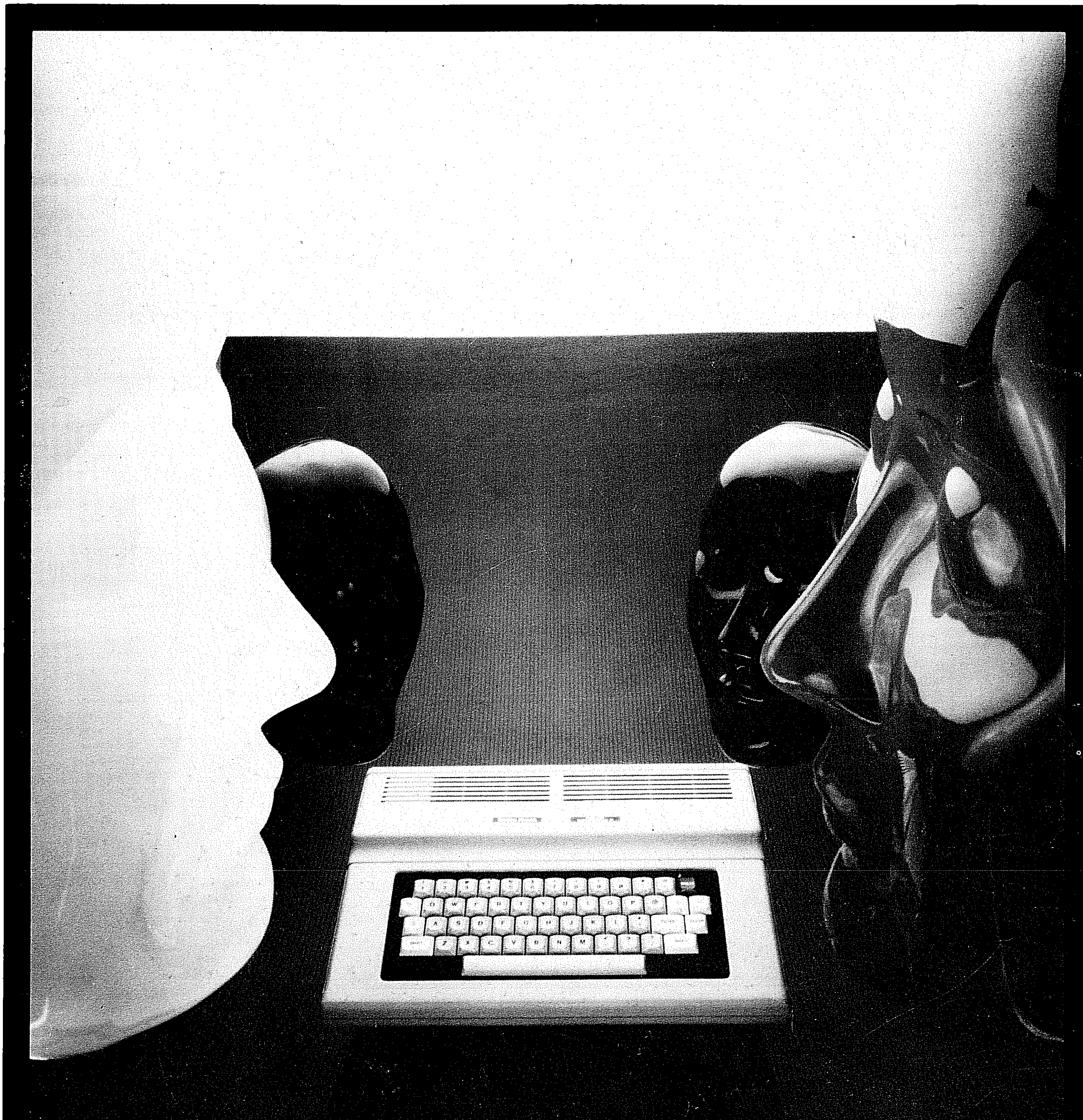
UNDERCOLOR

Vol. 1 No. 1 **December 10, 1984**



000642-120885-10000002
L. Clark
[redacted]
Paragould AR 72450

Table of Contents

Cover Photo by Charles Freiberg

Who's Who

Editor
Debra A. Marshall

Technical
Dennis Bathory Kitsz

Contributing Photographer
Charley Freiberg

Production Manager
Clare McCarthy

Paste-Up
Janet Patterson

God-Send
Joanne!!

