 DATA000333330033300
1380 DATA000330000033000
1390 DATA000000033330000
1400 DATA000000033000000
1410 DATAC  MATRIX QUADRANT
1420 DATA03330000111
1430 DATA32000111000
1440 DATA30011100000
1450 DATA30110000000
1460 DATA00100000000
1470 DATA00100000000
1480 DATA01000000000
1490 DATA01000000000
1500 DATA01000000000
1510 DATA10000000000
1520 DATA10000000000
1530 DATA*
OK
```

```
1000 DATA!0376  PARALELLOGRAMS
1010 DATA%,4,4,-8,8
1020 DATA+A,5,1
1030 DATA+B,7,3
1040 DATA+C,1,5
1050 DATAA
1060 DATA0001111
1070 DATA0011110
1080 DATA0111100
1090 DATA1111000
1100 DATAB
1110 DATA0002222
1120 DATA0022220
1130 DATA0222200
1140 DATA2222000
1150 DATAC
1160 DATA0003333
1170 DATA0033330
1180 DATA0333300
1190 DATA3333000
1200 DATA*
OK
```

```
1000 DATA!1234 HEXAGONS
1010 DATA%,20,-10,10,15
1020 DATA+A,1,6,-A,8,6,↑A,1,11,/
A,8,11
1030 DATA+B,11,1,-B,18,1,↑B,11,6
,/B,18,6
1040 DATA+C,11,11,-C,18,11,↑C,11
,16,/C,18,16
1050 DATAA
1060 DATA0000111
1070 DATA0001111
1080 DATA0011111
1090 DATA0111111
1100 DATA1111110
1110 DATAB
1120 DATA0000222
1130 DATA0002222
1140 DATA0022222
1150 DATA0222220
1160 DATA2222200
1170 DATAC
1180 DATA0000333
1190 DATA0003333
1200 DATA0033333
1210 DATA0333311
1220 DATA3333311
9999 DATA*
OK
```

```
REM  DRAGN

20 CLS:KK=0:A$="0347":GOSUB200:C
LEAR50:DIMX$(41),D(1),W(1),C(3),
CC(7)

1000 DATA!7346
1010 DATA%,82,0,0,82
1020 DATA+A,1,1,-A,42,1,↑A,1,42,
/A,42,42
1030 DATAA
1040 DATA0000000000000000001111000
0000000000000000
1050 DATA0000000000000022222211100
0000000000000000
1060 DATA0000000000022330011110000
1110000000000000
1070 DATA0000000033333000000000010
0000100000000000
1080 DATA0000000332330000000000010
0111000000000000
1090 DATA0000033223000000000200333
3100000033300000
1100 DATA0000333230030303000223333
3133200000300000
1110 DATA0000332300033333000322333
3333320003333000
1120 DATA0003323000000330033303333
3333320000330100
```

```
1130 DATA0103323033303330033333013333320033330100
1140 DATA0103332333333000330001333033200003001 00
1150 DATA0103333233333033000013333003320033301000
1160 DATA0013333323000003001333000033200333221000
1170 DATA0011333322003333010000003332003322332000
1180 DATA0331333333320003300100033320022223333200
1190 DATA0331333333320333300003333200233333333200
1200 DATA0033000333320003333003332022333333332300
1210 DATA0033300033332000333333333333333300332300
1220 DATA3003300033332000063333333333300033233303
1230 DATA0333300033322003300003333333300032233030
1240 DATA3333003333320003301100333330003323330333
1250 DATA0330033333200233003010000000003233300330
1260 DATA3300333322002203013000600000032333003303
1270 DATA0000333200023233033000000100032330033300
1280 DATA0003333200003330130000000100032330033000
1290 DATA0033332001003331330300001110032333033000
1300 DATA0033332001013333330303301210003233333300
1310 DATA0033332000113313300030001210000321133300
1320 DATA0033332001003133000333022300000222133000
1330 DATA0033333230000123300033002330033300231300
1340 DATA0033333233301123333333023000333333233100
1350 DATA0003333323330023333002330033330332330000
1360 DATA0003333332332223110000233030303032333000
1370 DATA0000333333223331300000233000000323330000
1380 DATA0000033333333311300000233300000223300000
1390 DATA0000003333311133303000023300003323000000
1400 DATA0000000033333333303033002333332330000000
1410 DATA0000000003333303300330002332222300000000
1420 DATA0000000000000003333333000223330000000000
1430 DATA0000000000000003333300000000000000000000
1440 DATA*
OK

1000 DATA!0713  CHINESE PUZZLE
1010 DATA%,24,0,0,24
1020 DATAA
1030 DATA2222000000222
1031 DATA2222000000222
1032 DATA2222000000222
1040 DATA2222222222222
1041 DATA2222222222222
1042 DATA2222222222222
1050 DATA0002220000000
1051 DATA0002220000000
1052 DATA0002220000000
1060 DATA0002220000000
1061 DATA0002220000000
1062 DATA0002220000000
1070 DATAB
1080 DATA0000001111111
1081 DATA0000001111111
1082 DATA0000001111111
1090 DATA1111111111000
1091 DATA1111111111000
1092 DATA1111111111000
1100 DATA0000001111000
1101 DATA0000001111000
1102 DATA0000001111000
1110 DATA0000001111111
1111 DATA0000001111111
1112 DATA0000001111111
1120 DATA+A,10,1
1130 DATA+B,13,10
1140 DATA/A,4,13
1150 DATA/B,1,4
1170 DATAC
1180 DATA3330000003333
1181 DATA3330000003333
1182 DATA3330000003333
1190 DATA3333333333333
1191 DATA3333333333333
1192 DATA3333333333333
1200 DATA0003330000000
1201 DATA0003330000000
1202 DATA0003330000000
1210 DATA0003330000000
1211 DATA0003330000000
1212 DATA0003330000000
1220 DATA+C,-2,13
1230 DATA/C,16,1
1240 DATA*
OK
```

I presume that anyone using assembly language has some sort of monitor allowing patches to be made to programs and saving them onto tape. Since most Interact owners have only a single printer, and don't change baud rates, the first patch to make is to set the baud-rate that you normally use into location 5330h; 5331h should be set to 0. Pencil into your Assemblex/Editext manual on page 10 the following changes to the Baud-rate table: 110-FD, 134.5-CF, 150-B9, 300-5C, 600-2E, 1200-16, 2400-B, 4800-5, 9600-2. Now, if you do use the "B" command, you have a list of appropriate values. Patch ito Editex the code for your baud-rate at 5330h. Remember, Assemblex has no printer initialization code, so you must print something with Editex before attempting to print with Assemblex.

Patches to Editex: (sets same modes as PORTS in BASIC)

Baud-rate: (see above)

5300h	2	(e.g. 9600 baud)

Interface initialization routine:

Micro-Video			Slaugh		
5288h	3E	MVI A,83	3E	MVI A,3	
5289h	83		3		
528Ah	32	STA C003	32	STA FFF6	Reset interface
528Bh	03		F6		
528Ch	C0		FF		
528Dh	2A	LHLD 5330	3E	MVI A,1	7bits+even+no interrupt
528Eh	30		01		
528Fh	53		32	STA FFF6	
5290h	22	SHLD C000	F6		
5291h	00		FF		
5292h	C0		3A	LDA 5330	Baud-rate
5293h	AF	XRA A	30		
5294h	32	STA CD04	53		
5295h	04		32	STA FFF5	
5296h	C0		F5		
5297h	3A	LDA C003	FF		
5298h	03		00	NOP	
5299h	C0		00	NOP	
529Ah	E6	ANI 7F	00	NOP	
529Bh	7F		00	NOP	
529Ch	32	STA C003	00	NOP	
529Dh	03		00	NOP	
529Eh	C0		00	NOP	

Character output routine:

52A1h	3A	LDA C006	3A	LDA FFF6	
52A2h	06		F6		
52A3h	C0		FF		
52A4h	E6	ANI 10	E6	ANI 08	
52A5h	10		08		
52A6h	CA	JZ 52A1	C2	JNZ 52A1	
52A7h	A1		A1		
52A8h	52		52		
52A9h	3A	LDA C005	3A	LDA FFF6	
52AAh	05		F6		
52ABh	C0		FF		
52ACh	E6	ANI 20	E6	ANI 02	
52ADh	20		02		
...					
52B2	32	STA C000	32	STA FFF7	
52B3	00		F7		
52B4	C0		FF		

Patches to Assemblex:

Character output routine:

	Micro-Video		Slaugh	
4AEBh	3A	LDA C006	3A	LDA FFF6
4AECh	06		F6	
4AEDh	C0		FF	
4AEEh	E6	ANI 10	E6	ANI 08
4AEFh	10		08	
4AF0h	CA	JZ 4AEB	C2	JNZ 4AEB
4AF1h	EB		EB	
4AF2h	4A		4A	
4AF3h	3A	LDA C005	3A	LDA FFF6
4AF4h	05		F6	
4AF5h	C0		FF	
4AF6h	E6	ANI 20	E6	ANI 02
4AF7h	20		02	
4AF8h	CA	JZ 4AEB	CA	JZ 4AEB
4AF9h	EB		EB	
4AFAh	4A		4A	
4AFBh	79	MOV A,C	79	MOV A,C
4AFCh	32	STA C000	32	STA FFF7
4AFDh	00		F7	
4AFEh	C0		FF	

Initialization routine:

4AD0h	3E	MVI A,83	3E	MVI A,3
4AD1h	83		03	
4AD2h	32	STA C003	32	STA FFF6
4AD3h	03		F6	
4AD4h	C0		FF	
4AD5h	2A	LHLD 4AE9	3E	MVI A,1
4AD6h	E9		01	
4AD7h	4A		32	STA FFF6
4AD8h	00	NOP	F6	
4AD9h	00	NOP	FF	
4ADAh	00	NOP	C3	JMP 4AE7 (or NOP through 4AE6)
4ADBh	AF	XRA A	E7	
4ADCh	32	STA C004	4A	

If your output comes out double spaced, the following patch is also required:

4B07h	FE	CPI 0	FE	CPI 0D
4B08h	00		0D	
4B09h	CA	JZ 4B0C	CA	JZ 4B12
4B0Ah	0C		12	
4B0Bh	4B		4B	

THE HARD FACTS OF LIFE
FIRE BUTTON EXTENDER CABLE
By Ron Kregoski
With Lora A. Leggett

This cable is used to prompt the computer to advance a running program enabling you to control the advance of the program and to be up to nine feet from the computer.

The cable comes with a male and a female end. Cut off the male plug Cut all lead wires from the cable except the gray and the white wires which will be connected to the push button switch. The Fire is made from Radio Shack parts so they may be easily obtained. For the housing of the switch, use the plastic case from a 1/4 inch Male Phone Plug (Cat. no. 274-1271) made by a company called Archer. The push-button switch is the type that is on when you hold down the button but goes off when released (Cat. no. 275-609.) Connect the gray and the white wires to the switch and tie a knot in the end of the cable so it will not pull out.

The push-button switches and the phone plugs come two to a pack though you need only one of each for the Fire Button Extender Cable. The cable that we used is available from Rhino Sales, 9880 E. Grand River, P.O. Box 929, Brighton, MI 48116 Telephone Area code 313-227-1788. 9-Foot Extension Cable, (Cat. no. 4004) sells for $6.95. The total cost for making a cable is about $13.00.

ADD FM RADIO SOUND TO YOUR INTERACT
By Ron Kregoski

An F.M. receiver which comes in a cassette type package is a great addition to our Interact since our computer gives us the ability to play music through our cassette unit on the TV speaker while a program is running. This radio receiver fits in any standard cassette recorder. You merely insert it into your machine and press the Play button or in our case, the READ button. There is a tuner on the radio cassette to select the station. A thin wire sticks out of the cassette unit to serve as the antenna for the radio.

The F.M. radio is available in the radio department of K MART and sells for about $24.00.

ADVERTISEMENT

FOR SALE---$300.00 16K Interact, LEVEL II BASIC, Leonardo Graphics, EZ Edit, Edu-BASIC, Edu-BASIC Overlay, FORTH, Micro Video Monitor, HI-LOW Monitor, AL, Alien Invasion, Pack Rat, Breakthrough, Microchess, Dogfight, Star Track, Mazes And Monsters, Guide To ROM Subroutines Basically Speaking, one Joy Stick
Write or call Nick Bulka, R.D.2, Box 1098, Putney, VT 05346
Phone 802-257-7377

Interact Owners- We have a new catalog that is full of software for your computer! Our new catalog has over 20 different programs that will run on your computer! Even if you have received an old catalog write for this new one! Catalog is free. Sample MGH Software Newsletter and sub. information is $1.00. MGH Software, Dept. 2, Box 645, Bayfield, WI 54814

SUPER MEMO FILE- Store, modify and instantly retrieve 8K of messages Cursor allows insertion and deletion of characters. Message can be modified without retyping. Each message can be up to 150 characters plus 17 for each key word. Character set allows 25 characters per screen line.
Send $11.00 to Sol Steinberg, Apt. Q9, Hyde Park Apartments, Bellmawr, NJ 08031

WANTED: Original copy of INTERACTIONS Vol. 1 no. 1 the first magazine published in 1980. Please call or write before sending the magazine. The first person who is able to supply Vol. 1 no. 1 will receive a free subscription to INTERACTION INTERNATIONAL Vol. V for the 1984 year. Contact George A. Leggett, 20562 Woodward, Mt. Clemens, MI 48043 Phone 313-791-4243

The program CRAZY CARDS, which was announced in the Advertisement Section of Vol. IV no. 2 is no longer available for purchase. There is nothing wrong with the program--we think it's great and so do those who have played the game, especially since it plays according to Hoyle where versions of the game sold for a popular video game system do not. We think you'll enjoy it so much that it will appear in a future issue of INTERACTION INTERNATIONAL. That is my right as publisher after all and so I will share what I thought once was of a quality to market rather than publish will end up being published--the reason for which it was created in the first place--and since no one tried it out anyway it will still be an INTERACTION INTERNATIONAL Exclusive!!!

QUALITY PROGRAMS FOR THE INTERACT
from: David J. Schwab
10 Jay Lee Court
Ann Arbor, Michigan 48104

SKETCH PAD - Basic program with extensive machine language subroutines for creating, modifying, and saving screen displays. Draws open and filled circles (round ones!), triangles, rectangles, lines, and letters with super-fast joystick positioning. Saves screen on tape with or without stop code (to create program banners). Hours of fun for all ages.....$8.00

EDU-BASIC OVERLAY - Allows for PEEK, POKE, and USR type facilities in EDU-BASIC. Also for use with Slash U80 port to direct output or listings to port. Use this powerful language to its full potential.....$8.00

QUEST in EDU-BASIC - An 8k adventure program. You must retrieve a treasure from an underground maze. Descriptions are given of each room and you have 6 directions in which to try to proceed. A pirate lurks in the maze and may steal the treasure back.....$5.00

8080 DISASSEMBLER in BASIC - This Basic program lists adresses, contents, corresponding ASCII character, and standard 8080 mnemonic assembler language op codes and registers for any memory locations. Includes complete instructions and sample output listing form.....$5.00