Advertising:
Judy Knapp
(802)485-6139

Advertising:
Spencer Knowlton
(207)785-4151

Office
(802)485-6440

Editorial Office
(207)785-5148

Published By: ColorPlus Inc., Box 6809, Roxbury, VT 05669. (802)485-6440.

Circulation Problems? Contact David Gunn, Box 6809, Roxbury, VT 05669, 802-485-6440.

Subscriptions: Until Dec. 1984: $20. After Dec. $33. Add $14 for first-class postage. Foreign: until Dec. 1984: Canada/Mexico $30 U.S.; other countries $40 U.S. *Under Color* is published 26 times a year. Call 802-485-6440, or write to *Under Color*, Box 6809, Roxbury, VT 05669.

Advertisers: rates available. **Editorial Office:** c/o Debra Marshall, RFD 1, Box 786, Union, ME 04862 207-785-5148. Submissions are welcome; include S.A.S.E. for returns. To receive author's guide, write: Author's Guide, Box 6809, Roxbury, VT 05669. Letters to Editor, manuscripts, article queries, etc. should be sent to editorial office.

Defuser by Terry Kepner

I want to interface my printer, an Epson MX-80 F/T with GraftraxPlus, to my late model TRS-80 64K Color Computer II, at the computer's serial I/O port. I also want to maintain this printer in-line with my 32K Model I (no expansion interface).

I have written to five different suppliers of serial-to-parallel printer connectors and have received as many different offers for interfacing the printer, but none would assure me that the printer can be effectively interchanged between the two computers.

The Epson printer is modified with the Epson TRS-80 interface board (no. 8120) and the Epson Interface Cable Connector (no. 8221, #825), connected to the Model I at the left rear of the keyboard. The new 64K Color Computer is interfaced to a Color Computer II disk drive zero. I also have a R/S Parallel LPII Interface (no. 1416) with cable.

I hope you can suggest a means to use both systems with the one printer.

Julian Jatyna
46 Old South Rd.
South Berwick, ME 03908

The standard Epson printer uses a parallel port, so all you need to use it on your Color Computer is a serial-to-parallel converter. However, since you're not using an Expansion Interface with your Model I (which normally contains the Model I's parallel port), you're using the Epson 50-pin-to-parallel interface board. If the interface board is plugged into the Epson, inside the case, you won't be able to use the Epson with your Color Computer unless you remove the board and use the normal parallel Epson printer port.

If you have an Expansion Interface with a parallel printer port, you could just get some type of hardware switch to select which computer was plugged into your printer, with all three devices plugged into the switch. In one position the switch connects computer A with printer, in the other position computer B is connected. You can't just plug both computers to the same printer cable without risking damage to all three devices; only one printer at a time can be electrically connected to a computer, and vice versa.

So, if the Epson/Model I interface board is outside the Epson, you can put the hardware switch between the interface board and the printer, letting you switch between the two computers (assuming you have a serial-to-parallel interface on the Color Computer).

If the interface board is inside the Epson, or the Epson doesn't have a parallel port, then you can't use the Epson with the Color Computer.

I have a 32K Extended Basic Color Computer. Could you tell me what hardware is necessary to use the computer to make contact with my office mainframe? Are there statistical packages available which are comparable to SPSS, BMD, or similar packages?

Bonnie MacLean
280 E. Western Reserve Rd.
Poland, OH 44514

You need two things: a modem with a Color Computer RS-232 cable, to connect the Color Computer to your telephone; and some type of telecommunications program for the Color Computer, so what you type goes to the phone lines, and what the mainframe sends is displayed on your monitor. For this to work, your office mainframe must have an auto-answer modem, so it can answer the phone when you call from home, plus the necessary software for the mainframe to accept instructions from the phone line.

Since I don't know the features of SPSS and BMD, I can't make any suggestions as to the acceptability of Color Computer statistical packages. Can someone else help?

Reading the November, 1983 DEFUSR column in The Color Computer Magazine, *I noticed a letter asking about 220 volt 50-cycle power and the Color Computer. I hope my experience can help. I'm stationed with the USAF in Europe and have been using my 16K revision D Color Computer, Line Printer VII, Radio Shack cassette, and Radio Shack disk drive for approximately one year here. In all this time my system has been running on 120 volt 50-cycles stepped down from 220 volt 50-cycles. I have experienced no problems using a standard step-down transformer. The 50 cycles has not changed anything noticeable. I have used my Color Computer for eight to ten hours at a time with no problems. I did have to make one very simple adjustment on the disk drive: setting the drive to 50Hz operation. The Radio Shack disk drive comes with 50/60-cycle hash marks on the drive wheel; after removing the cover, adjust the motor potentiometer until the 50-cycle marks appear to stand still when viewed under a fluorescent lamp (this can be done without voiding your Radio Shack warranty). The adjustment potentiometer is located next to the drive wheel. Without this adjustment, the drive won't recognize the DSKINIO command.*

In some situations a change from 60-cycle to 50-cycle power can cause a heating problem for the Color Computer power transformer. This is normal and over a long period of time could cause power supply problems. I haven't experienced any problems, but my computer system is located in a fairly stable temperature environment. If the area for the computer were very warm in the summer, I would recommend taking along a cooling fan.