```
10 REM                QUEST IN EDU-BASIC BY D.J.SCHWAB
20 REM                 FROM A PROGRAM BY ROGER CHAFFEE
30 REM
40 REM   THE PROGRAM TAPE FOR QUEST CONTAINS TWO FILES.  THE FIRST FILE
50 REM   IS THE EDU-BASIC PROGRAM AND THE SECOND IS A DATA FILE CONTAI-
60 REM   NING THE NODE INTERCONNECTIONS.  TO RUN THE PROGRAM:
70 REM
80 REM        1. LOAD EDU-BASIC.
90 REM        2. LOAD QUEST PROGRAM WITH LOAD COMMAND.
100 REM       3. START PROGRAM WITH RUN COMMAND BUT LEAVE READ BUTTON
110 REM          IN DOWN POSITION.  QUEST PROGRAM WILL READ DATA FILE.
120 REM
130 REM    IF YOU WANT TO START OVER WITHOUT RELOADING THE PROGRAM AND
140 REM    DATA, DELETE LINE 400 TO INHIBIT DATA FILE LOADING AND RESTART
150 REM    WITH A RUN COMMAND.
160 REM
170 REM    IN QUEST SOME PASSAGEWAYS, INCLUDING DEAD ENDS, OPEN AND CLOSE
180 REM    DEPENDING ON WHETHER YOU ARE CARRYING THE TREASURE FOR THE
190 REM    FIRST OR SECOND TIME SO DON'T BE AFRAID TO TRY THE SAME PASSA-
200 REM    GEWAYS AGAIN.  IF YOU WANT TO KNOW MORE ABOUT THE PROGRAM,
210 REM    CONSULT BYTE MAGAZINE JULY, 1979 OR SEND SPECIFIC QUESTIONS
220 REM    WITH A SELF-ADDRESSED STAMPED ENVELOPE TO ME.
230 REM
240 REM                        HAPPY HUNTING!!!
Ok
```

INTERACT EXPANSION PRODUCTS

Features:

Does not require major surgery to the Interact - no soldering of wires to the printed circuit board or cutting of circuit traces - The Interact can be quickly restored to original condition
Expansion not restricted by space inside the Interact
Allows for memory expansion up to nearly **133K**
32K RAM card (one such card brings total Interact memory to 48K)
has totally hidden refresh - no wait states as the case for the resident 16K or other internal expansions - this means that programs written to this block execute faster
Expansion frees memory space wasted because of the original Interact design

Supports up to four 5 1/4 inch floppy disc drives

Supports conversion to S-100

The expansion consists of an expansion interface board (IE) which buffers and brings out from the Interact enclosure the necessary signals on a 40 conductor ribbon cable. The ribbon cable is in turn plugged into a motherboard (IMB-2) which is housed inside an enclosure external to the Interact. Expansion products such as the IMEM-1 32K RAM card are then plugged into the motherboard.

Prices of bare boards for products available now:

IE Expansion Interface Board	$25.50
IMB-2 Motherboard	$24.50
IMEM-1 32K Memory Board	$44.50
IEN-1 Enclosure	$19.95

The above products are also available in complete kit and assembled forms. See below for address to send for literature.

The floppy disc and S-100 hardware are presently in existence in prototype form. If you are already on our mailing list, you will be notified when the finished product is available. Otherwise send a business-sized self addressed stamped envelope to:

Walter H. Jopke Jr. 5016 West 99th St. Bloomington, Minn. 55437

Also available INTERWORD Word Processing Software

1-$59.50, 2-$49.50 each, 3-$39.50 each, 4-$30.50 each,
5-9 $25.50 each, 10 or more - $19.50 each

To take advantage of the quantity price advantage for Interword, you must supply the names and addresses for all persons involved.

INTERACTION INTERNATIONAL
GEORGE A. LEGGETT
20562 Woodward
Mt. Clemens, Mi. 48043

FIRST CLASS

$ 3.00 A COPY
$15.00 FOR 6
ISSUES A YR.

INTERACTION INTERNATIONAL

A MAGAZINE FOR INTERACTORS AND FOR ALL PEOPLE WHO WANT THE COMPUTER KNOWLEDGE OF TOMORROW, FOR TODAY

SEPT.-OCT. 1983 VOL. IV NO. 5

TABLE OF CONTENTS

CREDITS

Editor/Publisher	George A. Leggett
Features Editor	Lora A. Leggett
Draftsman	Daryl Waananen
Programmers:	Ian Ippolito
	George A. Leggett
	Warren J. Moore
Production	George A. Leggett
Transcription	Lora A. Leggett
Printer	Ted McAdam

IT'S LONELY AT THE TOP
Or,
THERE'S GOT TO BE A MORNING AFTER
Publisher's Statement
By George A. Leggett

For the past four issues of this magazine I have been trying my best to inform and entertain you as much as I can. I appreciate and thank the countless compliments I receive on this task that has fallen before me; a task which I took willingly and do not regret in the least. However I am a bit surprised at some of the events since I took over this publication. The title above is a two-part title. I shall try to explain how I feel and how this came about.

First of all, It's Lonely At The Top. By that I mean I really thought there'd be a lot more support in contributions. Now before I rile up 8 people out there and you know who you are who have submitted great work, I do not mean them. Their names have appeared throughout the year and a few more names will still appear. If you're wondering where these other pieces of work are, they're coming, they're in reserve. Publishing a magazine is a lot like playing poker or at least a lot like the gambler song. You've got to know when to hold 'em and know when to throw 'em know when to walk away and know when to run...Of course every time I load up a program I'm running...anyway, the point is that all the work I have received has come from a small handful of people. This is terrific because without them we would have a magazine called George A. Leggett on the Intefact. You will only see my work if I don't get more of you out there to turn in some programs. Let me tell you right out where we stand now. I have enough material for two more magazines if I put it all in. That means I do nothing else and play all my trump cards as it were. Which would be foolish. This appearw to be a gloomy prospect. One issue will be out I hope by November. We do our best, we're running a bit late as you are aware by now. We all obviously have personal problems but we get the work done. So that leaves me with one more issue of material and then what? I know I can count on my 3 or 4 super contributors out there. But that's not fair. By no means do I want you to stop. If anything, keep the good material coming. But it's not fair that there are over 125 of you out there now only 8 are donating all the work!

The excuses are many and varied but the biggest excuse which is a crock is "I'm only a beginner. I'm only an amateur. What could I have?" Well you can't tell me you haven't written one program in your life that doesn't have a bug in it! I could be wrong but there must be at least 50 of you out there who have written something. One program in this issue in fact, is a 9-liner counting the REM statement and one line is a clear screen. So it's a 7-line program. And it's fantastic, there's nothing wrong with it. Do you think I'm some big bad Wolf up there whereby I only gobble up 48K programs? I may have to do that in my work but that's beside the point. I am still intrigued and amazed at the power of a 4K program. In this day and age of running 64K RAM and 2 disks and hard disk and what have you we are losing sight or at least I do of what a 4K program can do. This was mentioned to me by Mr. Moore who has two contributions in this issue, some of my trump cards which I referred to above. Four or five of you wrote me asking what I think of spread sheets and Video Calc and what have you. This program, in my oppinion, is as good as any of the rest for what it does in 4K minus the bar graphs or line graphs which are not that important anyway. This program is terrific if you want spread sheets. I have yet found out why I would want one but if you want one, here you are! This type of work is of course superior and so many of you would say "Gosh I could never do anything like that so hang it up." I disagree you could never do it, I believe you could. But that's beside the point. I do believe you could turn in a 10-or-20 line program that would amaze us. So let's say 50 of you could turn in a little short program and my fantastic 8 people can handle the 4K programs. That leaves us with about 70 other people. I'm sure some of you can write product reviews. There are so many products on the market. There's more software than we could write about in 8 years! You could write about what you're using an Interact for, how you're using programs that you find in this magazine or other magazines. (as if there were any other!) And how about other systems?

I guess it comes from growing up a hard way and not to get too heavy or personal, but I could not walk till I was 8 years old because of my cerebral palsy. It was more so because of my fear. I could walk with someone holding my finger. They'd let go of it and I'd fall. A stupid thing now some 22 years later. But the key word was fear. If you fear doing something, believe me, you're not going to do it. I hate to use an old cliche or quote here but...what the hell! "The only thing we have to fear is fear itself." You should all know who said that but for those of you who didn't, good old FDR. But it is true! If you're going to think your little 8-line program is a piece of you-know-what, everyone is going to think that. But if you think your little piece of work is the best thing anyone ever did or that you ever did that's all that matters. Send it in!! You've got nothing to lose. What you're going to gain is this: You're going to gain the experience of many people like you ...I know myself that I only come up with one or two brilliant lines in a whole program. The rest just does what it has to do. But one or two lines are super fantastic. When you multiply up those few great lines, you get hundreds of great thoughts. For example, in the TELESCOPE Program by Mr. Ippolito he combines his randoms with his Plot statements. Refer to line no. 40 and 60. I always put X = Random, Y = Random, C = Random, Return. Call it as a GOSUB and Plot X Y and C.

But if you're only going to use it once or twice it may not merit the memory. So this unique way is different. This is what I mean by that one brilliant idea. No one's going to give you a million dollars for it but it is something to be proud of.

Another thing in this issue is Mr. Moore's CMD. My overlay was done in all Hexadecimal to show you programmers what Hex number went in. He converted it all to Decimal. It runs in only 7 seconds, super quick. I thank him for all that time and energy. Now you can have it in the machine and running fairly quickly. So the point being that you have the same problem and yet there were two ways to approach it. Both ways are applicable.

If I sound harsh or irritable I am not. My wife points out that it may be true....however, it's because I care. I care about the future of this publication, I care about giving you what you want, I care about giving you information you can use. I care. I know you'll all do your best to make the 1984 year another great fine super stupendous and all that good year for all of us.

Now to the second part of my editorial title There's Got To Be A Morning After. This will be a lot harder to explain. Since I began working on this magazine last November many things have happened. In February I got a great opportunity to work with a man and his company on some commercial work using the Interact. He gave me a great break in my life and I thank him for it. I also wish to thank the man who was responsible for getting me in touch with the man; Mr. Dave Ross. Thank you, Dave! So as you see, Mr. Ross is not my enemy as many of you think he is. We were talking recently in his office about the fact that somehow he pictures himself and his company as "the bad guy". From the point of view of the Interactors. After a long meeting and a walk through his offices or building or computer center or whatever you would address it as, which were new to me, and I must break in and say that anyone in the area of Ann Arbor, MI, I'm roughly 70 miles from there, please drop in. I'm sure he would agree to my saying that he would be most delighted to show you some of the new Micro Video games and what's cooking. It's a very nice place to spend some time and the people were most hospitable and did their best to make me feel at home. I have no complaints against Mr. Ross except for one, but hey, we all can't be perfect. It's a very touchy area. I'm not even sure I should touch it. But I will.

The one thing is simply his policy. I hope I don't get in trouble for saying this but I must say at least this. I began the year by saying I would be as honest and open as I could with all Interact information. As the year progressed this became harder and harder to do to a point of near impossibility. As Mr. Ross plainly said, I am in two worlds; one with you in the Interact world the so-called amateur world as you would say. I also have a foot in the professional programming world the commercial world. According to Mr. Ross, never the twain shall meet. As I told Mr. Ross and am telling you, I strongly disagree with this but I respect the viewpoint of Mr. Ross. One can disagree and respect at the same time. I point this out because I would have never said this ten years ago. If you think I'm bad now, you should have known me in my late teens! Oh, boy!

The older that you get you've got to eat, you've got to pay the bills, and you've got to make sacrifices. I don't like any of this, it rips at my guts, as it were, but many of you call me and ask about things and I am just not at liberty to tell you about these things whatever they may be. Not that I wouldn't love to because there are some really marvelous things happening. I just hope and urge Mr. Ross to try and see it our way. I think everyone should have an equal right to knowledge. Oh, they should pay for it, yes. He's a businessman, I am and hey, let's face it, the buck... you must feed the wife and kids and that's where it stands. For business people and for that matter anyone. But the computer business is a very cut-throat tough business They can say what they want about secrets in war time. But I tell you many secrets in computer time are far more devistating not to lives but to economy. I was not aware of this till recently, I disagree with it, but such is life. Enough of this bitter taste of life, let's get down to some more shocking news. Hey, I'm out for you!!

There's one more thing I'm going to announce now. You'll see it in the Advertisement section and on the single paper included in this issue. The single sheet is a renewal form for the 1984 year, which some of you have already hopped on the bandwagon. I thank you at this time. Good news and bad news. The good news is--this year there will be a rate decrease. Yes, while everything else goes up, we are coming down. The price will be $12.00 for the year and magazines will continue to be sent First Class. However, as the saying goes, you never get something for nothing. I regret to say that there will be only four issues for 1984. This was a long heavy-hearted decision. It was based on many things: as I told you above, not enough materail, I must work between worlds,. This does not mean that we will not submit the same amount of work or shrink the magazine. As those of you who talk to us on the phone know that it's only me and my wife and frankly my wife does 90% of the typing. You know it's my typing if it's on Teletype or the printer. Hers is on the Remington typewriter. She don't complain much but it is a lot for her. We want to spend more time with the kids and I guarantee now that I spend more time with the Interact than anyone out there. If there's anyone that spends more, pray tell me,... Good luck to you. Because I'm spending when I get in full swing maybe 70 hours a week. A lot of time behind an Interact keyboard! This is not counting the work for INTERACTION INTERNATIONAL. What little time I do have is for the magazine and after 70 hours, it's a little hard to get into the groove of writing programs for the magazine. So as much as one loves the Interact, you can have too much of a good thing. Now, we are not going to cut down on the pages. In fact if material comes into us, we will beyond 36, 40, 44 or whatever we have to do to print all the material we get. And we better get it, right? Right! So you will probably if material is submitted get more but it gives us a little more time to get it together. A month may seem like a very long time to you but with all the hours on work, the kids, the house ...

And I'm very proud to say that this fall, a quick personal note here, my Grandmother and Grandfather Leggett are visiting my parents from England. They only live 5 miles away (the parents, not the grandparents!) I would like to spend as much time as possible with them. They're responsible for much of what I am now. I'm laying out what I'm doing and why. I don't have to, but as I said, I want to be as honest as possible.

So fill out the renewal form and please answer all the questions. Only a few of you received one last year but many of you left things blank. There is no way I can know what you want unless you tell me. Just because it's blank, it's not meant to stay that way!! Looking back at Mr. Cook's questionnaire, he was easy on you. I'm asking a lot of questions because I want to know more so I can do more. Which leads to the last point of this discussion. What To Program? I simply do not know what in the heck many of you want. Some of you don't want games, some of you don't want graphics. Many of you by your calls and letters want more technical information. Apparently you think I have a way with the word. I thank you. I enjoy it. I've told some of you on the phone that I think I'm boring you. You disagree. I still think that. Maybe it's because I do it every day and I assume everyone knows it. My timer keeps track of how many hours I put in my computers in one year. I'm not sure of the exact number thus far, but it's some outrageous number like 2500 hours. You people with 40-hour work weeks have it made! Many of you say anything's fine. I don't know Anything. I know Something about Everything. Anything you want to know about a particular Poke or ROM location or RAM or port or BASIC--anything but FORTH I'm sorry to say, I guarantee that I will answer it for you. If I don't know it (I'm in trouble if I don't) but if not I will find the answer. But the "anything's fine with me" response or no response at all is no help. I found out that the majority of you did not want to see games. Mind you that "majority" is only a handful of about 10 or 20 people who have kept in contact. It's like the Nielson ratings where only a handful of people are asked and a show is canned or continued. Of course over 90% of the population has never been asked what they think and a few people control what goes and what stays. Not to get political but it's just like the government. They ask us to vote and only a small handful of people vote. Then everybody complains about wwho got in office. If they all felt that way, he never would have gotten elected! It's true in every facet of life--if you're not part of the solution then you're part of the problem as my wife and I heard many times in high school. So I hope I didn't lose all of you. Out of 125 subscribers I'll probably end up with two--me and my wife! But hey, how you do this now and for the next few month is really going to affect things for 1985 as in will there be a 1985 magazine? I was committed in 1982 for a 1984 but boy, I'm warning you that unless I see more input and more material--George Orwell may have warned you about 1984 but George A. Leggett is warning you about 1985!!!

So with that I shall now discuss this issue in greater detail. I have already discussed the programs by Mr. Moore and Mr. Ippolito. I shall discuss my own contributions--Machine Shop Talk and Hard Facts.

The Machine Shop Talk project was demanded by about 10 people who went out of their way to call and write. If 115 of you are mad..... It's never been done before because it takes so many pages! It was a gamble and I hope you will all enjoy and learn from it. It's a Machine Language program fully documented. I'm not the greatest game programmer in the world but in my oppinion a good tutoring program. The write-up didn't come out the way I wanted. It's not the fault of Lora, it's just that I'm not happy with it. I wanted it different but it would have taken 30 pages which is totally unacceptable. So if you don't like it, tell us. If you love it, tell us. I don't care, just tell us why you do or don't. It was a very hard thing because doing it on paper with notes and then getting it to the press is two different things. I read it on tape byte by byte and you can look at the listing and see how long that is. Lora has spent an enormous time typing it up and that is a big reason why this issue is running late. So we are anxious to hear what you think of Machine Language programs of this type and whether you would like to see more of this type of thing in the future.

The project in Hard Facts is a very useful thing to me. I use it every day and that is why I put it in there. This was my last ace in This is all I have that is simple for the Hard Facts. I have one more project dealing with remote control for the Interact that I may consider for publication in the future but right now I have not the time to work with it and it may be too complex and lengthy to merit the time or space involved. We shall see. If you want to see Hard Facts continue you will have to submit material. Or, I will have to cancel The Hard Facts Of Life for 1984. Both Hard Facts and Machine Shop Talk need your help to survive. Although Machine Shop Talk is different in that I only ask for your ideas while Hard Facts needs ...well, hard facts!

One final thing: Y ou will notice that the subscription renewal form is a separate sheet from the magazine. I wanted it this way because I didn't want none of you ripping up my book. I do not like it in other magazines and certainly not in mine. If I could, my books would be hard cover. Obvious, we can't. I resent publishers who put a renewal form or questionnaire on the back of something you want to keep. I never do that. I recopy the page and send it in to them. I just refuse to cut up a book I paid money for or even one I didn't pay for. Books are a precious thing to me. (Take a gander at the movie Fahrenheit 451 if you have a chance. A classic) Thus, a separate subscription renewal form. So,

D O N ' T R I P M Y B O O K ! ! !

So there you have it, everything I had to say at once. I know it took a lot of pages and as you can see I have never had trouble in my life expressing the way I feel. I hope everything is okay with all of you and I hope to see you back in 1984 when the questions are asked: Will Interact Be The Big Brother? Who knows? You have the power! Let's see it happen in one way or another!

Sincerely,

George A. Leggett

INTERACTORS INPUT
Compiled By Lora A. Leggett

Many thanks for all of the calls and letters we have received during the past year. In this Interactors Input section, I shall attempt to answer some of the questions you have asked of myself snd George. Our appologies for not being as prompt about getting a note back to all of you but as you know time has a way of flying...

First, Mr. R. L. White of Union, MS asked about the extended Plot statement in the program SCREEN GRAPHICS CALCULATOR which appeared in Vol. IV no. 3. If you do not have the Microsoft 8K Fast Graphics BASIC, you can substitute the CMDB command in George's CMD overlay in that same issue and it will work exactly the same way that the 5-part plot statement does. The 5 numbers represent the starting point of your X, starting point of your Y, ending point of your X, ending point of your Y, and the color of the line.

Mr. Henry L. Dietz of Cuyahoga Falls, OH wrote to inquire about the availability of the FORTH Language tape by Russell Schnapp. George currently does have a copy of that tape and there was some discussion with Mr. Schnapp about making the product available through us as a distributor. However, we would only be interested in doing this if there is a real demand for this. So, thus far, we do not hold the Master or authorization to copy the FORTH and it would really take a demand on the part of the readers for us to move in that direction. We have received a FORTH contribution which you will be seeing and that is encouraging for those of you who wish to keep FORTH alive and well.

Mr. Ralph Bates of Evanston, IL writes to remind us that we should keep things simple so that even people who do not know a great deal about computer can learn new things. Thank you very much for your comments, Mr. Bates, and we do our best to keep in mind we are not talking to Doctoral graduates. George hopes he is not boring to those who do know all the terminology and who do spend many hours per week with their machines as he does. Also Mr. Bates asked about VIDEO CALC and should he invest in it. It all depends on your needs and if you are really going to spend a great amount of time working with it. You may want to look at the new program being published in this magazine by Mr. Warren Moore. This kind of a question is a bit difficult to answer without knowing your needs. Although we do have a copy of VIDEO CALC we have never really used it. George has shown me many ads in computer magazines for such programs and they have to me seemed a bit overpriced in general for what they are supposed to do. But one must weigh the need and application against what it is worth to him in terms of cost.

Our thanks also to Mr. John Herron of Plymouth, MI for a great letter in response to the Hard Facts column. He proposes a robot for the Interact. He has some very interesting and fun speculation, and it did start our minds whirring, which doesn't take much, and if we have available space, you all might enjoy his letter at some time. Our imagination and ingenuity keeps computers and people evolving.

Several of you have asked us about the availability of the French Victor Lambda computers which are Interacts that we discussed in Vol. IV no. 3. All we can really tell you is this: they are manufactured by a company called MICRONIQUE in France, and they can be made in 16K models or with upgraded memory for American standard TV. However, because of the competitive pricing of home computers today it doesn't seem economically justified to import them for distribution to the home market--they could never win when placed against such things as a Commodore 64. Let's face it--Interact was way ahead for its time in the late '70's, but compared to the computers of the '80's, well.... That's a hard pill for us die-hard Interactors to swallow--we'll always have a special place in our hearts for this madhine and would welcome its French cousin warmly. But--I really have noticed the picture is different when you try to show someone new who is just getting into computer all of what's around. They will undoubtedly take one of the "new-fangled" machines. So that is why I don't think a company would feel warranted in bringing a large number of computers to this country and distributing them out into the world of home computer users. I honestly do not know pricing and feasibility of bringing over just one computer for someone's personal use and I wonder if anyohe really would. I will state here that this is as far as I'm at leberty to continue this discussion for the record and stay within the realm of individual speculation and company policies and politics. It is a very interesting world—I think I'm still glad to live in America although it gets more difficult to feel free to total self-expression the more time goes by and the more you come into contact with the business world at large. Thanks for the tremendous response to that feature--it was one of my favorite stories all year--to me I can't improve on that!

Mr. Earl Graves of Ann Arbor, MI wrote to ask about the Optacon mentioned in last issue's Random Rems. They are manufactured and distributed by a company called Telesensory Systems, Inc. in Palo Alto, CA and visually handicapped individuals can go there for instruction on the use of the Optacon or be trained with it at a Rehabilitation center or other authorized instruction center closer to their home. I was fortunate to find out about a grant program provided by the Mott Foundation. I obtained my machine and instruction at the Service Center For Visually Impaired up in Flint, MI in 1980. The grant provided the machine and a $500.00 training fee was sponsored for me by the Chesterfield Township Lions Club. I went to the center 6 mornings and have to return yearly for an evaluation. Personally, the instruction didn't do as much for me as what I had learned on my own about letter shapes from a magnetic letter said we had at home as children and like the one I use with our own children. The stimulator array is 6 by 24 dots, (6 X by 24 Y) and you can vary the size of the letters with your zoom lens on the hand-held camera. When I last heard, the cost of the Optacon was around $2500.00. George thinks it would be marvelous to have a type of camera pickup similar to the Optacon to enter printed material into memory.

Fianlly, thanks to Mr. Jerry Goerz for his compliments. We do mention our physical limitations in passing, but no, they do not excuse us from accountability. So, if we merit some sharp criticism that is valid, we merit it as much as any other individuals doing the same job! Thanks again, and keep those letters coming!

TELESCOPE PROGRAM

The following is a transcription of a letter and program received from Ian Ippolito. The reason for the transcription is that the submission is too light for the printer to use. If there are any errors or typos, please excuse us. Thank you very much for the program.

May 26, 1983
Ian Ippolito
220 Viscount Avenue
Merritt Island, Florida
32952

Dear Mr. Leggett,

The following short program simulates looking through a telescope. Lines 10 +20 clear the screen and plot out two-hundred stars. Line 40 picks a random number from data in Line 100 to poke into location 24888 (Refer to INTERACTIONS, Vol. II, no. 3, pg.12). Line 50 scrolls the screen a random number of times. Since the stars are moving upward every time a print statement is executed, Line 60 plots out more stars to fill the empty spaces.

This program is most effective if you use a black-and-white t.v or a color t.v. with the color adjusted to act like one. Then turn the contrast up, focus the picture a little off so that the stars shine brighter, and turn the brightness down so that the back ground is a solid black color.

Sincerely,
Ian Ippolito

```
5 REM************      *TELESCOPE**      ************
10 CLS
20 FORX=1TO200:PLOTRND(1)*111+1,RND(1)*76+1,RND(1)*3:NEXT
40 RESTORE:B=INT(RND(1)*14)+1:FORA=1TOB:READC:NEXT:POKE24833,C
45 IFC=33THENPRINT:PRINT:GOTO60
50 FORT=1TORND(1)*10+1:PRINT:NEXT
60 FORR=1TO20:PLOTRND(1)*111+1,RND(1)*76+1,RND(1)*2+1:NEXT
65 GOTO40
100 DATA1,2,3,15,23,27,28,29,30,31,32,32,33,1,2
OK
```

April 15, 1983

W.J. Moore
571 So. Broadway
Pittsburg, Ca. 94565

Interaction International
George A. Leggett
20562 Woodward
Mt. Clemens, Mi. 48043

Dear George

Enclosed is what I believe to be a useful program. Also some different programming concepts. I will prepare an article on on CLOAD* and CSAVE* later to explain and demonstrate a more efficient use of these commands.

MINI-CALC is an electronic spreadsheet for the 16K Interact. It will run in either Level-II or 8K Fast Basic. Following are its main features:

1. Menu Driven
2. Column or Row manipulation
3. Easy entry into any part of array
4. User Specified Ranges - both Column and Row
5. Automatic Column and Row Totals
6. Automatic Viewing
7. Fast Data Tape Load and Save

In order to provide as much memory as possible for the working array, names are not permitted for columns or rows. Some other operations not available are:

1. Numerical formatting (Rounding, fixed decimal)
2. Multiple math operations (One math operation at a time)
3. Assignment of a constant to a column or row

the code has been optimized as much as possible. Due to pokes in line-620, Lines 10 thru 90 must be entered exactly as shown and not altered or program will bomb. (SN error) These pokes will change the math operator in Lines 70 and 80 in order to save additional programming of separate loops for each of the basic math operators.

OPERATION OF PROGRAM

1. INITIALIZING

The first screen after RUN will ask for dimensions of array. If you want to load data previously saved on tape, enter any values to move past initializing. If new array is being specified, try to keep array less than 400 (column x row). There is no test in the program to check on size of array. Basic will simply exit with 'OM' error message.

2. MAIN OPERATING MENU

To return to operations menu from any other screen, enter 'M' and press 'CR'. From this menu, you may select:

1. Input/Change
2. View
3. Calculate
4. Load Tape
5. Save Tape

3. INPUT/CHANGE

You will be prompted for range of rows and then columns. (To specify a single row or column enter same number for 'Range' prompt.) The program will now single step thru specified ranges prompting for input (or change). If no input or change is required, press 'CR'. After last item has been entered, program returns to Operations Menu.

Example: row=1 range=10 col=5 range=7
will prompt cols 5 thru 7 for rows 1 thru 10

Example: row=1 range=10 col=5 range=5
will prompt col 5 only for rows 1 thru 10

4. VIEW

You will be prompted for ranges same as Input/Change. Then 10 cells at a time are displayed until user specified ranges have been completed. Press any key except 'M' to continue viewing. (For more than 10 cells).

OPERATION OF PROGRAM

5. CALCULATE
The first screen prompts for column or row manipulation. Program operates on one or the other only.

The next screen prompts for a column or row. Only one may be used.

The next screen prompts for choice of math operator.

The next screen prompts for another column or row. Again, only one may be used. It may be the same as the first operand.

The final screen prompts for where the answers are to be placed. The selected column or row will overwrite any values previously present.

The program will now perform specified operation on all cells in columns or rows specified and re-calculate new totals for ALL columns and rows. Totals are stored in zero (0) parts of array. The working array is dimensioned as W(C,r).

NOTE: Division by zero is illeagal. Program handles this situation by not performing any operation. (skips over calculation.) No error messages will be given. This is done to preserve all other data.

6. LOAD TAPE Prompts are given to get data loaded. Loading data will always re-dimension array according to data on tape. If you need to know how array is dimensioned (forgot to mark cassette?), follow these steps:

1. Control 'C' (stops program)
2. ?C,R (prints values for columns and rows)
3. CONT (re-enters program)
4. 'M' press 'CR' (returns to operations menu)

7. SAVE TAPE
Prompts are given to save data. Leaderless tapes should be used or the leader should be taken up prior to writing to tape.

THIS SECTION NOT FOR PUBLICATION I used example in MV's VIDEO-CALC only to check on operation of this program. I am off a few pennies here and there due to no rounding but more importantly, program demonstrates ability to do the same job (less printing). It really surprises me that a complex program will fit in 4k of memory! I hope this program is suitable for your fine newsletter.

Sincerely

```
10 REM (MINI-CALC) BY W.J. MOORE
20 CLS:COLOR0,2,3,7:RESTORE:GOTO790
30 N=0:FORJ=ATOU:FORI=BTOV:IFF=1GOTO150
40 IFF=2GOTO100
50 IFP=172ANDS=1THENIFW(A2,J)=0GOTO170
60 IFP=172ANDS=2THENIFW(I,A2)=0GOTO170
70 IFS=1THENW(A3,J)=W(A1,J)/W(A2,J)
80 IFS=2THENW(I,A3)=W(I,A1)/W(I,A2)
90 GOTO170
100 N=N+1:OUTPUT"R  C  VALUE",6,71,1:OUTPUTJ,0,71-N*6,3
110 OUTPUTI,18,71-N*6,3:OUTPUTW(I,J),36,71-N*6,2
120 IFN=10THENN=0:GOSUB320:CLS
130 IFI=VANDJ=UTHENA$="M":GOTO320
140 GOTO170
150 GOSUB350:A$="":INPUTA$:IFA$=""GOTO170
160 GOSUB330:W(I,J)=VAL(A$)
170 NEXTI,J:IFS>0THEN190
180 RETURN
190 FORJ=1TOR:FORI=1TOC:W(0,J)=W(0,J)+W(I,J):W(I,0)=W(I,0)+W(I,J):NEXTI,J
200 RETURN
210 CLS:D=0:N=0:FORI=1TOLEN(T$)STEP4
220 A=VAL(LEFT$(MID$(T$,I,4),2)):B=VAL(RIGHT$(MID$(T$,I,4),2))
230 RESTORE:FORJ=1TOA:READA$:NEXT:IFN=0ORA=0GOTO260
240 IFE=1THENIFN<3GOTO260
250 D=D+1:IFF=1THENOUTPUTD,0,71-N*9,3
260 OUTPUTA$,18,71-N*9,B:N=N+1:NEXT
270 ONGGOSUB370,380,390
280 OUTPUTA$,18,11,3:E=0:RETURN
290 ONSGOSUB300,310:RETURN
300 OUTPUT"COLUMN",18,17,1:RETURN
310 OUTPUT"ROW",18,17,1:RETURN
320 A$=INSTR$(1)
330 IFA$="M"GOTO520
340 RETURN
350 OUTPUT"R=     C=",6,11,2:OUTPUTJ,12,11,1:OUTPUTI,48,11,1:PRINT
360 OUTPUT"NOW",6,11,2:OUTPUTW(I,J),30,11,2:PRINT:RETURN
370 A$="SELECT ABOVE":RETURN
380 A$="HIT ANY KEY":RETURN
390 A$="":RETURN
400 INPUT"    ROW";A:GOSUB330:IFA>RTHEN400
410 INPUT" RANGE";U:GOSUB330:IFU>RORU<ATHEN400
420 INPUT"COLUMN";B:GOSUB330:IFB>CTHEN420
430 INPUT" RANGE";V:GOSUB330:IFV>CORV<BTHEN420
440 RETURN
450 GOSUB210:GOSUB320:RETURN
460 F=0:INPUTA$:GOSUB330:GOSUB480:IFF=1GOTO460
470 RETURN
480 IFS=1THENIFVAL(A$)<1ORVAL(A$)>CTHENF=1
490 IFS=2THENIFVAL(A$)<1ORVAL(A$)>RTHENF=1
500 RETURN
510 T$="1901":F=0:G=2:GOSUB450:REWIND:RETURN
520 CLS:WINDOW23:T$="0101020203020402050106O1":COLOR0,1,2,3
530 F=1:G=1:GOSUB450:ONVAL(A$)GOTO540,550,570,700,750
540 T$="0201":COLOR3,4,0,1:F=0:G=3:GOSUB210:GOSUB400:F=1:GOSUB30:GOTO520
550 T$="0301":COLOR6,4,0,7:F=0:G=3:GOSUB210:GOSUB400:F=2:CLS
560 WINDOW65:GOSUB30:GOTO520
```

```
570 T$="040100001402150Z":COLOR0,1,2,3:F=1:G=1:GOSUB450
580 IFA$="1"THENS=1:GOTO600
590 IFA$="2"THENS=2:GOTO600
600 T$="07010902":COLOR7,3,5,2:F=0:G=3:GOSUB210:GOSUB290:GOSUB460:A1=VAL(A$
)
610 T$="0801100211021202130Z":COLOR7,3,4,2:F=1:G=1:GOSUB450:A=VAL(A$)
620 B=19928:D=19961:P=169:FORI=0TO3:IFA=I+1THENP=P+I:POKEB,P:POKED,P
630 NEXT
640 T$="07010902":COLOR7,3,5,2:F=0:G=3:GOSUB210:GOSUB290:GOSUB460:A2=VAL(A$
)
650 T$="2301":COLOR3,1,7,4:F=0:G=3:GOSUB210:GOSUB290:GOSUB460:A3=VAL(A$)
660 IFS=1THENA=1:U=R:B=A3:V=A3
670 IFS=2THENB=1:V=C:A=A3:U=A3
680 CLS:OUTPUT"CALCULATING",18,41,1
690 FORI=0TOC:W(I,0)=0:NEXT:FORJ=0TOR:W(0,J)=0:NEXT:F=0:GOSUB30:GOTO520
700 T$="0501170218021601050I":COLOR2,0,3,7:F=1:G=1:E=1:GOSUB450
710 ONVAL(A$)GOTO720,730
720 GOSUB510:GOTO700
730 T$="2001":F=0:G=2:GOSUB450:CLEAR
740 DIMS(2):CLOAD*S:C=S(0):R=S(1):DIMW(C,R):CLOAD*W:GOTO520
750 T$="0601170218021601060I":COLOR1,0,3,7:F=1:G=1:E=1:GOSUB450
760 ONVAL(A$)GOTO770,780
770 GOSUB510:GOTO750
780 T$="21012201":F=0:G=2:GOSUB450:CSAVE*S:CSAVE*W:GOTO520
790 WINDOW35:OUTPUT"DIMENSION YOUR",8,71,1:OUTPUT"ARRAY",36,65,1
800 PRINT"    ROWS";:INPUTR:PRINT"COLUMNS";:INPUTC:PRINT"OK (Y)?":A$=INSTR$(
1)
810 IFA$="Y"GOTO830
820 PRINT:GOTO800
830 DIMW(C,R),S(2):S(0)=C:S(1)=R:POKE19215,25:GOTO520
840 DATAOPERATIONS,INPUT/CHANGE,VIEW,CALCULATE,LOAD TAPE,SAVE TAPE
850 DATASELECT OPERAND,SELECT OPERATOR,(USE ONLY ONE),"  +","  -","  *"
860 DATA"  /",COLUMN,ROW,REWIND TAPE,PUT DATA TAPE,IN TAPE DECK
870 DATAPRESS REWIND,PRESS READ,PRESS,READ/WRITE,PUT RESULTS
```

Program Description

```
Lines  30 - 200  Subroutine that handles all array manipulations
lines 210 - 280  Subroutine that handles all screen displays
lines 290 - 510  Misc subroutines
lines 520 - 530  Operations menu screen and logic
line  540        input/change screen and logic
lines 550 - 560  View screen and logic
lines 570 - 690  Calculate screens and logic
lines 700 - 740  Load tape screens and logic
lines 750 - 780  Save tape screens and logic
lines 790 - 830  Initialize and dimension array
lines 840 - 870  Data used in various screens
```

Variables

```
C - dimension of columns
R - dimension of rows
W(c,r) - working array
S(2) - array for tape transfer - s(0)=c s(1)=r
A - row start
U - row stop
B - column start
V - column stop
F - flag
S - flag
P - operator pointer
T$ - codes for display
```

NOTE: Many of above are reused for other purposes. The intent was to keep the variable table down to save memory.