One last note: I have just purchased the new Radio Shack 64K Color Computer 2, the Color Computer 2 disk drive, and the Radio Shack version of OS-9. I have everything except the disk drive, and again have no power problems. Hope this information has been of help.

T.A. Ward
Box 1211
APO NY 09453

Thanks for giving us an account of your experiences overseas; my readers will appreciate it.

I read an article about an Apple computer interfaced to a laser disk player. Can the Color Computer

(64K) be interfaced to a laser disk player? If so, can I use my computer to write a machine code routine to control the laser disk? Also, do the companies that sell 64K programs use the full 64K?

Wayne Tracer
Star Rt. Box 1201
Chiloquin, OR 97624

The Apple/laser disk combination requires a special hardware board plugged into the Apple mother board, with a cable running to the laser disk. This board translates instructions from the 6502 CPU into commands that can be executed by the laser disk controller. To the best of my knowledge, no one has as yet developed a hardware interface for the Color Computer to use a laser disk, although it shouldn't be all that hard since the Color Computer ROMpak port makes all the control and data lines available.

We'll probably see this hardware configuration on the market soon, especially since the new laser disk memory units are moving out of the laboratory and into the market.

All the companies that make 64K programs require that your computer have 64K RAM before the programs will work. Most of them use the entire 64K (less system RAM requirements) for their operation. Programs that specify 32/64K usually use only 32K, but may have 64K options that only run when the extra memory is available; these programs automatically test for 64K memory.

I won a 64K Color Computer, and I recently purchased a Modem II and the Deluxe RS-232 Communications ROMpak. I read the Modem II technical manual to discover if it is capable of auto-dialing with tones or pulse. Unfortunately, the Deluxe ROMpak programs the modem for fast-pulse rotary dialing only. This is OK for local calls, but on MCI and Sprint I'm unable to use the modem's auto-dial function; these services require tones. Is there any way to get the tone option programmed into the modem instead of the pulse? And do you know of any BBSs in my area?

Michael S. Parker
7147 Old King Rd. S. #22
Jacksonville, FL 32217

Sorry, but the only way I know of to change the modem to tone dialing is to reprogram the communications ROMpak to use tone instead of pulse, which would require disassembling the ROM, finding the offending code, removing the ROMpak ROM, and replacing it with a new one containing the fix; none of which is easy.

Does anyone have a possible solution, or know where M.P. can get a "fixed" Communications ROM?

Here's a short list of Color Computer BBSs in Florida: Colorama of St. Petersburg, FL, 813-345-8100; Colorama of Dunedin, FL, 813-733-2415; Ft. Lauderdale, FL, 305-525-1192; Lantana, FL, 305-533-0333; and Hialeah, FL, 305-681-8490.

I recently bought a 16K to 32K upgrade kit. The advertisement says the kit has easy instructions, but I found them hard to follow. I also had to buy extra tools to install the kit.

Please tell me how to get the memory chips off the board.

Marc Jenkins
713 Vernon Rd.
Philadelphia, PA 19119

Removing memory chips from IC sockets is simple in concept, but requires a bit of dexterity. Use a small screwdriver, or a small nail-file, to gently pry up each chip. Or you can use a commercial IC puller, which looks like an oversized pair of tweezers with the tips bent towards each other. The ends fit under the chip, and you pull the chip straight out of the socket.

When using a screwdriver to pry up a chip, be careful not to bend any of the IC's legs. If you do, carefully straighten the bent leg. If you break it off, you'll have to replace the chip. Also, memory chips are static sensitive, so always handle them by their bodies, not the legs, and set them down on black foam or a sheet of aluminum foil. Good luck.

I'm interested in exploring the world of BBS. I own a 64K Extended Color Basic computer. At the moment I don't have a Color Computer modem (it's second on my wish list, after disk drive zero). I do have, however, an IBM 8775 terminal and radial-Vac modem (Model VA2450) that I use, on loan from my employer, to tie into a mainframe at work. Are there any BBSs I can access with this terminal? If so, are there any BBSs in the south New Jersey area?