```
1 REM NEW BASIC COMMANDS
2 REM (B-BOX)(F-FILL)(L-LINE)(P-PRINT)(T-TRIANGLE)
3 REM      CMDBXS,YS,XL,YL,C
4 REM      CMDFXS,YS,C,BO   (BO=BORDER COLOR)
5 REM      CMDLXS,YS,XE,YE,C
6 REM      CMDP[illegible]
7 REM      CMDTX1,Y1,X2,Y2,X3,Y3,C
8 REM CSAVE* AND CLOAD* - NO LONGER AVAILABLE
10 :
20 DATA18953,235,205,19,74,205,40,74,205,19,74,201,42,249,74,235,42,251
30 DATA74,34,249,74,42,253,74,34,251,74,235,34,253,74,201,58,244,74,205
40 DATA47,6,42,251,74,34,245,74,42,253,74,34,247,74,33,68,74,34,220,74
50 DATA205,119,74,201,42,249,74,34,245,74,106,99,34,247,74,33,223,4,34
60 DATA220,74,205,119,74,42,251,74,34,245,74,42,253,74,34,247,74,33,68
70 DATA74,34,220,74,201,58,244,74,205,47,6,33,223,4,34,220,74,175,87,95
80 DATA111,58,245,74,71,58,247,74,144,242,138,74,47,60,48,1,71,58,246
90 DATA74,79,58,248,74,145,242,158,74,47,60,103,125,246,2,111,124,79,144
100 DATA250,170,74,120,65,79,125,246,4,111,120,31,230,127,47,60,103,229
110 DATA213,197,58,245,74,71,58,246,74,79,125,230,4,202,197,74,122,83,95
120 DATA125,230,2,123,202,206,74,47,60,129,95,125,230,1,122,202,217,74,47
130 DATA60,128,87,205,223,4,193,209,225,20,120,146,248,124,129,103,202
140 DATA177,74,250,177,74,144,103,28,195,177,74,19199,13,3,205,40,74,205
150 DATA19,74,205,40,74,195,9,74,24928,4,0,1,2,3,25127,220,49,141,76,205
160 DATA250,102,195,12,102,254,66,202,86,98,254,70,202,132,98,254,76,202
170 DATA206,98,254,80,202,249,98,254,84,202,217,100,254,150,202,53,104
180 DATA254,150,202,53,104,195,53,104,35,205,191,117,50,248,74,205,146
190 DATA119,50,247,74,205,146,119,50,245,74,205,146,119,50,244,74,205,146
200 DATA119,50,246,74,229,1,244,74,205,162,5,225,58,255,74,205,47,6,201
210 DATA35,205,191,117,87,205,146,119,95,205,146,119,79,197,205,146,119
220 DATA193,71,229,205,174,98,21,205,192,98,20,28,197,213,193,205,16,6
230 DATA193,184,194,152,98,225,201,205,0,6,20,197,213,193,205,16,6,197
240 DATA209,193,184,194,174,98,201,62,21,50,177,98,205,174,98,62,20,50
250 DATA177,98,201,35,205,191,117,50,245,74,205,146,119,50,246,74,205,146
260 DATA119,50,247,74,205,146,119,50,248,74,205,146,119,50,244,74,229,205
270 DATA107,74,225,58,255,74,205,47,6,201,35,205,191,117,50,255,74,195
280 DATA242,98,25433,3,195,77,68,25704,2,48,98,25817,50,35,205,191,117,50
290 DATA249,74,205,146,119,50,250,74,205,146,119,50,251,74,205,146,119,50
300 DATA252,74,205,146,119,50,253,74,205,146,119,50,254,74,205,146,119
310 DATA50,244,74,229,205,0,75,195,241,98,0,0
500 RESTORE
510 READS,C
520 IFS=0ANDC=0THEN END
530 FORA=STOS+C-1
540 READV:POKEA,V
550 NEXT
560 GOTO510
```

W.J. Moore
571 So. Broadway
Pittsburg, Ca. 94565

Mr. George A. Leggett
20562 Woodward
Mt. Clemens, Mi. 48043

Dear George,

Your recent contribution of NEW BASIC COMMANDS OVERLAY is certainly of commercial quality. For those that did not have enough stick-with-it-ness with the INTERACT, I bet they would kick themselves if they knew of this terrific work. And all that is required is to subscribe! Just fantastic George.

I would like to resubmit YOUR PROGRAM again in a different format. The reason for changing format is to speed up putting code in memory. This version does not have to make conversions and only requires seven (7) seconds to run. Hardly worthwhile to prepare a special tape with a monitor.

Sincerely,

W.J. Moore

MACHINE SHOP TALK

CANNON FIRE GAME

By George A. Leggett

This is going to be a very different and unique MACHINE SHOP TALK. I am finally acting upon a big request from the many letters and calls I have received from all of you out there which is a complete working 8080 Machine program. Since I began this series of articles I have shown you routines but the question has been let's see how you put it all together and make something out of it. That's exactly what we're going to do without further ado.

About The Game: Why a game? I chose to program a game because many things happen very fast in a game and it uses a lot of graphics unlike a business program or a data base. Besides being too long and complex I have been doing much of that type of programming in BASIC so it was a refreshing change to work in 8080 again. I think that the game is a fairly good one. I'm no super-duper game writer. I don't get in to the playing of games as much as the programming of them. Action games are definitely not one of my stronger points being that I have an extremely hard time winning at them you can see the problem of testing the program. With this game CANNON FIRE the major goal is not to create a new PACMAN or some other award winning video game but to create a tutoring program as you requested that is also hopefully enjoyable to play.

I shall try to break the program down and explain each part of it and how it works which could be lengthy. I used the least number of bytes possible because we would not want 82 pages on one program which is another reason for a simple game.

How the game is played is simple. The screen can be described in four parts. The top part of the screen is the scoreboard. The part below that is a ship coming from the right to the left. Under that is a ship coming left to right. At the bottom of the screen is your cannon. That goes left and right. The bottom ship may fire upon your cannon and you may fire upon both ships. There is a twist to this game. In other video games when when the enemy or alien or what have you hits you, usually after three hits the game is over. That bothered me for five years So I changed that rule right away doing something I think is neat. First of all you do not begin with zero points. You begin with 100 points. Every time you fire your cannon 2 points are diducted. Every time you hit the top ship you get 20 points, the bottom ship 10 points. Now, if the bottom ship hits you, you lose 100 points. Now you can see whereby you have 0 or less points the game is over. We'll talk more about that later. We are using the left Joy stick input. The POT turns your cannon left and right, the FIRE Button shoots the pellets.

You can write any scenario you wish. I guess scenarios are what sells games. As I explain all the "ins" and "outs" it will sound humdrum. However, I could begin by saying you are in the 22nd century. You are defending a major city. Enemy planes are flying overhead dropping nuclear implosion bombs. Your energy reserves lower with each hit. Your deflector shields are up. How long can you withstand the attack? Will you have enough power to ward off your enemy indefinitely? The fate of the world and the future is in your hands. Well, this is marketing. This is how you can make any piece of garbage sound great. We're not doing that here. It's a good teaching game.

I spent more time programming it in a way I could explain it to you than programming it the way I normally would. There is a differemce as we shall now see. All numbers are in Hexadecimal. When we say a block we mean 256 Bytes. For example, Block 5800 means every byte between 5800 and 58FF thus, the 5800 Block.

The game lies between 5800 and 5F7F. Every block has a specific job to do. This is for clarification and this is mainly how I program anything. I assign a block a job and leave it be. Let us begin by naming each block and the job it does and then break each part down in more detail.

Block 5800 is what I call the main loop. It initializes the game, resets the scoreboard and then begins play by a series of calls which check for cannon firing and hits etc. The 5800 block is only used from 5800 to 583B. It's a very short block but it controls everything. You'll find many of these blocks are not full. Why not put it all together? Yes, if you were selling this you would condense it down to only 1 or 2 or 3 blocks. But for simplicity 2e will not do that.

5900 Block is not being used.

5A00 Block is called Blow Up Ship Effect. 5A00 to 5A26.

5B00 Block is movement of both ships.

5C00 Block Movement of your cannon.

5D00 Block controls firing of the cannon, firing of the missile coming at the cannon, the hit target for the cannon firing upon the ship and the hit cannon for the ship firing upon the cannon. 5D00 to 5DBB.

5E00 is the Scoreboard Block.

5F00 Data Block. The first part is the character set for the ships and cannon and different memory locations for storing X and Y Axes (axises?) for ships firing and all data we need to call during the game. Thus a temporary storage area.

The stack is at 4C00 and moves of course backwards into 4B00 block. I put my stack here now mainly because this way if you blow up your stack it will only bump into the screen and nothing is lost. If you put it higher and you wipe out something you'll blow up your program. A little goody I've learned from doing a lot of BASIC modification. I shall now discuss each block in more detail.

Main Loop 5800 to 581A loads the stack, clears the screen. Load B and C with the color table. Call the Color Routine. Load H and L with the Character Table. Store it where ROM 1 will look to find your character set. CALL 5E26 Clear the scoreboard to 100 points. CALL 5C17 which prints your cannon on the screen.

581B to 583B This is the main game loop which it will pass through continually as the game is played. Routines are called and must return there. It's a sequential loop meaning it does one thing and comes back and does another. This is how everything is moving or appears to be moving at one time. If you don't return to this loop quick enough things will cease to move around. There are exceptions which we will come to l ater. We now have a series of calls to carry out the sequence.

CALL 5C0E This checks to see if the POT has been moved. If the POT is in the same position it returns. If the POT has been moved it then prints the new location of the cannon.

CALL 5B1A Move lower ship left to right. Print the lower ship, moving it one position on the screen, erasing the old ship's position and return.

CALL 5B39 Move upper ship right to left. This moves one notch right erasing the whole ship in the same manner as the routine for the lower ship and return.

CALL 5D00 Fire Cannon Routine. If the Fire Button is not pressed it will simply return to the main loop. If it is pressed it will fire the cannon.

CALL 5D5A Missile Fire where the ship fires upon you. If the time is right which is predetermined by the clock it will fire. If it's not the right time it will return to the loop. At this time we load A 5F7B This loads A with the position of the cannon fire. It compares it for 30 Hex and if it's less than that Jump if minus back to the start of the loop. If it's greater than that, it calls the scoreboard.update routine. When you fire your cannon, that pellet keeps going up the screen. and when it hits the top of the cyan line which notifies to update the scoreboard and to deduct your 2 points. The 2 points have previously been deducted when you hit the Fir Button but this will update the figure on the scoreboard. At this time, Subtract A from A. Why? Make A 0 again. Many people use EXCLUSIVE OR but A from A is 0 in my book. Store this in 5F7B to make sure when it scans here again it reads 0. Jump to 581B and do the whole thing all over again. This concludes the main loop.

I shall now break down each of these calls farther and go through each of them as they appear in the main loop.

Starting at 5815 the call is to 5E26 which is Scoreboard reset. Move into B 6 which means to do this 6 times since the scoreboard has 6 digits. Load H and L with the memory location of where to find our numbers. My scoreboard routines use direct decimal ASCII. Everything is stored in ASCII. All numbers are preceeded by a 3 meaning a group of 3 16's for example, a 0 is represented by 30 and a 5 by 35. This is easier than converting binary to decimal and vice versa. We think in decimal. Let it do it too. Move into A 30 or, a 0 then move A into M or the memory location. Increase H, decrease B Jump if not 0 meaning do this 6 times. Put 6 zeros 5F70 to 5F75. Now move into A 31 which is the digit 1. Store this at 5F72 which is the hundreds place. The 1's place is at 5F70, the 10's place at 5F71 and so on to 5F75 which is the 100,000's place. This is your reset scoreboard routine, which you could reset to whatever score you like. A 35 would reset to 500.

Next in the main loop at 5818 is a call to 5C17. This is part of the move cannon routine. This will be explained when the whole routine is called. I'm using a part of the routine to display the cannon.

Next in the main loop at 581B is a call to 5C0E which is now the whole move cannon routine. Load A with 5FF8. This is the left POT Digital to Analog readout. Move that into B because we want to keep that info. Load A with 5F78 which is the last position your cannon was located. Compare B with A. Are they equal? The compare always subtracts one from another. If 0 the cannon was never fired. Acutally, to be more precise, the cannon was never moved. Now if it was moved we now get back to 5C17. CALL 5C00. Here we have a call within a call. Don't worry. Later there are five calls within a call! You better get used to it. 5C00 is to clear the cannon. Erase it. Okay, we could print Color 0 over it but this is slow. Here is what I did from 5C00 to 5C0D. Load H and L with 4800 which is the bottom of the screen. Move into A 0 then move that into memory. Increase H and L by 1 and now move H to A. We always do our logic with A. Compare Immediately 4A. This means we are asking if H and L at 4A00 or at the end of screen memory? Jump if not 0 meaning the answer is no, H and L are only at 4801. It will keep doing this very very fast and fill up the next 512 bytes with zeros thus, where your cannon was it won't be there any more. C9 return. Return where? Back to 5C1A. Move Register B into D. Move into E 40. This sets your X and Y coordinates for your cannon. Move intoAa 1 in this case Color 1 Call the print color routine. Move into C 1 in this case it is character 1 or your cannon. CALL 043E which prints character 1 at the X and Y positions given by D and E. Move D into A now. This is your X axis which you now store at 5F78 the X axis of the cannon and then Return. And that is how the cannon moves. We now go back to the main loop.

The next address in our main loop is 581E CALL 5B1A which is the routine to move the bottom ship left to right. Move into A Color 2 and Call the Print routine. CALL 5B00.

5B00 **clears** the ship in the identical way we clear the cannon. We are erasing the screen memory from 4400 to 4500 Hex. We compare the X axis of that ship in 5F79 for a 60 which is 96 decimal. How far is it from the end of the screen? Jump if minus over 1 byte to 5B2B. Why? 5B2A is subtract A from A which would make it 0 and that means if that ship is too far to the right of the screen make A 0 then add 5 to A. If it's not too far right don't 0 Register A just add 5 more to it.

The number at 5B2C is important. This controls the speed of how fast that ship moves. It is set in increments of 5 pixels. I have had it as fast as increments of 15 pixels. You can use increments of however many pixels you wish. The higher the number the faster the ship moves across the screen. We then store that number at 5F79 since we must remember that number for the next time around. We move into E a 20 which is our Y axis and is a constant. It never varies. Move A into D which is our X axis which we have just updated. Move into C 03 which is character 3 or the ship. Call the character routine at 043E and RETURN. Back we go to the main loop.

At 5820 we CALL 5B39 to move ship right to left. It's identical to the left to right routine we just discussed but it's all backwards. You call 5B0D clear right ship routine which is the same as clear left ship routine only different addresses. Instead of comparing for the right side of the screen we are comparing for the left side of the screen. Here we jump over 2 bytes. Why? We can't put A at 0, we put A at 64 Hex or 100 decimal. Otherwise, everything is the same.

Back to our main loop at 5824. CALL 5D00. Fire Cannon. Did you fire the cannon? Load A with 5FF7 This is the Fire button. Compare for 80. Why 80? Because the button is not the ideal 0 to FF. It varies and 80 is the middle ground. Therefore it has got to be some number between 0 and 7F or between 81 and FF for On and for Off. Return if positive meaning you never pressed the Fire button. If you did press the Fire button, CALL 5E13 then CALL 5E13. Yes, I said it twice twice! This call is to the scoreboard to decrease it by 1 point for each call. We will come back to that routine later.

Store A at 5F7B

At 5D0F we load A with 5F78. the cannon X position. Add 3 to this so it fires its bullet from the middle of the cannon, not the edge. Move into E a 3E which is the Y position for the tip of the cannon. With the X and Y positions of the cannon in D and E, move into C a 3 for the color you want your bullet to fire in. CALL 0600 the Plot Routine for a pixel. PUSH D your X and Y positions, CALL 5B1A which is again your move ship left to right and CALL 5B39 move ship right to left. We muxt update them or they won't move. POP D to restore the X and Y data. What you've done is to buy time for yourself.

Move into C 0 and call 0600 the Plot routine to erase the dot you plotted. Thus a dot appears, your ship moves and the dot goes away. Move E into A which is your Y axis. Subtract 3. Remember, we're moving it up the screen and in Machine Code when we move up the screen the numbers decrease example, Coordinate 0, 0 is at the upper left. Move it back to E. PUSH D and POP B Why this command? I could have said Move D to B and Move E to C because I wanted the information in D and E also to move to B and C but it happens that I remember the H ex codes for Pushes and Pops easier than that of the Moves so I did it this way. Call the Get Pixel routine. This routine takes the X and Y location now in B and C returns the color of that pixel. Now it compares it for Color 1 which is the color of cannon and top line of scoreboard. This means you have hit nothing. Return if 0. Now compare for Color 2 which is the color of the ship. Jump if not 0 meaning nothing is there but a blank to 5D17 a7d do the whole thing over again. Move the little pellet up 1, move the ship left and right and check all that information.

If, however, it is Color 2 then we go to the Hit Target routine which begins at 5D3A. Move E into A Subtract 5 and move A back into E. This is so you get the explosion over the ship. Compare A now for 12 H ex. This is to see if it's the lower ship or the higher ship. Why? If it's the higher ship you get 20 points; if it's the lower ship you get 10 points. If it is the high ship, CALL If 0 5E84 which is the Times 10 Scoreboard routine. If it's not it will skip over this.

Now at 5D43 Move into A 3 Call the Print Routine. Move into C 4 Call the Character routine using the fourth character, the explosion character. Call 5E84 which is the increase Times 10 Scoreboard routine again. CALL 5E3D Update Scoreboard Routine.

CALL 5A00 for the Blow Up Ship Effects Routine. Here is how this routine works: Move into A OA or 10 Decimal. PUSH it. We're going to do this 10 times. Load B and C with 5F6F and Call the Color Routine. This turns off Color 3 to a background color. Load B and C with OA (10 Decimal) Load D and E with 20 (32 Decimal). CALL 07BF Tone Routine. Make a beep.

Load B and C now with 5E60 our first color table to turn back on Color 3 and Call the Color Routine. Now Load B and C with 5 Decimal, Load D and E with 48 Decimal Call the Tone Routine 07BF POP PSW A and Flags. Decrease A by 1 Jump If not 0 to 5A02 meaning do all that over again 10 times, when it is 0 C9 Return. That is your ship effects routine. We return back to 5D56.

5D56 C9 RETURN. From the Hit Target routine we will now be returning to our main loop. Remember that we can return by hitting a cyan border or by hitting a ship. We return to 5827 in our main loop which says CALL 5D5A Missile Fire. This is it firing upon you. Load A with 5FEF This is the 1/60 clock. Compare for 10, Return if Positive. Meaning 246 sixtieths of a second it will do nothing. You can vary this time to whatever interval you would like. They will come at you faster or slower. If you return, you are back in the main loop. If its number is up, we Move into A 3 Call Print routine for Color 3. Move into A 28 or the Y Coordinate of where the missile will begin to drop. Store that at 7F7C. Missile Y Position.

Now move it from A into E. Load A with 5E79. This is the X position of the left to right ship. Move that into D so that D and E now have an X and a Y position of where that missile will be coming from. Move into C 5 or Character 5 the missile. CALL 3E04 which is the character Output routine Move into A 0 Call the Print Routine for Color 0 the blank. Call the character. This means we have printed the missile and then erased the missile. It blinks. Increase E the X axis. PUSH D POP B merely moves it into B and C as discussed earlier and CALL 0610 to return the pixel color of X and Y positions. Compare it for 1 or Color 1. That's us, the cannon in Color 1. Jump if 0 to the Hit Cannon Routine at 5D9D. If it's not 0 meaning it didn't hit our cannon yet PUSH D CALL 5B1A and CALL 5B39. Our old friends Move The Ships! CALL 5C0E Move The Cannon. If you moved it. POP D Get that value back. Increase E. Increase D. Increase E. Increase D. Four times. Plus the one we mentioned before--five times. We want that missile to fall very fast at us. (Well, we don't want it, but it will anyway) Move that into E for our new Y position. Compare that for 48 meaning Did the missile hit the bottom of the screen yet? Jump if minus to 5D67

and do it all over again. If not, C9 RETURN.

If the missile hit your cannon detected by Color 1 Jump to 5D9D Hit Cannon Routine. Move into A 3. Call the Print Routine. Move into C 4 or the character 4. Call the character routine.
CALL 5E90. This is the Decrease Scoreboard by 100 points routines. CALL 5E3D to print the new score. Move into A 0A meaning we're going to do the following thing 10 times. PUSH A to remember that number. CALL 5A00 for the Blow up ship effects routine. We're using it again. Only this time it is you being blown up. One can change that and I will discuss endings later. POP A and the flags, Decrease A by 1.Jump if not 0 meaning did you do this routine 10 times? Jump to 5DAF no. If yes, CALL 5C17 to print your cannon. You just blew it up so you must replace it. That is the Hit Cannon routine. We return to 582A where we continue our loop as mentioned earlier.

This leaves us with two last things, the character set and the scoreboard. For further information about how characters are made, refer to 8080 FOR EVERYONE or to INTERACTIONS Vol. 1 no. 4 Page 2. All characters are 8 by 8 pixels. At 5F00 and 5F01 are 8 and 8 meaning 8 X by 8 Y pixels. Following every 8 bytes represent a character. You can change or alter them however you wish. I'm not a very good artist when it comes to these things. I leave it in your hands.

The scoreboard is made up of two major parts. First, the printing of the scores with full zero suppression which is not usually done in video games. Secondly, the increment and decrement of the scores. As I mentioned earlier, the scoreboard has two tables: an ASCII table going from 1's, 10's, 100's, etc. and a Print table which does the reverse; 10,000's, 1000's, 100's, 10's, 1's. This is after all the way the score is printed. But it's easier to compute the scores using 1's first and so on then later to use the reverse order to print.

The routine for adding up scores begins at 5E00. Load H and L with 5F70. This is the address of the beginning table or the 1's place. This routine increases scoreboard by 1. Move it to A from M. Increase A by 1 and compare it for 3A. Why? Remember we're using ASCIIA 39 is a 9 in ASCII and a 3A is a carry in this case. Jump if 0 to 5E0C meaning yes, we have a carry. If we have no carry and it's 9 or under, move it back to memory A to M and RETURN. That's all. If it is greater than 9 and we have a carry, move into A a 30 or an ASCII 0. Move that into memory now increase H and L by 1 and jump back to 5E03 to bring up the next digit and do it all over again. Increment each place as you go along.

To decrease the scoreboard by 1 the routine is at 5E13 Load H and L with the same 5F70 Move it into A, we decrease A now, and this time we compare it for 2F or less than 30 or ASCII 0. Jump if 0. If it's not, move it back to memory from A and return. Thus it was a 2 minus a 1 is 31 so move it and leave it. If it was 0, move into A after you Jump to 5E1F a 39 which is 9 and move that into memory Increase H and L by 1. Jump to 5E16 and do it all over again for the different places. This is increasing and decreasing by 1. How about increasing and decreasing by more than 1? At 5E78 Decrease by 10. All this does is load A with 10 PUSH and Call the decrease routine 10 times. It can only decrease 1 at a time. This is not the most efficient way to do this but it is the easiest. If you were doing it more efficiently you could load H and L with the placement of 10's and then Call 5E16. Either way will work. I'm giving both ways so you may see it. At 5E84 is the increase by 10 routine. Which is nothing more than 10 loops of the increase by 1 routine. At 5E90 is the Decrease by 100 routine. This is a FOR loop of calling the decrease by 10 calls the decrease by 1 routine! 5E9C increase by 100 which calls increase by 10's which calls the increase by 1's. Again, these are put here to demonstrate loops in a loop. They are not efficient and they are rather slow. But, they work in this example. If you were going to rewrite it I recommend loading H and L simply with the place you want to increment like 10's are at 5F71 and then Call the Increase Decrease routines at 3 bytes up and it will work great too.

That leaves us with 5E3D Scoreboard Display routine. Load B and C with 5F55 and Call 05A2 Box routine in Rom 1. At 5F55 is the size and coordinates of our box. Refer to your ROM listings to see how this or any ROM routines mentioned in this program work. Any calls of 07FF or less refer to ROM routines. Load B and C with the Call to another box which draws that line at the bottom of the scoreboard. Move into A Color 0 Call the print routine. Yes, we're printing in Color 0. We just made a box so we're going to print the background color in the box. It adds a nice touch. Load H and L with 5F40 or the print storage address of the scoreboard. Load D and E with 5F75 which is the 100,000's place of the ASCII scores we discussed earlier. We must now reverse to print the right way. Move into B 6 Do it 6 times. 6 places. Load A with D and E. D and E contain the address 5F75 Move A to M, now 5F40, Increase H and L, Decrease D and E, Decrease B, and Jump if not 0 to 5E56. Increase the screen storage by 1 place while decreasing the ASCII storage by 1 place reversing places whereby 1's are now in the 6th place. Thus, scores are printed correctly. If that sounds confusing it may be but that's how it works!

Load B and C with 5F40 and D with 27 and E with 06 for X and Y coordinates and B and C contain the address of the screen scoreboard. At 5E64, load A using B and C as an address. Compare it with 39 meaning: is there 900,000? The scoreboard is designed to go only to 99,999 because I'm using the last place to detect if you went below 0 thus it will count to nine hundred thousand nine hundred whatever. Jump to the Lose Routine if 0 We'll get back to this.

At 5E6A now load A again using B and C. I know we did that but we're going to do it again. This time we compare it for 30 or a 0. This is your zero suppression. Jump if not 0 to 054F Now how can we jump to a ROM routine and not return because all ROM Routines have returns. That's great! We're going to return to our main loop or wherever we came from when we called this routine in the first place. Let ROM 1 return us. Why should we take more bytes Yes you can have a direct jump to a ROM routine and it will bring you right back to whatever is on your stack. So make sure you don't push anything on your stack when you jump to a ROM routine or when you return it's going to return to the last thing you pushed. So make sure a Call is the last thing on your stack. Boy, that's confusing even to me!

If it is 0, increase B and C by 1 B and C are the address. Move D into A and Add 6. Move A back to D. What did we do here? We took the X coordinate of our print routine and we added 6 or moved it over a place. That way the 1's place will always be the farthest right thing on the scoreboard. Jump to 5E6A and do all that over again.

Now at 5E67 I said Jump to a lose routine. Well, I mentioned no numbers and on your printout sheets of the listing it says 0 0. What do you mean Jump to 0 0? You're going to fill in the blank. You are going to make me a lose routine. Right now it will jump to Depress L R to Restart. So when you have a score of 0 or below, you will have to restart it. Anyone out there ready for a challenge? I know there are some great people out there so I want to see some fancy lose routines and we will publish them in a later issue.

This concludes the discussion of the game. If you have questions, please write or call and I'll be happy to answer any questions. I took approximately 10 hours to complete writing it so if there is a bug or something please excuse me. I don't think there is. I surprised myself in how quickly I got through it. I guess I haven't lost my touch in Machine Code. Many things can be done to change this basic outline such as: changing colors, characters, different blowup routines for your cannon and ships, putting in tones, for the Fire button, putting in two or three missiles. Let yourself go. All I did was give you a shell, a basic concept. The goal was to teach you how Machine Language is put together as a whole. You are the pilot, you have the control now. In this case I can say shape up or ship out or you'll be blown off the screen.

I have been very hesitant over the year to do this project as it does take a lot of space and I hope those of you not interested are not bothered by it. Last month we got all you FORTH nuts. This time, Machine Code nuts. Hopefully, we can get to everyone's fancy in time. Thank you for another great Machine Shop Talk.

Sincerely,

George A. Leggett

```
5800 31 LXI SP
5801 00  *
5802 4C  *
5803 CD CALL
5804 73  *
5805 05  *
5806 01 LXI B
5807 60  *
5808 5F  *
5809 CD CALL
580A 36  *
580B 06  *
580C 21 LXI H
580D 00  *
580E 5F  *
580F 22 SHLD
5810 E1  *
5811 5F  *
5812 CD CALL
5813 26  *
5814 5E  *
5815 CD CALL
5816 3D  *
5817 5E  *
5818 CD CALL
5819 17  *
581A 5C  *
581B CD CALL
581C 0E  *
581D 5C  *
581E CD CALL
581F 1A  *
5820 5B  *
5821 CD CALL
5822 39  *
5823 5B  *
5824 CD CALL
5825 00  *
5826 5D  *
5827 CD CALL
5828 5A  *
5829 5D  *
582A 3A LDA
582B 7B  *
582C 5F  *
582D FE CPI
582E 30  *
582F FA JM
5830 1B  *
5831 58  *

5832 CD CALL
5833 3D  *
5834 5E  *
5835 97 SUB A
5836 32 STA
5837 7B  *
5838 5F  *
5839 C3 JMP
583A 1B  *
583B 58  *



5A00 3E MVI A
5A01 0A  *
5A02 F5 PSH PSW
5A03 01 LXI B
5A04 65  *
5A05 5F  *
5A06 CD CALL
5A07 36  *
5A08 06  *
5A09 01 LXI B
5A0A 0A  *
5A0B 00  *
5A0C 11 LXI D
5A0D 20  *
5A0E 00  *
5A0F CD CALL
5A10 BF  *
5A11 07  *
5A12 01 LXI B
5A13 60  *
5A14 5F  *
5A15 CD CALL
5A16 36  *
5A17 06  *
5A18 01 LXI B
5A19 05  *
5A1A 00  *
5A1B 11 LXI D
5A1C 30  *
5A1D 00  *
5A1E CD CALL
5A1F BF  *
5A20 07  *
5A21 F1 POP PSW
5A22 3D DCR A
5A23 C2 JNZ
5A24 02  *
5A25 5A  *
5A26 C9 RET

5B00 21 LXI H
5B01 00  *
5B02 44  *
5B03 97 SUB A
5B04 77 MOV M,A
5B05 23 INX H
5B06 7C MOV A,H
5B07 FE CPI
5B08 45  *
5B09 C2 JNZ
5B0A 03  *
5B0B 5B  *
5B0C C9 RET
5B0D 21 LXI H
5B0E 00  *
5B0F 42  *
5B10 97 SUB A
5B11 77 MOV M,A
5B12 23 INX H
5B13 7C MOV A,H
5B14 FE CPI
5B15 44  *
5B16 C2 JNZ
5B17 10  *
5B18 5B  *
5B19 C9 RET
5B1A 3E MVI A
5B1B 02  *
5B1C CD CALL
5B1D 2F  *
5B1E 06  *
5B1F CD CALL
5B20 00  *
5B21 5B  *
5B22 3A LDA
5B23 79  *
5B24 5F  *
5B25 FE CPI
5B26 60  *
5B27 FA JM
5B28 2B  *
5B29 5B  *
5B2A 97 SUB A
5B2B C6 ADI
5B2C 0F  *
5B2D 32 STA
5B2E 79  *
5B2F 5F  *

5B30 1E MVI E
5B31 20  *
5B32 57 MOV D,A
5B33 0E MVI C
5B34 03  *
5B35 CD CALL
5B36 3E  *
5B37 04  *
5B38 C9 RET
5B39 3E MVI A
5B3A 02  *
5B3B CD CALL
5B3C 2F  *
5B3D 06  *
5B3E CD CALL
5B3F 0D  *
5B40 5B  *
5B41 3A LDA
5B42 7A  *
5B43 5F  *
5B44 FE CPI
5B45 05  *
5B46 F2 JP
5B47 4B  *
5B48 5B  *
5B49 3E MVI A
5B4A 64  *
5B4B D6 SUI
5B4C 0A  *
5B4D 32 STA
5B4E 7A  *
5B4F 5F  *
5B50 1E MVI E
5B51 12  *
5B52 0E MVI C
5B53 02  *
5B54 57 MOV D,A
5B55 CD CALL
5B56 3E  *
5B57 04  *
5B58 C9 RET
```

```
5C00 21 LXI H
5C01 00  *
5C02 48  *
5C03 3E MVI A
5C04 00  *
5C05 77 MOV M,A
5C06 23 INX H
5C07 7C MOV A,H
5C08 FE CPI
5C09 4A  *
5C0A C2 JNZ
5C0B 03  *
5C0C 5C  *
5C0D C9 RET
5C0E 3A LDA
5C0F F8  *
5C10 5F  *
5C11 47 MOV B,A
5C12 3A LDA
5C13 73  *
5C14 5F  *
5C15 B8 CMP B
5C16 C8 RZ
5C17 CD CALL
5C18 00  *
5C19 5C  *
5C1A 50 MOV D,B
5C1B 1E MVI E
5C1C 40  *
5C1D 3E MVI A
5C1E 01  *
5C1F CD CALL
5C20 2F  *
5C21 06  *
5C22 0E MVI C
5C23 01  *
5C24 CD CALL
5C25 3E  *
5C26 04  *
5C27 7A MOV A,D
5C28 32 STA
5C29 73  *
5C2A 5F  *
5C2B C9 RET

5D00 3A LDA
5D01 F7  *
5D02 5F  *
5D03 FE CPI
5D04 30  *
5D05 F0 RP
5D06 CD CALL
5D07 13  *
5D08 5E  *
5D09 CD CALL
5D0A 13  *
5D0B 5E  *
5D0C 32 STA
5D0D 7B  *
5D0E 5F  *
5D0F 3A LDA
5D10 78  *
5D11 5F  *
5D12 C6 ADI
5D13 03  *
5D14 57 MOV D,A
5D15 1E MVI E
5D16 3E  *
5D17 0E MVI C
5D18 03  *
5D19 CD CALL
5D1A 00  *
5D1B 06  *
5D1C D5 PSH D
5D1D CD CALL
5D1E 1A  *
5D1F 5B  *
5D20 CD CALL
5D21 39  *
5D22 5B  *
5D23 D1 POP D
5D24 0E MVI C
5D25 00  *
5D26 CD CALL
5D27 00  *
5D28 06  *
5D29 7B MOV A,E
5D2A D6 SUI
5D2B 03  *
5D2C 5F MOV E,A
5D2D D5 PSH D
5D2E C1 POP B

5D2F CD CALL
5D30 10  *
5D31 06  *
5D32 FE CPI
5D33 01  *
5D34 C3 RZ
5D35 FE CPI
5D36 02  *
5D37 C2 JNZ
5D38 17  *
5D39 5D  *
5D3A 7B MOV A,E
5D3B D6 SUI
5D3C 05  *
5D3D 5F MOV E,A
5D3E FE CPI
5D3F 12  *
5D40 CC CZ
5D41 34  *
5D42 5E  *
5D43 3E MVI A
5D44 03  *
5D45 CD CALL
5D46 2F  *
5D47 06  *
5D48 0E MVI C
5D49 04  *
5D4A CD CALL
5D4B 3E  *
5D4C 04  *
5D4D CD CALL
5D4E 34  *
5D4F 5E  *
5D50 CD CALL
5D51 3D  *
5D52 5E  *
5D53 CD CALL
5D54 00  *
5D55 5A  *
5D56 CD CALL
5D57 00  *
5D58 5B  *
5D59 C9 RET
5D5A 3A LDA
5D5B EF  *
5D5C 5F  *
5D5D FE CPI
5D5E 10  *
5D5F F0 RP

5D60 3E MVI A
5D61 03  *
5D62 CD CALL
5D63 2F  *
5D64 06  *
5D65 3E MVI A
5D66 23  *
5D67 32 STA
5D68 7C  *
5D69 5F  *
5D6A 5F MOV E,A
5D6B 3A LDA
5D6C 79  *
5D6D 5F  *
5D6E 57 MOV D,A
5D6F 0E MVI C
5D70 05  *
5D71 CD CALL
5D72 3E  *
5D73 04  *
5D74 3E MVI A
5D75 00  *
5D76 CD CALL
5D77 2F  *
5D78 06  *
5D79 CD CALL
5D7A 3E  *
5D7B 04  *
5D7C 1C INR E
5D7D D5 PSH D
5D7E C1 POP B
5D7F CD CALL
5D80 10  *
5D81 06  *
5D82 FE CPI
5D83 01  *
5D84 CA JZ
5D85 9D  *
5D86 5D  *
5D87 D5 PSH D
5D88 CD CALL
5D89 1A  *
5D8A 5B  *
5D8B CD CALL
5D8C 39  *
5D8D 5B  *
```

```
5D8E CD CALL
5D8F 0E *
5D90 5C *
5D91 D1 POP D
5D92 1C INR E
5D93 1C INR E
5D94 1C INR E
5D95 1C INR E
5D96 7B MOV A,E
5D97 FE CPI
5D98 42 *
5D99 FA JM
5D9A 67 *
5D9B 5D *
5D9C C9 RET
5D9D 3E MVI A
5D9E 03 *
5D9F CD CALL
5DA0 2F *
5DA1 06 *
5DA2 0E MVI C
5DA3 04 *
5DA4 CD CALL
5DA5 3E *
5DA6 04 *
5DA7 CD CALL
5DA8 90 *
5DA9 5E *
5DAA CD CALL
5DAB 3D *
5DAC 5E *
5DAD 3E MVI A
5DAE 0A *
5DAF F5 PSH PSW
5DB0 CD CALL
5DB1 00 *
5DB2 5A *
5DB3 F1 POP PSW
5DB4 3D DCR A
5DB5 C2 JNZ
5DB6 AF *
5DB7 5D *
5DB8 CD CALL
5DB9 17 *
5DBA 5C *
5DBB C9 RET

5E00 21 LXI H
5E01 70 *
5E02 5F *
5E03 7E MOV A,M
5E04 3C INR A
5E05 FE CPI
5E06 3A *
5E07 CA JZ
5E08 0C *
5E09 5E *
5E0A 77 MOV M,A
5E0B C9 RET
5E0C 3E MVI A
5E0D 30 *
5E0E 77 MOV M,A
5E0F 23 INX H
5E10 C3 JMP
5E11 03 *
5E12 5E *
5E13 21 LXI H
5E14 70 *
5E15 5F *
5E16 7E MOV A,M
5E17 3D DCR A
5E18 FE CPI
5E19 2F *
5E1A CA JZ
5E1B 1F *
5E1C 5E *
5E1D 77 MOV M,A
5E1E C9 RET
5E1F 3E MVI A
5E20 39 *
5E21 77 MOV M,A
5E22 23 INX H
5E23 C3 JMP
5E24 16 *
5E25 5E *
5E26 06 MVI B
5E27 06 *
5E28 21 LXI H
5E29 70 *
5E2A 5F *
5E2B 3E MVI A
5E2C 30 *
5E2D 77 MOV M,A
5E2E 23 INX H
5E2F 05 DCR B

5E30 C2 JNZ
5E31 2D *
5E32 5E *
5E33 3E MVI A
5E34 31 *
5E35 32 STA
5E36 72 *
5E37 5F *
5E38 97 SUB A
5E39 32 STA
5E3A 76 *
5E3B 5F *
5E3C C9 RET
5E3D 01 LXI B
5E3E 55 *
5E3F 5F *
5E40 CD CALL
5E41 A2 *
5E42 05 *
5E43 01 LXI B
5E44 50 *
5E45 5F *
5E46 CD CALL
5E47 A2 *
5E48 05 *
5E49 3E MVI A
5E4A 00 *
5E4B CD CALL
5E4C 2F *
5E4D 06 *
5E4E 21 LXI H
5E4F 40 *
5E50 5F *
5E51 11 LXI D
5E52 75 *
5E53 5F *
5E54 06 MVI B
5E55 06 *
5E56 1A LDAX D
5E57 77 MOV M,A
5E58 23 INX H
5E59 1B DCX D
5E5A 05 DCR B
5E5B C2 JNZ
5E5C 56 *
5E5D 5E *

5E5E 01 LXI B
5E5F 40 *
5E60 5F *
5E61 11 LXI D
5E62 06 *
5E63 27 *
5E64 0A LDAX B
5E65 FE CPI
5E66 39 *
5E67 CA JZ
5E68 00 *
5E69 00 *
5E6A 0A LDAX B
5E6B FE CPI
5E6C 30 *
5E6D C2 JNZ
5E6E 4F *
5E6F 05 *
5E70 03 INX B
5E71 7A MOV A,D
5E72 C6 ADI
5E73 06 *
5E74 57 MOV D,A
5E75 C3 JMP
5E76 6A *
5E77 5E *
5E78 3E MVI A
5E79 0A *
5E7A F5 PSH PSW
5E7B CD CALL
5E7C 13 *
5E7D 5E *
5E7E F1 POP PSW
5E7F 3D DCR A
5E80 C2 JNZ
5E81 7A *
5E82 5E *
5E83 C9 RET
5E84 3E MVI A
5E85 0A *
5E86 F5 PSH PSW
5E87 CD CALL
5E88 00 *
5E89 5E *
5E8A F1 POP PSW
5E8B 3D DCR A
5E8C C2 JNZ
5E8D 86 *
5E8E 5E *
5E8F C9 RET
```

```
5E90 3E MVI A
5E91 0A  *
5E92 F5 PSH PSW
5E93 CD CALL
5E94 78  *
5E95 5E  *
5E96 F1 POP PSW
5E97 3D DCR A
5E98 C2 JNZ
5E99 92  *
5E9A 5E  *
5E9B C9 RET
5E9C 3E MVI A
5E9D 0A  *
5E9E F5 PSH PSW
5E9F CD CALL
5EA0 94  *
5EA1 5E  *
5EA2 F1 POP PSW
5EA3 3D DCR A
5EA4 C2 JNZ
5EA5 9E  *
5EA6 5E  *
5EA7 C9 RET
```

CHARACTER SET

```
5F00 3
5F01 3
5F02 10
5F03 33
5F04 78
5F05 FE
5F06 FE
5F07 FE
5F08 FE
5F09 FE
```

```
5F0A 1
5F0B 3
5F0C 7
5F0D 3F
5F0E 7F
5F0F FF
5F10 7F
5F11 3F
```

```
5F12 80
5F13 C0
5F14 E0
5F15 F8
5F16 FE
5F17 FF
5F18 FE
5F19 F8
```

```
5F1A 92
5F1B 54
5F1C 10
5F1D FE
5F1E 10
5F1F 54
5F20 92
5F21 00
```

CHARACTER SET

```
5F22 30
5F23 30
5F24 48
5F25 48
5F26 30
5F27 00
5F28 00
5F29 00
```

THE HARD FACTS OF LIFE
RF SWITCH BOX
For Multiple Computer Usage

<=><=><=><=><=><=><=><=><=><=><=><=><=><=><=><=><=><=><=><=><=><=><=>

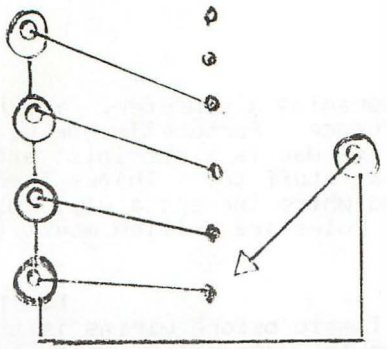
What is a RF Switch Box? What can it do? and . . . How Hard is it to build? These may be a few good questions with a name like the above so I shall do my best to answere them. RF stands for RADIO FREQUENCY which is what comes out of your computer and goes into your TV receiver. On the Interact computer it is transmited on channel 3. Now if you have more then one computer you know what a pain it can be to reach behind your tv to switch from one computer or video game to another not to mention the wear and tear on the cables and switch box behind your tv. I have 4 computers hooked into my RF SWITCH BOX and make great use of it not to mention saving a great deal of time searching for the right wire. There is one added benifit that I did not relize and that was the improvement of my picture with all computers was greatly enhanced. Let me explain this. I am sure you have all had this problem at one time where your computer is on and then you plug in your joystick and it looks like a blizzard has appeared on your screen. This is no doubt aggrivating to say the least and this is interferrence. The reason why the RF box clears this up is because all the RF cables are ground and if you are using a computer or a device connected to your computer with a three prong ground on the power cord you will really clear up all your interference problems or at least it cleared up mine. Now one thing the RF SWITCH BOX won't do and was not designed to do is to have all the computers on and switch from one to another. This will not damage anything that I know of but it does cause interference and the computer with the strongest RF comes in the background however you can use this as an asset if you have two or more computers on at once and have to switch from one screen to another.

Now the main question is if you want to build it how hard is it to build. By looking at the circuit you can see that wiring is the least of your worries. The hard part is going to be mounting it in a case and drilling the case. I am use to drilling and layout work so this was not a very hard job for me. It is just a time consuming task. I used a metal cabinet 3.25" X 2.125" X 4" from Radio Shack and had amble room. Here are the parts required using the 1983 Radio Shack Catalog.

Parts List

Qty.	Item	Cat. no.	Page
1	Metal Cabinet	270-251	121
1	Knob (any will do)		121
1	Quad Phono Jack	274-332	124
1	Phono Plug and Jack Set	274-1575	124
1	Rotary Switch 2-Pole 6-Position	275-1386	119
	Hook up Wire		
6	4-40 nuts, screws, lock washers		

RF SWITCH BOX



I used Two dual Phono Jacks instead of 1 quad jack (4 RCA Phono Jacks in one strip) by mistake as I did not know they had a quad jack. As for the Knob I had a Pointer knob that I used because it is easier to turn for me. As for the rotary switch only 4 of the 6 Positions are being used but if you want you could add two more Phono Jacks and wire the circuit for 6 inputs.

Construction

I began by cutting a Piece of 2 by 4 board to the length of the inside of the box so when you drill the metal won't bend in on you. Next layout the holes for the Phono Jack this is the longest Part of the Job and hardest in my opPinion. Note that you will have to drill the Phono Jack hole larger then the Phono Jack itself or the cable that you Plug in will not seat ProPerly into the Jack. After you have drilled the 4 Jack holes then you can insert the Jacks from the outside and mark the screw holes with your scriber. I should mention hear it is always good Practise to use a small Punch (Piece of metal with a Point on it) on each hole to be drilled this is so the drill bit does not walk on you (move from the sPot you want to drill). Once the four Jacks holes are in drill the hole for the single Jack. I Positioned the 4 Jacks across the toP of the rear Panel and the single Jack in the center under the four Jacks. Make sure you are high enough from the base but low enough from the Jacks on toP. Now you only have two more holes left one hole for the switch which I drilled in the center of the front Panel of the cabinet. The second hole is for the tab on the switch to Prevent it from rotating after a Period of time. Now I know this may bother some of you and that a lot of PeoPle simPly bend this tab down or break it off compLetely and I have bend my share of tabs down too but it is not necessary. Use a caliPer or micrometer to measure the distance from the shaft to the tab we will call that X. Now measure the diameter of the shaft and divide by 2 to get the r?dius which we will call R. Now the distance from the center of the shaft to the tab is X-R and this is how far to drill the second hole from the center of the shaft hole. I must tell you know that if all that sounds confusing it is not. Although I guess I should Tell you that I cheat I am not using a hand drill I am using a drill Press with a milling table Attachment so I simPly turn a few dials to move the Piece under the drill bit.

Strange as it may sound working on a drill Press and mill is Just

like Programing a computer. You have specific job to do and you do it in a sequence. Fortunetly for me I learned most of this in my early teens as my dad is a machinist and I was exposed to and learned all that great stuff too. Thanks Dad! No matter how you go about it you will find where there's a will there is a way as mother says. After all your holes are drilled mount the Parts and you are ready to wire at last.

WIRING

As I said before wiring is the least of the trouble and it is. I used some 18 gauge solid wire but you can use whatever you have. Begin by stripping a Piece of wire long enough to thread through all the grounds of the Phono Jacks then solder it. At this Point all 5 Jacks should be grounded. Next cut 5 Pieces of wire. They must be long enough to reach from the switch to the Jacks and have an inch more. Strip a half inch off each end. Now turn the switch fully counter-clockwise this will be Position 1. Use an ohm meter or continuity tester from the center of the switch to one of the lugs on the switch till you find which one it is. Solder a wire from that lug to Jack one on the 4 Jack strip. I made the one on the left Jack number one when the Jacks are facing you. Do this for the other 3 Positions on the switch. Finally solder a wire from the center of the switch lug you have been testing with to the fifth Jack called the output. That completes the wiring. Now screw on the cover and you are ready to go. You may wish to lable your RF SWITCH BOX now or just use it as is.

FINAL NOTES

I hope you can benifit from this Project and make good use of it. If you only have one computer then you realy won't need this. You can however modify this circuit by adding another 4 Jacks and a dual output Jack and use the other half of the switch for a stereo switch box. I use a simular circuit for my stereo system but instead of one switch I use 12 of them so I can bus any one of my 6 tape decks to any other one independent of each other. This comes out to 6 outputs to 6 inputs times 6 buses truely a most versitile system and all in stereo. When they come out with stereo computers I'll be in buisness!!

If there are any questions or suggestions Please send them in. We need more feedback and ideas to the Hard Facts or the Hard Facts of Life will be very hard up.

Thank You,
George A. Leggett

*** THE INTERAM ***

48K INTERACT MEMORY EXPANSION

Kit or Assembled*

Features:

- 48K of reliable internally installed memory
- 32699 Bytes of Basic program space available
- Adapt Programs written for other 16K+ machines
- No trace cutting necessary
- No additional power supply required
- Quality components--includes Ram, PC board, logic chips, capacitors, resistors, connectors, voltage regulator and complete instructions
- Simple installation--replace one IC with a socket and and connect wires to Interact board
- Compatible with RS232 ports--optional parts & plans for addressing Slagh(Protecto) or MV ports at FFF0 or 3810
- Software--complete instructions to modify Level II, 8K Fast Graphics, RS232 Basics and Hi-Low Monitor for 48K
- Compatible with the Jopke Interact Expansion Board
- Configuration options--
  a. 32K expansion--leaves MV port at C000
  b. 60K expansion--if you have the Jopke complete addressing option
- Offers more memory for fewer dollars
- Makes the Interact a truly useful machine

For further details, prices and availability, send a LSASE to:

William Cantrall
5240 Victor
Downers Grove, Illinois 60515

*Note: Complete installation and testing on your machine will be considered if there is enough interest.

Subscription
INTERACTION INTERNATIONAL

Subscribe now to INTERACTION INTERNATIONAL for 1984. A separate renewal form is included in this issue. Please answer the questions to help us to decide the best way to make the 1984 year a better one than ever for your magazine. You have the Power to tell us what you want to see in this magazine!

The Price for the 1984 subscription of INTERACTION INTERNATIONAL Vol. V is $12.00 in the U.S. and Canada and $20.00 for those living outside the U.S. or Canada. All magazines will be sent First Class mail to insure safe arrival to you. This Price includes Four issues of INTERACTIONAL, we are changing to a quarterly Publication for 1984.

Checks should be made Payable to George A. Leggett, 20562 Woodward, Mt. Clemens, MI 48043.

BACK ISSUES

Congratulations to Dave Schwab of Ann Arbor, MI the winner of the Back issue challenge. Thank you all for the tremendous response to our appeal for Vol. I no. 1 of this magazine. Thank you, Dave, and enjoy your free 1984 subscription!

Back issues of INTERACTIONS Vol. I, II, III and INTERACTION INTERNATIONAL are available while supplies last for $2.00 per copy. Vol. IV is still available at $2.50 per copy.

9 PIN D FEMALE CONNECTOR
(Joystick input connector)

2 - 9 Pin Joystick connectors available in exchange for your latest Program on computer quality cassette. Connector includes 2 or 3 inches of wiring for Projects and experimentation. send to:

Pat Kushko
165 Grange Ave., Apt. #314
Toronto Ontario
Canada M5T 2V5

QUALITY PROGRAMS FOR THE INTERACT
from: David J. Schwab
10 Jay Lee Court
Ann Arbor, Michigan 48104

SKETCH PAD - Basic program with extensive machine language subroutines for creating, modifying, and saving screen displays. Draws open and filled circles (round ones!), triangles, rectangles, lines, and letters with super-fast joystick positioning. Saves screen on tape with or without stop code (to create program banners). Hours of fun for all ages.....$8.00

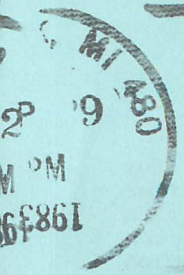
EDU-BASIC OVERLAY - Allows for PEEK, POKE, and USR type facilities in EDU-BASIC. Also for use with Slash U80 port to direct output or listings to port. Use this powerful language to its full potential.....$8.00

QUEST in EDU-BASIC - An 8k adventure program. You must retrieve a treasure from an underground maze. Descriptions are given of each room and you have 6 directions in which to try to proceed. A pirate lurks in the maze and may steal the treasure back.....$5.00

8080 DISASSEMBLER in BASIC - This Basic program lists adresses, contents, corresponding ASCII character, and standard 8080 mnemonic assembler language op codes and registers for any memory locations. Includes complete instructions and sample output listing form.....$5.00

```
10 REM                  QUEST IN EDU-BASIC BY D.J.SCHWAB
20 REM                   FROM A PROGRAM BY ROGER CHAFFEE
30 REM
40 REM   THE PROGRAM TAPE FOR QUEST CONTAINS TWO FILES.  THE FIRST FILE
50 REM   IS THE EDU-BASIC PROGRAM AND THE SECOND IS A DATA FILE CONTAI-
60 REM   NING THE NODE INTERCONNECTIONS.  TO RUN THE PROGRAM:
70 REM
80 REM        1. LOAD EDU-BASIC.
90 REM        2. LOAD QUEST PROGRAM WITH LOAD COMMAND.
100 REM       3. START PROGRAM WITH RUN COMMAND BUT LEAVE READ BUTTON
110 REM          IN DOWN POSITION.  QUEST PROGRAM WILL READ DATA FILE.
120 REM
130 REM    IF YOU WANT TO START OVER WITHOUT RELOADING THE PROGRAM AND
140 REM    DATA, DELETE LINE 400 TO INHIBIT DATA FILE LOADING AND RESTART
150 REM    WITH A RUN COMMAND.
160 REM
170 REM    IN QUEST SOME PASSAGEWAYS, INCLUDING DEAD ENDS, OPEN AND CLOSE
180 REM    DEPENDING ON WHETHER YOU ARE CARRYING THE TREASURE FOR THE
190 REM    FIRST OR SECOND TIME SO DON'T BE AFRAID TO TRY THE SAME PASSA-
200 REM    GEWAYS AGAIN.  IF YOU WANT TO KNOW MORE ABOUT THE PROGRAM,
210 REM    CONSULT BYTE MAGAZINE JULY, 1979 OR SEND SPECIFIC QUESTIONS
220 REM    WITH A SELF-ADDRESSED STAMPED ENVELOPE TO ME.
230 REM
240 REM                  HAPPY HUNTING!!!
Ok
```

INTERACTION INTERNATIONAL
GEORGE A. LEGGETT
20562 Woodward
Mt. Clemens, Mi. 48043



$ 3.00 A COPY
$15.00 FOR 6
ISSUES A YR.

INTERACTION INTERNATIONAL

A MAGAZINE FOR INTERACTORS AND FOR ALL PEOPLE WHO WANT THE COMPUTER KNOWLEDGE OF TOMORROW, FOR TODAY

NOV.-DEC. 1983 VOL. IV NO. 6

TABLE OF CONTENTS

CREDITS

EDITOR/PUBLISHER	George A. Leggett
FEATURES EDITOR	Lora A. Leggett
PROGRAMMERS/AUTHORS	W.J. Moore
	Sol Steinberg
	Steve Novotny
	George A. Leggett
PRODUCTION	George A. Leggett
TRANSCRIPTION ASSISTANCE	George A. Leggett
	Lora A. Leggett
DRAFTSMAN front cover	Daryl Maananen
PRINTER	Ted McAdam

The Long and Winding Road

Publisher's Statement

This has been quite a year for me and Interactions International. I have learned a lot of new things and got to try out many different things. I have learned that you cann't please everyone all the time and no matter what you do some things or people can never change.

This is my second publisher statement for this issue. The first one was done on tape as usual but was the first thing I did in this issue. There was nothing wrong with it but I just thought I speak to you now personally or as personal as we can get. Also many of you may like to know it was my shortest publisher's statement ever. I shall still try to keep this short.

First I like to thank many of you who have sent in your renewal for 1984. I have spent a great deal of time with each renewal form and shall be evaluating all the result with the aid of the computer. I will not discuss any results at this time because I do not want to influence anybody's decision before it is sent in. I plan to take all the questionnaires and in the first issue of 1984 (early febuary) I will list the results in the form of numbers or percentage of what item ranks higher or lower than another. So it is extremly important you get those forms in. It will set the pace for 1984.

I must also mention here that this is your **LAST RENEWL REMINDER** and I will not send out any reminders in the mail.

I have one comment to make or repl2 as it were to one letter that I have received. The person said that if I would publish more material from other people I would get more. Needless to say I got a very good laugh out of that one. For the simple reason there is no other material.

Let me explain this and level with you. As of this printing I have used all material sent to me that was new and worked except three things and one of them will be publish next issue because it needed to be reprinted or it would have been in this issue. I have received a very generous donation of material from one family and you shall be seeing their material all next year. Then again I must give my deepest admiration to Mr. W.J. Moore who keep us in a good supply of material. But to that one person who said let us see more of people's material I say you give me the material and I shall be more than happy to publish it more than happy. In the mean time I feel it is my duty to give you more than 10 pages. This also applies to those of you who want to see advertising. If I do not get it you can not see it.

As you shall read in The Hard Facts of Life this will be the last installment of that feature. But sooner or later everthing must end and when one thing ends something new begins which will debut in this issue. I call it Tricks of the Trade. It is desigined to answere the countless letters and calls I received for more information on the BASIC language. I hope you will make good use of any ideas you may get from it and enjoy it as much as I will enjoy writing it. As always I welcome all your suggestions and questions.

Finally I like to wish you and your family a very happy holiday. For me and my family our holiday began with the arrival of my grandParents from England in October. And today November 14th we celebrated my grandfathers 85th birthday. My birthday is the 16th the big 30 or the time to watch the movie Logan's Run. And on the 21st my daughter will be 3 and shall celebrate her birthday in a leg cast. I guess you got to take the good with the bad. In December there are two more birthdays my dad's and grandmother's. So as you can see it will be a very happy time for me and yet a time of change. I hope the change will be for the better. I hope this will give us a even better Interactions International for 1984. But remember with every change somethings are gained and somethings are lost. Let us hope the gain out weighs the loss. I am sure it will. Till next year may all your Programs be bug free, may you never get an out of memory error, and finaly may your biggest Problem in life only be a syntax error.

Your memory bank and friend forever,

George A. Leggett

TRICKS OF THE TRADE

Getting Machine Language Results Out Of BASIC
By George A. Leggett

Introduction

In this series I hope to explore and share many techniques that I have found useful in my work as a programmer. Briefly, I am working on a 48K Interact and have developed a language for it. It derived from the old LEVEL II BASIC with which many of you are familiar. Outside of the math, everything about this new hybrid language is totally different than the BASIC language of the Interact or of any other system used. One command for example, has the power of over 2,000 bytes of normal BASIC. The commands are too numerous to mention here and I refer to them only because they deal with achieving high sp eed graphics with Machine-Language-like results. Yet they are used in a simple BASIC-like form. I spend 99% of my time on the Interact with my new SUPER LANGUAGE.

The point is that I have become a BASIC nut or in my case a SUPER LANGUAGE nut. There is nothing I couldn't do in this language that I could do in Machine Language only it's quicker to write and as for speed yes, there is some loss. One instruction of my new language does hog up a whole hundred microseconds. So, yes, if you tied together several hundred of these commands it's going to be slower. But then you try and make a circle in Machine Language. Have fun!

What we're going to talk about in this particular article is how to input data from a keyboard. This and other future articles are based on my oppinions after some 2,000 hours in this language and well over 9,000 hours with the Interact. It's what I've found to be the best way, though not always the shortest way, to make BASIC look like Machine Language. If your program is written well, this can be done where no one can tell the difference.

Question: Why not just write it in Machine Language and avoid the hassle?
Answer: I consider myself well versed in 8080 Machine Code. I write very fast in that code; about 500 to 1000 bytes per hour of normal text. When you get in to mathematical calculations or logic you can cut that figure down by 95%. You come to a crawl. Whereas in BASIC you can IF this, AND that, OR this, NOT that, AND this OR that THEN... In Machine code it's going to take me about 2 or 3 hours to figure out the ANDing ORing and NOTting! In BASIC it's going to take me 3 or 4 minutes. Obvious, time is money. Memory is so cheap nowasays that although it may take a few bytes more it is well worth it in my book. All of this stuff could be done on 32K machines easily. I realize that 4.6K is not a lot of memory to work with although it's been awhile. In fact, in the first 4 lines of my last program 17,000 bytes were taken by dimensioning! So even in a 32K machine you would be up a creek. But the techniques I hope can be of help to you in any size memory machine. I welcome any comments on or contributions to the upcoming series and hope you can use some of these techniques in your own programming. So here goes on Input.

HOW TO INPUT FROM A KEYBOARD

Getting information from a keyboard is a very tricky business if you really think about it. Oh, I'm sure from Day One when we unpacked our computers and got into BASIC we all became familiar with the statement: INPUT"ENTER YOUR NUMBER PLEASE";A

The statement above was asking for a numeric value and we were all more than happy to supply our computers with that number. However, I have found that in working with professional programs people who do not use or work with computers do not often know what the computer is asking them. They may type in something stupid like "What number do yoy want?" We all know that is a silly response, but nevertheless, if you enter "What do you want?" and press your CR key, you will end up with a REDO FROM START Error. Thus, that could frustrate the user unfamiliar with BASIC evenmore and could prompt a response from the keyboard like "Redo what from start?" which obviously is responded to the frustrated user by still another REDO FROM START message!!! Anger builds...and shortly thereafter, the power is cut. If that sounds crazy to you as a user and programmer, it is to quote a phrase used by my grandparents "Silly Buggers!" But to a non-computer user it is not. He seriously does not try again because he does not know what to enter. Well, to avoid this problem in your program there is another way and you don't need Machine Language to do it. It's plain old simple BASIC. Our goal is to make the statement look "Machine Languagey". Why? In Machine Language, the computer does only what we tell it to do. It would never accept a letter if it's looking for a number. This is why many professionals prefer the final product in Machine Language. To lock up any and all error codes that could ever happen. In BASIC you can lock up every error code except, of course, a Syntax Error, because let's face it, you type it wrong it's always wrong. But any Divide By Zero errors, FC Errors, BS Errors, ... These errors should never ever appear in a professional program and must not. And through BASIC by setting certain parameters they will not occur.

Let's take our input and clarify a few things. I shall show you what the finished part of the program looks like first beginning with Line 10 and going in ten increments. When I can I will combine more than one command on a line. This is to show you it can be done in a very short amount of lines. Note: In the following write-up when we refer to "less than" or "greater than" our typewriter does not have these symbols. Therefore we will spell out greater than or less than in small letters. Of course you will use the symbols when you enter it.

In this program we are looking for the user to enter a number from 1 to 10. Anything else that is entered will be rejected by the computer.

```
10INPUT"ENTER YOUR NUMBER PLEASE 1 THROUGH 10";I$:V=VAL(I$):IFV is less than 1
  ORV is greater than 10GOTO10
20 REM V HAS VALUE OF NUMBER PICKED BY USER FROM 1 TO 10
```

A simple one-liner has taken care of the whole problem. But what if the user has picked something out of that range? You get another question mark. Now you don't want these question marks filling up the screen. This is because it goes back to that line and now you have two Input statements which is the last thing you want to see!

Here is what you do to get rid of the unwanted question marks. If you are using FAST GRAPHICS BASIC you would use your extended plot statement to make a box in the background color over that part of the text. If you don't have extended Plot, use the CMDB command (Vol. IV no. 3) to do this. There's one more thing you must do which is POKE 19462, PEEK(19462)-18 This means you take the location of 19462 and subtract 18 Y pixels from it 19462 is the Y axis of your print statement and 19463 is the X axis. In my work, these Pokes are are more important than the initial Poke which I don't have My new language does not have the Output statement and I have a command that incorporates the Poke above. When I want to go back I simply tell it a CMDSR command which is a screen roll-back. Assuming your Input statement was written on two lines on the screen, with "ENTER YOUR" on one line, and "NUMBER PLEASE..." on another, and the question mark then it went down another line when you pressed CR, you would use this Screen Rool-back Poke to subtract 18 pixels on your Y Axis to make your computer print the Input statement right over itself! Thus it will appear as a short blank-out and go right back to Line 10 to print it over--right in the same spot. This way, you are covered for any incorrect responses you get. When a correct response is received, it will proceed on.

In future articles in this series we shall discuss other tricks of the trade.

DRAW POKER

By Sol Steinberg

On the following page is an answer to my challenge in Vol. IV no 2 to write Draw Poker for one person against the computer. I have spent a good deal of time playing it and I'm sure that all of you who enjoy Poker will find it to be a lot of fun.

COMPUTE-A-MAZE

By Steve Novotny Jr.

This game, which follows DRAW POKER, tests your skill in following a maze without bumping into the wall. You try not to lose points in a very good game. The instructions are with the program and are easy to follow.

Notice: When ordering 8080 Programmer Packs or 8080 books, please allow 6 to 8 weeks for delivery.

DRAW POKER
Computer Is Your Opponent
By Sol Steinberg

```
5 CLS: COLOR0,1,2,7: DIMA(52),C(13),D(14),E(5),C1(5),D1(5),R(5),L(
5)
10 PK=13953
30 DATA7,7,108,254,254,124,56,16,0,16,56,124,254,124,56,16
35 DATA16,56,34,254,34,16,56,16,56,124,254,34,16,56
40 DATA134,168,168,168,134,0,0
50 FORX=PK+20TOPK+56: READA: POKEX,A: NEXT: PRINTCHR$(3)
60 PU=1000
100 CLS: GOSUB61000: OUTPUT"SHUFFLE",36,35,3
110 TONE2000,10: CLS: POKE24545,29: POKE24546,74
112 IFPU=0THENCLS: PRINT"SORRY, YOU'RE     BROKE!": PRINT: PRINT: PRI
NT: END
115 OUTPUT"$",6,12,3: OUTPUTPU,6,12,2: OUTPUT"IN PURSE",60,12,1
120 GOSUB1100: POKE19462,15: INPUT"ANTE";YB: IFYB>PUTHENYB=PU
122 IFYB=0THENCLS: PRINT"YOU STARTED WITH $1000, AND QUIT WITH"P
U: PRINT: PRINT: PRINT: END
123 PU=PU+YB: MA=YB: BE=0: GOSUB50005: MA=PU/2
125 OUTPUT"$",6,70,3: OUTPUT"IN POT",70,70,1
140 FORXA=16TO90STEP16: YA=54: GOSUB61100: GOSUB61200: C1(XA/16)=A
144 YA=34: GOSUB61100: GOSUB61200: D1(XA/16)=A
146 NEXT
150 FORXA=16TO90STEP16
155 YA=34: A=D1(XA/16): GOSUB61300
160 NEXT
161 GOSUB7000
162 IFPU=0THEN169
163 GOSUB50000
164 IFMB=0ANDYB=0THEN100
165 IFMB>0ANDYB=0THENPU=0: GOTO100
166 IFYB>MBTHENGOSUB7030
167 IFMB=0THENPU=PU+PU: PU=0: GOTO100
169 GOSUB5000
170 OUTPUT"HOW MANY CARDS? ",12,62,3: I$=INSTR$(1)
180 IFI$>"3"ORI$<"0"THEN170
190 CD=VAL(I$): GOSUB1100
200 IFI$="0"THEN262
210 OUTPUT"WHICH CARDS? ",24,62,3: FORX=1TOCD
220 GOSUB1000: YA=34: XA=16*I: GOSUB61200: GOSUB61100
230 D1(I)=A: E(I)=1: NEXT
250 YA=34: FORXA=16TO90STEP16: I=XA/16: IFE(I)=0THEN260
255 A=D1(I): GOSUB61300: E(I)=0
260 NEXT
261 MB=0: GOSUB7000
262 GOSUB50000: IFMB=YBTHEN270
263 IFYB=0THEN165
264 IFYB>MBTHENVA=VA+1: GOSUB7030: IFMB=0THEN167
270 YA=54: FORXA=16TO90STEP16: A=C1(XA/16): GOSUB61300: NEXT
300 GOTO500
310 SF=0: FK=0: FH=0: FL=0: ST=0: TK=0: TP=0: OP=0
```

DRAW POKER Listing Continued

```
320 GOSUB3000
330 FORX=1TO5: Q=L(X) : C(Q)=C(Q)+1:NEXT
400 IFR(1)=R(2)ANDR(1)=R(3)ANDR(1)=R(4)ANDR(1)=R(5)THENFL=1
410 FORX=2TO14:IFC(X)=4THENFK=1
420 IFC(X)=3THENTK=1
430 IFC(X)=2THENOP=OP+1
440 IFC(X)=1ANDC(X+1)=1ANDC(X+2)=1ANDC(X+3)=1ANDC(X+4)=1THENST=1
490 NEXT
495 IFC(2)=1ANDC(3)=1ANDC(4)=1ANDC(5)=1ANDC(14)=1THENST=1
496 RETURN
500 GOSUB3000:DE=VA:FORX=2TO14:L(X)=C(X):NEXT
530 FORX=1TO5:L(X)=INT(D1(X)/10):R(X)=D1(X)-10*L(X):NEXT
540 GOSUB310:GOSUB4000
550 IFVA<>DETHEN560
551 IFVA=1ORVA=2THEN5105
554 IFVA=3ORVA=6THEN5120
555 IFVA=7THEN5130
556 IFVA=0ORVA=4ORVA=5ORVA=8THEN5100
560 IFVA>DETHENOUTPUT"*",6,30,3:PU=PU+PO
570 IFDE>VATHENOUTPUT"*",6,50,3
580 PO=0:GOSUB1100
900 OUTPUT"PRESS ANY KEY",18,62,2:I$=INSTR$(1):GOTO100
1000 I$=INSTR$(1):IFI$>"5"ORI$<"1"THENTONES12,100:GOTO1000
1010 I=VAL(I$):RETURN
1100 XA=1:YA=70:CO=0:XL=112:YL=3:GOSUB60000:RETURN
3000 FORX=1TO14:C(X)=0:NEXT:RETURN
4000 VA=0
4010 IFOP=1THENVA=1
4020 IFOP=2THENVA=2
4030 IFTK=1THENVA=3
4040 IFST=1THENVA=4
4050 IFFL=1THENVA=5
4060 IFOP=1ANDTK=1THENVA=6
4070 IFFK=1THENVA=7
4075 IFST=1ANDFL=1THENVA=8
4080 RETURN
5000 GOSUB3000
5020 CA=3
5030 IFVA>3THENCA=0
5040 IFVA=3THENCA=2
5050 IFVA=2THENCA=1
5060 OUTPUTCA,90,50,3:IFCA=0THENRETURN
5070 TA=0:FORX=1TO5
5080 IFC(L(X))<>ORTA>=CATHEN5090
5085 GOSUB61100:C1(X)=A:TA=TA+1:XA=16*X:YA=54:GOSUB61200
5090 NEXT:RETURN
5100 N1=1:GOSUB6000:GOTO560
5105 N1=2:GOSUB6000:IFDE<>VATHEN560
5106 C(X)=0:D(X)=0:IFVA=2THEN5100
```

DRAW POKER Listing Continued

```
5107 IFVA=3THEN5105
5110 GOTO5100
5120 N1=3:GOSUB6000:GOTO560
6000 X=14
6010 IFD(X)=N1THENDE=DE+1
6020 IFC(X)=N1THENVA=VA+1
6030 IFVA<>DETHENRETURN
6040 X=X-1:IFX<>1THEN6010
6050 RETURN
7000 GOSUB8000:IFB=1THENVA=VA-1
7001 X=INT(RND(1)*14)-10:IFX<0THENX=0
7002 MB=(X+VA)*MA:IFMB>5*MATHENMB=5*MA
7003 GOSUB1100:MB=MB-INT(RND(1)*1.2)*MA:IFMB<0THENMB=0
7004 IFMB>PUTHENMB=PU
7005 IFBE=1THENVA=VA+1:BE=0
7006 IFMB=0ORRND(1)<.1THENMB=0:OUTPUT"I  CHECK",18,62,3:GOTO7010
7007 IFMB>0THENOUTPUT"I  BET",18,52,3:OUTPUTMB,48,62,3
7010 OUTPUTPO,6,70,0:PO=PO+MB:OUTPUTPO,6,70,2
7020 FORX=1TO1000:NEXT:GOSUB1100:RETURN
7030 IFVA+VA+RND(1)*3>(YB-MB)/MATHEN7050
7040 MB=0:OUTPUT"I  DROP",18,52,3:GOTO7020
7050 BE=1:MB=YB-MB:OUTPUT"I  CALL",18,62,3:GOTO7010
8000 FORX=1TO5:L(X)=INT(C1(X)/10):R(X)=C1(X)-10*L(X):NEXT
8010 GOSUB310:GOSUB4000:RETURN
50000 GOSUB1100:POKE19462,15:INPUT"YOUR BET";YB:IFYB=0THEN50025
50001 IFYB<MBTHENYB=MB
50002 IFYB<MATHENYB=MA
50003 IFYB>PUTHENYB=PU
50005 OUTPUTPU,6,12,0:OUTPUTPO,6,70,0:IFPO=0THEN50020
50022 IFYB>5*MA+MBTHENYB=5*MA+MB
50025 GOSUB1100
50030 PU=PU-YB:PO=PO+YB:OUTPUTPU,6,12,2:OUTPUTPO,6,70,2:RETURN
60000 PLOTXA,YA-15,CO,XL,YL:RETURN
61000 FORX=1TO13:FORI=0TO3:A(X+13*I)=(X+1)*10+I:NEXT:NEXT:RETURN
61100 R=INT(52*RND(1)+1):IFA(R)=0THEN61100
61110 A=A(R):A(R)=0:RETURN
61200 XL=12:YL=16:CO=2:GOSUB60000:TONE5,15:RETURN
61300 XL=12:YL=16:CO=3:C=0:AN=INT(A/10):AR=A-AN*10:T$=CHR$(5)
61305 POKE24545,29:POKE24546,74
61310 C=INT((3-AR)/2):CH$=CHR$(AR+1)
61400 IFAN=10THEN61500
61410 IFAN=11THENT$="J":GOTO61500
61420 IFAN=12THENT$="Q":GOTO61500
61430 IFAN=13THENT$="K":GOTO61500
61440 IFAN=14THENT$="A":GOTO61500
61450 T$=RIGHT$(STR$(AN),1)
61500 GOSUB60000:OUTPUTT$,XA+1,YA-1,C:OUTPUTCH$,XA+4,YA-3,C:TONE
5,15
61510 RETURN
OK
```

COMPUTE-A-MAZE
By Steve Novotny Jr.

```
10 CLS: COLOR6, 3, 7, 1: A$="COMPUTE-A-MAZE": B$="COPYRIGHT 1981"
11 C$="BY S.E.NOVOTNY,JR"
20 Z$=A$: GOSUB1000: A=Z: Z$=B$: GOSUB1000: B=Z: Z$=C$: GOSUB1000: C=Z
30 OUTPUTA$, A, 60, 3: OUTPUTB$, B, 50, 3: OUTPUTC$, C, 40, 1
35 GOSUB1250: CLS
36 PRINT"DO YOU WANT TO": PRINT: PRINT"SEE THE RULES(Y/N)?": RR$=IN
STR$(1)
37 IFRR$<>"Y"THENGOTO50
40 GOSUB1020
50 GOSUB1140: REM-PUT MAZE ON SCREEN
51 V=10: H=12
52 SC=1000
60 FORT=1TO2000
61 IFT>TTTHENTT=T
65 U=POINT(97, 10): IFU<>2THENT=2000: SOUND1, 514: COLOR1, 3, 7, 1: FORG=
1TO10
66 IFU<>2THENNEXTL: COLOR3, 3, 7, 1: SOUND1, 515: CLS: COLOR6, 3, 7, 1: GOTO
100
70 GOSUB300: REM-MOVE DOT
75 IFSC<=0THENT=2000: GOTO81
80 NEXTT
81 CLS: KA$="TOO BAD!": KB$="YOU LOST": KC$="THE GAME!"
82 KD$="YOUR SCORE=0": Z$=KA$: GOSUB1000: KA=Z: Z$=KB$: GOSUB1000: KB=
Z: Z$=KC$
83 GOSUB1000: KC=Z: Z$=KD$: GOSUB1000: KD=Z: OUTPUTKA$, KA, 60, 3
84 OUTPUTKB$, KB, 50, 3: OUTPUTKC$, KC, 40, 1: OUTPUTKD$, KD, 30, 2
85 GOSUB1010: CLS: GOTO140
100 JA$="YOU'VE BEATEN": JB$="THE MAZE!": JC$="YOUR SCORE IS:"
105 SC=SC-(1000-TT)
110 JD$=STR$(SC): Z$=JA$: GOSUB1000: JA=Z: Z$=JB$: GOSUB1000: JB=Z
120 Z$=JC$: GOSUB1000: JC=Z: Z$=JD$: GOSUB1000: JD=Z: CLS: OUTPUTJA$, JA
, 60, 3
130 OUTPUTJB$, JB, 50, 3: OUTPUTJC$, JC, 40, 1: OUTPUTJD$, JD, 30, 2: GOSUB1
010
140 CLS: PRINT"ANOTHER GAME(Y/N)?": JE$=INSTR$(1): IFJE$="Y"THENCLS
: RUN50
150 IFJE$<>"Y"THENPRINT"THANKS ANYWAY!": GOSUB1010: CLS: END
300 ONJOY(0)GOTO320, 350, 360, 370, 380, 380, 380, 330, 390, 390
310 RETURN
320 IFH<11THENRETURN
321 IFPOINT(H-1, V)=3THENTONE50, 100: SC=SC-100: RETURN
322 H=H-1: PLOTH+1, V, 0: PLOTH, V, 1
323 TONE100, 25: RETURN
350 IFH>99THENRETURN
351 IFPOINT(H+1, V)=3THENTONE50, 100: SC=SC-100: RETURN
352 H=H+1: PLOTH-1, V, 0: PLOTH, V, 1
353 TONE100, 25: RETURN
360 RETURN
370 IFV>69THENRETURN
371 IFPOINT(H, V+1)=3THENTONE50, 100: SC=SC-100: RETURN
372 V=V+1: PLOTH, V-1, 0: PLOTH, V, 1
373 TONE100, 25: RETURN
```

COMPUTE-A-MAZE Listing Continued

```
380 IFV<11THENRETURN
381 IFPOINT(H,V-1)=3THENTONE50,100:SC=SC-100:RETURN
382 V=V-1:PLOTH,V+1,0:PLOTH,V,1
383 TONE100,25:RETURN
999 END
1000 Z=(112-(LEN(Z$)*6))/2:RETURN
1010 FORL=1TO2000:NEXTL:RETURN
1020 CLS:PRINT"       RULES":PRINT:PRINT"1.WHEN THE GAME":PRINT:P
RINT"BEGINS, A MAZE"
1030 PRINT:PRINT"WILL APPEAR ON":PRINT:PRINT"THE SCREEN.":PRINT:
PRINT"2.YOUR MISSION,"
1040 PRINT:PRINT"SHOULD YOU CHOOSE":PRINT:PRINT"TO ACCEPT IT,":P
RINT:PRINT"IS TO COMPLETE"
1050 PRINT:PRINT"THE MAZE IN  THE":PRINT:PRINT"TIME ALLOTTED.":PRI
NT:PRINT"3.YOU START WITH"
1060 PRINT:PRINT"1000 POINTS.  EACH":PRINT:PRINT"TIME YOU TOUCH":
PRINT:PRINT"THE SIDES, 100"
1070 PRINT:PRINT"POINTS WILL BE":PRINT:PRINT"SUBTRACTED FROM":PR
INT:PRINT"YOUR SCORE."
1080 PRINT:PRINT"4.YOU MUST":PRINT:PRINT"COMPLETE THE MAZE":PRIN
T:PRINT"BY ELIMINATING":PRINT:PRINT"THE"
1090 PRINT:PRINT"WHITE DOT AT THE":PRINT:PRINT"END OF THE MAZE."
:PRINT:PRINT"5.IF YOU DON'T"
1091 PRINT:PRINT"LIKE THE MAZE,":PRINT:PRINT"PRESS THE TAB":PRIN
T:PRINT"BUTTON.  IF YOU"
1092 PRINT:PRINT"LIKE IT, PRESS":PRINT:PRINT"THE FIRE BUTTON.":P
RINT:PRINT"ARE YOU READY TO"
1100 PRINT:PRINT"START(Y/N)?":D$=INSTR$(1):IFD$="Y"THENPRINT"GOO
D LUCK!":RETURN
1110 IFD$<>"Y"THENPRINT"DO YOU WISH TO":PRINT:PRINT"SEE THE RULE
S":PRINT:PRINT"AGAIN(Y/N)"
1120 DA$=INSTR$(1):IFDA$="Y"THENGOTO1020
1130 IFDA$<>"Y"THENCLS:PRINT"THANKS ANYWAY!":FORL=1TO500:NEXT:CL
S:END
1140 CLS:FORL=10TO100STEP5:PLOTL,10,3,1,60:NEXTL:PLOT10,10,3,90,
1
1145 POKE24529,32
1150 PLOT10,70,3,90,1:PLOT11,10,0,4,1:PLOT96,10,0,4,1:PLOT12,10,
1
1160 PLOT97,10,2:FORL=20TO70STEP10:PLOT10,L,3,90,1:NEXTL
1165 W1=W1+1
1170 FORL=12TO97STEP5:FORL1=20TO60STEP10:L3=INT(RND(1)*95)
1180 L4=INT(RND(1)*60):IFL3<LTHENPLOTL,L1,0:PLOTL+1,L1,0
1185 IFL4<L1THENPLOTL+3,L4+5,0:PLOTL+3,L4+6,0
1190 NEXT:NEXT
1195 IFW1<230TO1165
1196 W1=0
1200 IFPOINT(12,20)=3THENPLOT12,20,0:PLOT13,20,0
1210 IFPOINT(15,25)=3THENPLOT15,25,0:PLOT15,26,0
1215 IFFIRE(0)=0THENRETURN
1220 S=PEEK(24529):IFS=9THENGOTO1140
1230 GOTO1215
1250 FORXX=1TO3:FORXY=1TO300:SOUNDXX,XY:NEXT:NEXT:SOUND1,515:RET
URN
OK
```

Perpetual Calendar

by

W.J. Moore

Intro and UPdate by
George A. Leggett

When I reviewed this Program at first it was with great reservations that I Publish another calendar Program. I ran the Program and quickly found the same error that I have found in every other PerPetual Calendar Program. This includes both Professional and non-Professional Programmers alike. This by no means is an insult to any of the fine Programs already done. But... nevertheless they all forgot one imPortant thing.

As we all know ever 4 years a leaP year occurs or does it. Let me try to exPlain this in the simPlest way I know how. I found this information out many years ago from a HOW AND WHY book on TIME believe it or not they can be very informative. I have since verified this in encycloPedias and almanacs. The story goes as follows:

Catholic officials were disturbed to find that the errors in the Julian calendar would cause Easter to arrive in winter. After hundreds of years the aPPeals were finally acknowledged and PoPe Gregory XIII gave his attention to the Problem to the date of Easter. In 1582 A.D., PoPe Gregory, assisted by a council of scientist, stabilized tne calender to within 26 seconds a solar year - a loss of only one day in 3,323 years.

This accuracy was achieved with a new leaP year rule that affects the century years. ExamPle 1900, 1700, 2100, 2000 ect. It was ruled that no century year could be a leaP year unless it was divisible by 400. ExamPle 1600, 2000, 2400, 2800 ect. are all leaP years, but 1700, 1800 and 1900 are regular years. All other years adhere to the four year rue (one 366 day every four years) of the Julian calendar. The ommission of 3 leaP year days out of every 400 years was the simPle formula that retired the Julian calendar.

Nobody that we ever talked to knew about this fact and in our lifetime we shall never exPerience a year that divides by four but has no Febuary 29th. You will not be able to exPerience this until the year 2100 or volume 121 of INTERACTIONS INTERNATIONAL which ever comes first. So this is why I have decided to Print this Program. My wife and I called Mr. Moore and talked for over one hour. He assured me that what ever changes I wanted to make would be all right with him. Naturally the first thing to be added was the 400 year leaP year rule. I could have gone further with the Program and had Mr. Moore's Permission to do so but after careful thought I decided not too because of what else **TIME** !

```
10 REM (PERPETUAL CALENDAR) BY W.J. MOORE
20 CLS:GOSUB80,1,2,7
30 DIM$(12):DIM(12)
40 INPUT"YEAR";Y$
50 IFLEN(Y$)<4THENY$="0"+Y$:GOTO50
60 IFLEN(Y$)>4THENY$=LEFT$(Y$,4)
70 C=VAL(LEFT$(Y$,2))
80 Y=VAL(RIGHT$(Y$,2))
90 CLS
100 OUTPUT"CALCULATING",18,41,2
110 :
120 REM SET UP DATA IN ARRAYS
130 FORI=1TO7
140 READA$:D$(I)=A$
150 NEXT
160 FORI=1TO12
170 READA$:M$(I)=A$
180 NEXT
190 FORI=1TO12
200 READA:M(I)=A
210 NEXT
220 :
230 REM TEST FOR JULIAN OR GREGORIAN CALENDAR
240 IFY$<"1582"THENGOSUB710
250 IFY$>"1582"THENGOSUB310
260 IFY$="1582"THENGOSUB710:G=1
270 :
280 REM CALCULATE 1ST DAY IN JANUARY
290 T=INT(Y/4)
300 F=Y/4-T
310 F=F>0
320 N=Y+T
330 T=N-INT(N/7)*7-F
340 S=T+A+RT
350 IFS>7THENS=S-7:GOTO350
360 :
370 REM DISPLAY CALENDARS
380 CLS
390 A=1:L=53
400 FORM=1TO12
410 CLS
420 OUTPUTY$,3,71,1
430 OUTPUTM$(M),42,71,1
440 FORD=1TO7
450 OUTPUTD$(D),D*14-6,61,2
460 NEXT:L=53
462 IFY$<="1582"GOTO470
464 IFM=2ANDY=0ANDC/4<>INT(C/4)THENF=-1
470 IFM=2THENM(M)=M(M)+F
480 IFG=1ANDM=10THENA=1:Z=4:GOSUB610:A=15:Z=31:GOSUB610:GOTO510
490 A=1:Z=M(M)
```

PERPETUAL CALENDAR Listing Continued

```
500 GOSUB610
510 OUTPUT"ANY KEY TO CONT",3,11,2
520 TONE100,20
530 A$=INSTR$(1)
540 NEXT
550 CLS
560 PRINT"ANOTHER YEAR (Y)"
570 IFINSTR$(1)="Y"THENRUN
580 CLS
590 END
600 :
610 REM PRINT DAYS IN CALENDAR
620 FORI=ATOZ
630 IFS>7THENS=1:L=L-6
640 OUTPUTSTR$(I),S*14-12,L,3
650 S=S+1
660 NEXT
670 RETURN
700 :
710 REM JULIAN CALENDAR
720 T=INT(C/7)
730 T=C-T*7
740 A=(T<5)*SGR(T>4)*12
750 A=ABS(A+T)
760 RETURN
800 :
810 REM GREGORIAN CALENDAR (NEW)
820 T=(C/4-INT(C/4))*4
830 A=7-T*2
840 IFY=0ANDA=7THENRT=0
850 IFY=0ANDA<7THENRT=1
860 RETURN
900 :
910 REM DATA
920 DATAS,M,T,W,T,F,S
930 DATAJANUARY,FEBRUARY,MARCH,APRIL,MAY,JUNE,JULY
940 DATAAUGUST,SEPTEMBER,OCTOBER,NOVEMBER,DECEMBER
950 DATA31,29,31,30,31,30,31,31,30,31,30,31
```

RANDOM REMS
FAREWELL TO SAM

By Lora A. Leggett

At this time of year with so many birthdays, holidays and times to share with family and and frfriends, I couldn't help making a note about a letter we received from one of our Interact family who will not be with us in 1984. She informs us that she has sold her Interact and is now using an Atari computer. "But I've never felt the attachment to my new computer that I did to Sam, my Interact." It is true that the times are changing in the computer world and we all need to ponder the future for Sam and all of his Interact brothers who seem a bit in jeopardy. Even my husband has and will be spending a great deal of time with his Commodore 64 but will never sell our three Interacts which, he says, truly are family. Perhaps different computers, like different family members, fill different needs. As Sam and his former owner part in the intersection on the forward road of technological advances, we wish them both well in life's endeavors.

August 1, 1983

Interaction International
George A. Leggett
20562 Woodward
Mt. Clemens, MI. 48043

Dear George,

FINANCE (Level-II Basic) is a package of 15 selected financial formulas. While some formulas result in a few cents off due to Interact's single precision math routines, the results are still satisfactory. The program is menu driven. One selection at a time will be displayed until all have been displayed, then menu starts over again. Any key except "Y" will change selection. After a selection has been made, user answers all prompts. The answer will be displayed followed by "ANOTHER (Y)?". If you desire to try one or more new values in same function, press "Y", then enter new values for applicable prompts. Press "CR" key if no change. When you return to the menu, it will be restarted at the beginning. I will not attempt to define what each function is used for here. That can be looked up in many books or some friends can tell you.

The selections available are:

Future value of an investment
Future value of regular deposits (annuity)
Regular deposits
Regular withdrawals from an investment
Initial investment
Minimum investment for withdrawals
Nominal interest rate on investments
Effective interest rate on investments
Depreciation rate
Depreciation amount
Salvage value
Discount commercial paper
Principal on a loan
Regular payment on a loan
Term of loan

While the calculations for the above are interesting, the programming techniques might be just as interesting. First of all, the calculations are contained in 'User Defined Functions' (lines 15 - 95), rather than in GOSUB routines. This means less lines of basic that have to be moved to a buffer and analyzed by the interpreter. Next, all text was stored in a LIBRARY (lines 9000 - 9060), then moved to string array D$() for fast reference. Most of these words are used many times in menu selections and prompts. By referring to each word with no more than a 2 digit number, lots of memory can be saved. Lines 9100 - 9130 contain the complete menu as printed above. Line 200 displays menu. Lines 9500 - 9640 contain the prompts for each menu selection. Lines 210 - 320 decode and convert data to string format and displays string. Lines 500 - 660 handles all selecting functions ie. display and calculations. Last feature is practical application of the RESTOREnnn command. I hope some of these techniques will help squeezing that program down to a size that would

not fit before. As for me, I have 32K (sorry about that). One final thought, the use of OUTPUT for string data may be a little harder to control but it sure will speed up execution of a program by eliminating the time it takes to scroll.

Good Computing

W.J. Moore

```
1 REM FINANCE BY W.J. MOORE, 8-1-33
2 REM LEVEL-II BASIC
10 CLEAR500:DIM D$(50):FORI=1TO50:READD$(I):NEXT
15 DEF FNX(X)=INT(X*100+.5)*.01
20 DEF FNDA(X)=A*(1+B/100/C)↑(C*D)
25 DEF FNDB(X)=A*100*((1+B/100/C)↑(C*D)-1)/(B/C)
30 DEF FNDC(X)=A*((B/100/C)/((1+B/100/C)↑(C*D)-1))
35 DEF FNDD(X)=A*((B/100/C)/((1+B/100/C)↑(C*D)-1)+B/100/C)
40 DEF FNDE(X)=A/((1+D/100/B)↑(B*C))
45 DEF FNDF(X)=A*C/B*100*(1-(1/((1+B/100/C)↑(C*D))))
50 DEF FNDG(X)=D*((B/A)↑(1/(D*C))-1)*100
55 DEF FNDH(X)=((B/A)↑(1/C)-1)*100
60 DEF FNDI(X)=(1-(B/A)↑(1/C))*100
65 DEF FNDJ(X)=A*B/100*((1-B/100)↑(C-1))
70 DEF FNDK(X)=A*(1-B/100)↑C
75 DEF FNDL(X)=A*B/100*C/360
80 DEF FNDM(X)=A-FNDL(X)
85 DEF FNDN(X)=A*100*D/C*(1-1/(1+C/100/D)↑(D*B))
90 DEF FNDO(X)=(C/100*B/D)/(1-(C/100/D+1)↑(-D*A))
95 DEF FNDP(X)=-(LOG(1-B*C/100/(D*A))/LOG(1+C/100/D)/D)
200 RESTORE9100:F=1:X=15:GOSUB220:GOTO200:REM MAIN MENU HERE
210 F=0
220 FORI=1TOX:N=1:IFF=1THENCLS
230 FORJ=1TO7:IFN=0THEN250
240 READN:D$=D$+D$(N)+" "
250 NEXTJ
255 IFF=1THENOUTPUTD$,6,47,3:D$=""
260 IFF<>1THENPRINTD$:D$=""
270 IFF=0THENINPUTA(I):GOTO300
280 IFF=1THENOUTPUT"(Y)?",30,23,3:IFINSTR$(1)="Y"THENK=I:GOTO500
290 IFF=2THENRETURN
300 NEXTI
310 IFF=0THENA=A(1):B=A(2):C=A(3):D=A(4)
320 RETURN
400 X=4:GOTO420
410 X=3
420 GOSUB220:X=1:F=2:GOSUB220
430 RETURN
```

FINANCE Listing Continued

```
500 CLS: F=0
510 ONIGOTO520,530,540,550,560,570,580,590,600,610,620,630,640,650,660
520 RESTORE9500: GOSUB400:X=FNLA(X): GOTO900
530 RESTORE9510: GOSUB400:X=FNLB(X): GOTO900
540 RESTORE9520: GOSUB400:X=FNLC(X): GOTO900
550 RESTORE9530: GOSUB400:X=FNLD(X): GOTO900
560 RESTORE9540: GOSUB400:X=FNLE(X): GOTO900
570 RESTORE9550: GOSUB400:X=FNLF(X): GOTO900
580 RESTORE9560: GOSUB400:X=FNLG(X): GOTO900
590 RESTORE9570: GOSUB410:X=FNLH(X): GOTO900
600 RESTORE9580: GOSUB410:X=FNLI(X): GOTO900
610 RESTORE9590: GOSUB410:X=FNLJ(X):GOTO900
620 RESTORE9600: GOSUB410:X=FNLK(X): GOTO900
630 RESTORE9610: GOSUB410: PRINTINT(FNLL(X)*100)*.01: GOSUB220:X=FNLM(X): GO
TO900
640 RESTORE9620: GOSUB400:X=FNLN(X): GOTO900
650 RESTORE9630: GOSUB400:X=FNLO(X): GOTO900
660 RESTORE9640: GOSUB400:X=FNLP(X): GOTO900
900 PRINTFNX(X): PRINT: PRINT"ANOTHER (Y)?": IFINSTR$(I)="Y"THENI=K: GOTO500
910 GOTO200
9000 DATA A LOAN, AFTER X, AMOUNT, AN, ANNUAL, (ANNUITY), COMMERCIAL
9010 DATA COMPOUNDING, COST, DAYS, DEPOSITS, DEPRECIATION, DISCOUNT, EACH
9020 DATA EFFECTIVE, FOR, FROM, FUTURE, IN, INITIAL, INTEREST, INVESTMENT
9030 DATA INVESTMENTS, MATURITY, MINIMUM, NOMINAL, NUMBER, OF, ON, ORIGINAL
9040 DATA PAPER, PAYMENT, PAYMENTS, PER, PERIODS, PRICE, PRINCIPAL, RATE, REGULA
R
9050 DATA RESALE, SAVAGE, TERM, TO, TOTAL, VALUE, WITHDRAWAL, WITHDRAWALS, YEAR
9060 DATA YEARS, =
9100 DATA18, 45, 28, 4, 22, 0, 18, 45, 28, 39, 11, 6, 0, 39, 11, 0, 39, 47, 17, 4, 22, 0
9110 DATA20, 22, 0, 25, 22, 16, 47, 0, 26, 21, 38, 29, 23, 0, 15, 21, 38, 29, 23, 0
9120 DATA12, 38, 0, 12, 3, 0, 41, 45, 0, 13, 7, 31, 0, 37, 29, 1, 0, 39, 32, 29, 1, 0
9130 DATA42, 28, 1, 0
9500 DATA20, 22, 0, 26, 21, 38, 0, 27, 23, 8, 35, 34, 43, 0, 27, 23, 49, 0, 18, 45, 50, 0
9510 DATA3, 28, 39, 11, 0, 26, 21, 33, 0, 27, 23, 11, 34, 43, 0, 27, 23, 49, 0, 18, 45, 50, 0
9520 DATA44, 45, 2, 49, 0, 26, 21, 33, 0, 27, 23, 11, 34, 48, 0, 27, 23, 49, 0, 39, 11, 50, 0
9530 DATA20, 22, 0, 26, 21, 38, 0, 27, 23, 47, 34, 43, 0, 27, 23, 49, 0, 3, 28, 14, 46, 50, 0
9540 DATA44, 45, 2, 49, 0, 27, 23, 8, 35, 34, 48, 0, 27, 23, 49, 0, 26, 21, 33, 0, 20, 22, 50,
0
9550 DATA3, 28, 47, 3, 26, 21, 38, 0, 27, 23, 47, 34, 43, 0, 27, 23, 49, 0, 25, 22, 50, 0
9560 DATA37, 0, 44, 45, 0, 27, 23, 49, 0, 27, 23, 3, 35, 34, 43, 0, 26, 21, 33, 50, 0
9570 DATA20, 22, 0, 44, 45, 2, 49, 0, 27, 23, 49, 0, 5, 21, 33, 50, 0
9580 DATA30, 36, 0, 40, 36, 0, 49, 0, 12, 38, 50, 0
9590 DATA30, 36, 0, 12, 38, 0, 43, 0, 12, 50, 0
9600 DATA30, 36, 0, 12, 38, 0, 49, 0, 45, 50, 0
9610 DATA18, 45, 0, 13, 38, 0, 10, 43, 24, 0, 13, 50, 0, 9, 50, 0
9620 DATA39, 32, 0, 42, 19, 49, 0, 5, 21, 38, 0, 27, 23, 33, 34, 43, 0, 37, 50, 0
9630 DATA42, 19, 49, 0, 37, 0, 5, 21, 38, 0, 27, 23, 33, 34, 43, 0, 39, 32, 50, 0
9640 DATA39, 32, 0, 37, 0, 5, 21, 38, 0, 27, 23, 33, 34, 43, 0, 42, 50, 0
OK
```

Anyone accustomed to the TRS-80, Apple or a large computer is amazed at the apparent slowness of the Interact. I say apparent because the Interact is not a slow computer. The fact is that it has an agonizing slow display. The standard computer speed test of running 10 GOSUB 10 and counting the seconds until an out of memory error occurs is not really an accurate test of computer speed but more a test of the BASIC version used. The July, 1982, article, "Keeping Time", in Popular Electronics, page 79, can be used to prove Interact is faster than both the TRS-80 and Apple II. The simple timing loop FOR A = 1 TO X : NEXT where X is a large number is a combined measurement of the BASIC's speed and the clock speed of the CPU. In 30 seconds, a TRS-80 can do 3750 calculations or actually iterations, the Apple II can do 7500 (and this is with a very fast integer BASIC), and the Interact can do 15000. The Interact with Level II BASIC is twice as fast as the Apple II and four times faster than the TRS-80. It's only our screen display that is slow, not the CPU or circuits. Our version of Microsoft BASIC is fast, but to show that most of the slowness is still due to the language and not to the computer, those of you who have fig-FORTH can try this: a 30000 loop in Level II takes 60 seconds (in theory--by my stopwatch, it takes between 62 and 65 seconds but it's hard to measure). In FORTH, it takes only 6 seconds! The program is

```
: TIME-TEST  30000 0 DO LOOP  ;
```

from the August 1980 BYTE article, "What is FORTH?". Interact fig-FORTH is 20 times faster than Apple II BASIC and 40 times faster than TRS-80 BASIC. So don't let anyone tell you the Interact is slow. It may have a slow tongue but it's got a fast brain.

Richard Jones, RFD 2, Cole Camp, MO 65325

REVIEW OF DAVE SCHWAB'S SKETCH PAD AND EDU-BASIC OVERLAY

by Richard Jones, RFD 2, Cole Camp, MO 65325

I want to second Steve Cook's review and recommendation of Sketch Pad. Although not limited to making banners by any means, that is its most obvious use. I made about 12 banners in 90 minutes with pictures equal to Compute-a-color and with the advantage of omitting the stop code. Perhaps its best feature is the ease (and speed) with which it can be used. Instant triangles, circles and rectangles are made using only the joystick. Characters can be placed quickly anywhere on the screen. It should an excellent educational tool for small children as well as being entertaining to anyone who likes to draw.

The Edu-Basic overlay is a true bargain. Printer commands for the Slagh port are added along with an unlimited number of machine language calls similar to the USR command in Level II. This overlay converts Edu-Basic into a very powerful and useful language and still leaves 8191 bytes of usable program space. (It fits into about one third of the unnecessarily long array space.) The original integer BASIC for the Apple II and now FORTH on the Interact are proving that decimals are not needed for good programming. And last but not least, true PEEK and POKE functions are added to the language.

Review of the AL Language

by Dan DeLong

The AL language tape is being sold by Micro-Video as an Editor/Assembler/ Monitor for the 8080 Assembly language.

After using the Edit-x/Assembler-x, I was more than willing to replace it with something different.

With AL you can go from editing to assembling to the monitor without loading tapes back and forth. All of the programs are on one tape and all are in memory at the same time. AL uses 4k of memory. It uses from 4C00H to 4CBBH and from 6000H to 6FFFH. In my 16k system the symbol table is set up at 5B80H to 5F80H, my text area is from 7000H to 7FFFH. For those of you who can subtract in hex, this gives you only 4k of text area, with somewhat over 4k down at 4D00H and up which is empty. (More on this later.)

You can expand the symbol table, or relocate it if you want. You can do the same with the text area.

The editor portion of AL gives you all the functions that Edit-x had, plus quite a few new things. You now can move your text around as you want, you can start your List from a name in the text, you can also use the FREE command and find out how much memory you have left.

The assembler part is fantastic. you can use hexidecimal, octal, base4 or binary in your program. The BRK statement will return to the monitor so you can check out all the registers, this is great for debugging. You The DB operator has been changed so that you can define an entire string at a time, such as: INVCOM DB 'INVALID COMMAND',00H. Did you see the comma and the 00H following? You can put a whole bunch of things on the same line with the DB operator and have it all assembled correctly.

This is later. As I said before you only have a 4k text area. I didn't really expect so little, especially since AL only uses 4k. I expected to be able to use somewhere around 8k for my text. When I found this, I started doing some looking at what my source actually looked like. I found that as you type in your source, the editor converts it to some type of code. When you list your source, the editor converts the code back to mnemonics so you really don't see what has happened. What all this does is save a whole lot of memory. I figure that it saves at least half of your source in code. This means that you end up with about 8k (equivalent) for your text area. This is not bad.

In all good things there has to be a couple bad things:

1) The FREE command returns how much memory is left in your text area in HEX. This gets old real fast!!

2) The converting of your source to code by the editor sometimes messes up. I typed in KEYIN DB 07E7H. The editor converted this to KEY IN DB 07E7H. The the assembler gave me an error for having a space between KEY and IN. I fixed this by changing KEYIN to INKEY.

3) The monitor lets you write your assembled code to tape only when you specify the beginning and ending of the code. The only problem with this is that nothing tells you where your code has ended at. I fixed this by putting a DONE DB 00H at the end of my code. Now I find the DONE in the symbol table, and I now know where my assembled code ends at.

4) This last thing is just good old fashioned personal preference. I don't like errors to be printed in some code number. I would rather have the errors printed so that I could understand them without having to look them up in the manual.

Finally, in spite of the four things I listed above, AL is fantastic. It is flexible, fast and all on one tape. If you program in assembler, you will get your money's worth in AL. I love it, in spite of a few quirks.

Also, even though I downgraded Edit-x/Assemble-x at the start of this review, I still have a lot of respect for the person who wrote it. The person wrote it, which is something I don't think I could do. The person also wrote it without the benifit of an assembler, that I think deserves a lot of credit.

Dan DeLong

Review of WING IT

WING IT is an action game being sold by Micro-Video.

The object in the game is to get the butterflies from the right side of the screen to the peninsulas on the left side.

You must move your butterflies through four lanes of traffic and onto the median strip. The median strip is mowed at times, so you cannot stay there too long. From the median strip you move across lily pads, rafts and such. Finally you make it to the peninsula.

The faster you move across the screen, the more points you get. You also get points for landing on the honey box (on rafts) and flowers (on peninsulas).

The longer you play the faster the cars go, and the fewer the rafts and lily pads.

The game is quite a bit of fun. My wife and child play it so much that I have a hard time getting other things done. Another thing I like is that the game is basically non-violent. You don't get points for killing things, you do get points for keeping the butterflies alive, this I like.

Dan DeLong

MACHINE SHOP TALK

TRICK OR TREAT!!!!!

THE ENDS OF TIME

By

George A. Leggett

Since Halloween approaches us at the time of this writing I thought a most appropriate way to start although one could carry this name on all year long. Because the Machine Shop Talk I'm about to lay on you will be short and yet infinitely long. In fact, it will be the longest Machine Shop Talk I ever did for that matter that anyone ever did and for that matter that anyone will ever do and yet it will take it will take a few amount of words to do it. Our subject is simple; Infinity!

Now if that opening paragraph doesn't intrigue you nothing will. Here's what I mean. From a young boy I have always been intrigued with numbers. Math was everything to me. At 9 years old algebra was a breeze. Multiplying 3- and 4-digit numbers by 3- and 4-digit numbers was done in my head. In seconds! This is not a brag or a boast but a very happy fact. I just enjoyed working with numbers and somehow words didn't catch up till I was a lot older. I think a lot of people whether they know it or not can relate to numbers. How well they relate depends on how much they put into it. I put almost every waking moment into it. I grew up with a slide rule in my hand. Yes, for those of you younger people out there, there were no calculators. Way back ...to those of you who are older, it was just last week. Somehow I'm caught in the middle between last week and eternity. Which leads us to this Machine Shop Talk.

What is the largest number you can make your computer come up with? Well something like Blah-Blah-Blah +10 to the 37th Power. If you have a scientific calculator (I have a T.I. 59) it goes to 10 to the 99th power. A great difference as you can see. Our Interact in BASIC has that problem and for that matter so does any other BASIC because of the format which I won't go into here. But what if you wanted a really big number--how about 2 to the hundredth power or 2 to the 500th power? Thousandth power? ....Intriguing or boring depending on your point of view. The following program shall demonstrate this. If anyone out there finds an end to this program...well Amen!!! At this writing I have not worked out all the mathematics of it yet but the numbers will amaze you in that your little Interact using the screen memory as your display can make Infinity look like a microsecond.

What we're going to do is use 4000 Hex to / 4800. 1048 bytes. We multiply this by 8 bits to get 16,384 powers. Put that in your memory banks a while! A very big number. But how big is that number? Just how big is what we want to ascertain here. If you tried to print 10 to the 16,384th power, you'd get an Overflow on an Interact or any computer and calculator I know of. But if we break it down to bits where every bit was represented by a flashing light like the old 1950's computer movies where bits were represented by flashing lights.....

We're going to turn our screen into a blinking flashing 1950's style computer. Let's say, a very minute part of it. We will do it like this:

Every bit will be a color. There are four colors, thus, four colors and four 2-bit bytes represent all 256 combinations. (See reference note at end of article) Now, 1 byte has 256 numbers therefore, 2 bytes have 256 times 256 or 65,535 bits. That's only 2 bytes. We have 2048 bytes. If someone out there wishes to multiply 256 to the 2048th power let me know...I'm waiting for the answer myself. But this is exactly what the program will do. It will keep shifting and carrying to the next byte along the line. At the speed of the 8080 (ours is very fast, about 1.75 microseconds) if we could go 1.75 million counts a second, how many seconds will it take to to fill up the screen with all FF or 255? A day? A week? A month? A year? A century? Try an Eternity to an infinite power. Then you may be halfway down the screen. Yes, your little Interact holds the ages of the universe within itself.

Let's figure how long it would take to fill up the first Y row and remember coordinate 1, 1 is the upper left corner of the screen. How long would it take to fill up the 28 bytes (28 times 4 = 112 or the X axis) Let's calculate. There are 112 pixels in the first Y row from X 1 to X 112. The problem is 2 to the 112th power. The answer rounding off is 5.19 times 10 to the 33rd power. Now based on 1.75 million cycles a second how long would it take if we could address each of these every microsecond? (We cannot in this program as you will see. The program takes many machine cycles to execute.) But if it could, how long would it take? To find that answer we divide the other answer by 1.75 million. Rounding off, the answer is 2.96 times 10 to the 27th seconds. But what does that mean in terms of real time? What you do is divide that number by 60 seconds in a minute, 60 minutes in an hour, b6 24 hours in a day, 365.25 in a year, (this accounts for the leap year) and you get 9.40 times 10 to the 19th years or Forty Million Million Million years!!! And that's only for the first Y row of 112 pixels on the X axis! We have 76 more!!!!!..... Actually we can go to bed a little early as it were since we're only going up to 4800 Hex which is only about 61 Y down the screen.

Now: The challenge to someone out there and good luck, is to tell me the exponent number of years it would take to fill up the screen all the way down the Y rows of the screen to 4800 on the screen. Pray tell the number of years, centuries, millenia, mega-bega-bega-millenia..... And this is only the screen. How about filling up imagine filling up the memory in your 16K computer. If any of you solve this via a program as I'm sure it would take a complex program to solve this, or maybe a simple one, it's all a point of view, please submit it. I would be very interested and I'm sure many other people would be too. See what we come up with. I mean, it isn't every day you're challenged to solve Infinity! Or is it?

I'd like to thank at this time the inspiration behind a story like this the book One Two Three Infinity and the soundtrack album to the series COSMOS (See Reference note). I played the record during the making of this transcription to get me into the mood of the Infinite. They make you think. As powerful and as fast as the computers are, they are mere infants in the world of tomorrow.

George A. Leggett

THE ENDS OF TIME LISTING

Address	Hex	Mnemonic	Description
5800	CD	CALL	Call Clear Screen Routine in ROM 1 at 0573
5801	73		The purpose is to erase the memory and put
5802	05		zeros in the memory.
5803	21	LXI H	Load H and L with the memory location of the
5804	00		upp er left corner of the screen. You may
5805	40		move this to about 4208 to see results easily.
5806	34	INR M	Increase memory of HL by 1 Store result on screen.
5807	C2	JNZ	Do the Increase again until it reaches zero.
5808	06		
5809	58		
580A	23	INX H	This moves H and L to act like a carriage left to right
580B	34	INR M	Increases next location by 1.
580C	C2	JNZ	Increase again until it reaches 0 then proceed
580D	03		on to next step at 580F
580E	58		
580F	C3	JMP	
5810	0A		
5811	58		Increase to next location and continue as before.

You may want to enter a few extra commands before this routine. I prefer a black screen for this exercise and you may wish to play around with the H and L value of 5803 to change the location of it. You can see after a very short time that even though the first byte and the second byte move at an enormous amount of speed each byte afer that is 1/256 slower than its predecessor. the next byte after that will be 256 squared, then 256 cubed then 256 to the fourth power and so on. Needless to say each is slower than the one before! The earlier discussion of how long it would take to fill up the screen was based on everytinng happening in 1 Machine Cycle. Obviously it takes much more than that. The 5806 to 5809 loop which simply increases and jumps back to do it again takes 20 machine cycles. The loop from 580A to 580E which jumps back to 5803 takes 35 machine cycles. Of course this is multiplied by the number of times it is in the loop, whether it is carrying the byte or increasing memory locations. To help you work out the fine details, I'll lay on a few numbers for you.

21 LXI H 10 machine cycles. 34 INR M 10 cles. Briefly, every command in the routine takes 10 machine cycles excluding 580A which is a 23 and takes 5 machine cycles. The machine runs at approx. 1.75 Megahertz A machine cycle takes 500 nanoseconds (NS). Two cycles take 1 microsecond. In 1 second your machine may execute three and a hal million machine cycles. An enormous number! But compared with the ends of time it is bothing but a horse and wagon on a road that never ends and for that matter never begins.

Sincerely,
George A. Leggett

The Hard Facts Of Life

by
George A. Leggett 20562 Woodward Mt. Clemens, Mi. 48043

<=><=><=><=><=><=><=><=><=><=><=><=><=><=><=><=><=><=><=><=><=><=><=>

The Hard Facts Go Soft
Adapting Level II Basic for all your hadware needs

This Hard Facts deals with the software that makes the hardware go. Level II basic or Fast Graphic Basic which are one and the same in my book except for a few bytes can easily be adapted to run on 32K or 48K memory systems with only 16 Pokes!! Also I shall show you how to use the Micro Video Port without RS-232 Basic with only a short Basic Program of a few hundred bytes!

I may sound like I am attacking Micro Video by this opening but I am not let me explain. First the memory. Those of you who bought 32K expansions from Micro Video have no doubt bought 32K Basic by now so I felt that I am certainly no threat to that market. The other part of you that have built your own 48K systems have either purchase the 32K Basic or have your own adaptation or have ask me for the answere.

So why adapt Level II Basic anyway? Because there is numerous documentation on it. I know most of you who have written me and returned the renewl form don't agree there is enough but compared with 32K BASIC which has zilch on it Level II is a walking encyclopedia of knowledge. For the rest of you out there (vast majority) this is also for you if your thinking of upgrading (I cann't believe so many have not yet).

What does 32K BASIC have in it? It has all of Level II commands plus the extended plot of Fast Graphics BASIC plus the RS-232 BASIC plus a line command and a different USR function. It may sound like a lot but it really isn't compared with the memory you have. In fact as many of you know I wrote my own language called Super Language which is not availible (sorry) but has all of the above plus a 8080 monitor with 16 commands plus 45 new basic words plus 1K of user routines (eg. circles, timers ect.) and the RS-232 transmit and receive functions and theres still 1K of free machine language left plus 16K or 32K of user basic! I say all this merely to point out the power you have with the old language and don't settle for less. You will Find that the better you know Level II BASIC and its' memory maps the more you can do. Example for you few Forth people out there you are saying that you like Forth because it is faster well there are many ways that I know right now to speed up BASIC so it can run much faster. Examples: Print 40 times faster it can Poke a number 120 times faster than before and as one final example it can even be told to go to a subroutine with a name. Example the soubroutine may be a deck of cards you can tell it gosub cards. So you see Forth has nothing on me. Of course the langauge I wrote may have begun from Level II but is now completely different but the point is Level II gave me an excellent understanding of BASIC from the inside. This is why I do not bother with 32K BASIC at all. You may say that Level II does not have the line command well this is true but are you not forgetting the CMD Overlay in Vol. IV no. 3 which gives you all that and more.

Before we go on I must tell you that if you use the Level II BASIC along with the CMD overlay and the RS-232 program listed here it should

satisfy all you needs for now. But remember to get all the neat commands mentioned in Super Language took me 3 years of study with 8080 and BASIC writing over 2 million bytes of Programs till I knew what commands I needed and over 1200 hours to develop the Language itself. As they say anything worth having in life does not come easy it takes work and in this case a lot of it.

The following should be enter in this sequence. Press the cr key after each entry. The only difference in 32K and 48K is three bytes moving the memory limit up. The Pokes move the start of BASIC Program area, variables, strings, and arrays.

"Changing Level II BASIC to 32K or 48K"

1. Load Level II or Fast GraPhics BASIC
2. Type NEW
3. Poke19215,25 Level II BASIC only
4. Enter the following Pokes

32K BASIC	48K BASIC
POKE 19542,204	
POKE 19543,191	POKE 19543,255
POKE 19546,1	
POKE 19547,128	
POKE 19670,254	
POKE 19671,191	POKE 19671,255
POKE 19690,254	
POKE 19691,191	POKE 19691,255
POKE 19709,3	
POKE 19710,128	
POKE 19711,3	
POKE 19712,128	
POKE 19713,3	
POKE 19714,128	
POKE 19715,0	
POKE 19716,128	

5. Type NEW
6. You now have 16K or 32K free for Programming depending on which BASIC you used 32K or 48K. Everything will work as before you will just have more of it.

I hope this will be of use to you. All you have to do now is load in the CMD Overlay and save the whole thing on tape for one load. You can do this by using the standard write table from any monitor but you will have to move the monitor so you do not write over basic or use my write routine in the 8080 Programmer Pack. Remember addresses 4D22 hex to 5F80 are now free to use for machine language routines or to expand your BASIC language.

"RS-232 BASIC from BASIC"

I wrote this Program in March of this year before my Micro Video Port broke down. I do not have a Printer for that Port but I do have a Type 'N' Talk and this is what I used to test my Program. To my

surprise it worked great. This program is not to eliminate RS-232 BASIC or other overlays as you will still need them to list your program but you won't need them in your Program when you want Graphics and RS-232 usage. There is some delay with the higher baud rates as they work faster then BASIC but 600 baud and lower as I recall work fine. I think the Program that follows is set at 1200 baud. I arrive at this idea from looking at BASICALLY SPEAKING and the sheets that come with the Port and notice that if you could change all the Parameters with simple Pokes why not go all the way. Well you can. Your notice all the Peeks and Pokes are negitive this is because the numbers are above 32767. The Port sits at C000 hex which is 49152. Refer to Vol. IV no. 1 for negitive Peeks and Pokes and how to use them in Machine Shop Talk The Happy Marriage.

Line 5 may be omitted if you are using Fast Graphics BASIC. Lines 10 to 50 initalize the Port eg. baud rate, Pairity, markers, word length ect. Line 110 inputs a string to be transmitted to variable XT$. If you were going to use this in a Program you would omit this and just begin at line 120 with your string in XT$. Line 120 to 150 issolate each letter from the string left to right and store its ASCII value in variable A. Note by changing this you can output your string from right to left thus making your Printer type backwards. Line160 Puts a carriage return after the sting. You may want to change this. Line 999 sends you back to line 100 which clears the screen and starts the whole thing over again. Now we get to the subroutine at line 1000 which is simple. Line 1000 sends the ASCII to the Port to transmit. Line1010 waits around till the letter is gone and line 1020 returns from the routine.

That's about the size of it short and simple. If I get enough calls or letters I will go into more depth and even receiving data in BASIC.

A FOND FAREWELL

This concludes this series of the Hard Facts of Life. I know that most of you still want to see it in 1984 but I simply don't have the material or time. A single Project can take 2 to 8 weeks to fully develop. I will always be working on hardware it is in my blood, however my hardware intrest have gone way beyond modems and Printers and memory all of that has been done and there is no challenge to me once it has been done. Maybe it is the inventor in me. My intrest now deal with speech and video recognition, artifical intelligence and robotics. It is my opPion the Interact can no longer fill that need for two reasons.

1. The difficulty in interfacing. No external edge conectors.
2. Cost. If you blow the unit up for some unknown reason it would be very expensive and time consuming to say the least. I am using a VIC-20 for my basic interfacing and then once the circuit works I can easily use it on my Commodore 64. If I wipe out a VIC-20 I am only out $60.oo If I wipe out an Interact there goes my life savings.

If I receive any material for HARD FACTS of LIFE I will Publish it.

Sincerly,

George A. Leggett

RS-232 BASIC PROGRAM LISTING

```
1 REM RS-232 TRANSMITER IN LEVEL II BASIC
2 REM GEORGE A. LEGGETT
3 REM MARCH 21,1983
5 POKE 19215,25
10 POKE -16381,131
20 POKE -16384,88
30 POKE -16383,0
40 POKE -16380,0
50 POKE -16381,PEEK(-16381)AND127
100 CLS
110 INPUT"ENTER YOUR        TRANSMISSION";XT$
120 L=LEN(XT$)
130 A=ASC(LEFT$(XT$,1)):GOSUB1000
135 IFL=2GOTO150
137 IFL=1GOTO160
140 FORX=2TOL-1:A=ASC(MID$(XT$,X,1)):GOSUB1000:NEXT
150 A=ASC(RIGHT$(XT$,1)):GOSUB1000
160 A=13:GOSUB1000
999 GOTO100
1000 POKE-16384,A
1010 IFPEEK(-16378)AND16=0GOTO1010
1020 RETURN
```

Note this program is not included on your tape.

ENDS OF TIME REFERENCES

1. Interactions Vol. 1 no. 3 page 7 Interactory Factotory
   This artical aplies to characters but the same binary techniques can be used.

2. BASICALLY SPEAKING page 3-16 Advance Graphics -- POKEing the screen
   Excellent example of how bits are mapped and used on the screen.

3. ONE, TWO, THREE... INFINITY by George Gamow Facts and speculations of Science If you love numbers and mind trips this book is a must.

4. COSMOS sound track from the series COSMOS by Dr. Carl Sagan Great background music for getting into this kind of writing.

INTERACTION INTERNATIONAL
GEORGE A. LEGGETT
20562 Woodward
Mt. Clemens, MI 48043

ROYAL OAK, MI 480
DEC 6
PM
1983