My next question is: can this modem be used with my Color Computer? If so, what type of cables or adapters would I need?

Samuel Murphy, Jr.
9 Larchmont Dr.
Burlington, NJ 08016

I don't have access to either piece of equipment, so I can only answer in general terms: unless the IBM terminal uses special protocols, you should be able to use it with any BBS. Normal BBS communications use 300 baud, seven bit words, one stop bit, even parity, and XON/XOFF protocol. If you can't set these parameters on your IBM terminal, then it can't be used for normal BBS work. If the modem is a standard RS-232 input, 300 or 1200 baud unit, then you should be able to use it with your Color Computer using the Radio Shack Color Computer RS-232 cable (which is designed to plug into an RS-232 modem). If the modem doesn't have a DB-25 RS-232 connector on it, you'll have to come up with your own Color Computer-to-modem cable layout.

Has anyone successfully used a Radial-Vac modem with the Color Computer?

Here are two New Jersey BBS numbers (both are 300 baud numbers): Colorama of Highland Park, 201-572-0617; and Meadowland CoConet, 201-773-8265. *(end)*

Box 6809

Here's a few notes from readers:

Ideas:

*How do you program an EPROM to start on Reset like Basic?

*A weather station using A-0, temperature sensors, wind speed, etc.

*How to program EPROMs using DEFT Systems Pascal

*How to get OS-9 and Basic 09 to talk to the outside world

*How to write device descriptors for OS-9
—Frank Dupont, Detroit, MI

Comments:

*At first I thought you [Dennis] were quitting your job with *TCCM*. That didn't seem right. Then I thought perhaps you were fired, but that didn't seem right either . . .—Dave Shewchun

*Custom Color *has* to continue, for many good things to sink my teeth in are still under blankets . . . —David Nicol

*Nice to hear that a new CoCo publication is starting up . . .—Allan Ashton

*[Referring to this publication] Please accept this letter as my vote in favor of such a project!—Jon Morse

*[Referring to Custom Color] I have always enjoyed your column (even though I usually didn't understand it) . . .—Jock Kraft

*. . . you've given us our magazine back—only better!—Dennis Peterson

Info:

*I own one of the last "battleship gray" CoCo's—one of the group with the "A" on the end of the serial number. It started life as a 16K Standard Color Basic machine, but is now adjusting to its new career with Extended Basic. I eventually hope to have 64K, at least one disk drive, and perhaps a switchable color monitor driver. I say switchable because I would like to keep the TV interface around—it's convenient to have. I also need a printer.

You've probably got the idea I'm a fan of the CoCo, and so I am. However, I do have a few gripes with the machine. The new keyboard took care of one problem, but that display . . . . If only Tandy hadn't slavishly devoted themselves to Motorola IC s. The chips are fine for hi-res, and I don't even mind the "checkerboard" lowercase, but 32 columns is just a bit limiting. Oh, well, I "grew up" using an Apple II+ with only 40 columns and not even an attempt at lowercase, so I can adjust. Besides, trying to fit useful information into a 32 × 16 screen nicely is a challenge, and I enjoy a good challenge.

Speaking of Apple, I don't know whether you're familiar with those machines, but a CoCo with 64K, disk, and EDTASM+ ROMpak is comparable to or maybe even a bit more powerful than an Apple II+ was. Even the assembly languages are similar. The major differences are as follows:

Color Computer	**Apple II+**
• 32 column display	• 40 column display
• up to 4 DD disk drives	• up to 4 SD disk drives
• ROM monitor/assembler available	• ROM monitor built in, assembler available
• TV video output	• monitor video output
• 256×192 hi-res graphics, 9 distinct colors	• 280×192 hi-res graphics, 6 distinct colors
• 64×32 lo-res graphics, 9 distinct colors, free mixing of text	• 40×48 lo-res graphics, 15 distinct colors, limited text
• built-in RS-232 port, 1 expansion (ROM) slot	• 8 expansion slots
• limited but expanding software base	• very broad software base

The Apple has a slight edge in the video department, but each machine has its own advantages in this area. Overall, the two computers are roughly identical in power. And, of course, the price difference is rather profound.

Certain people have tried to compare the CoCo to the new IBM PC, but I think that IBM is just a little out of the CoCo's league in terms of software availability and ease of expandability—David M. Botkin

DENNIS KITSZ

A Real-Time Talking Clock

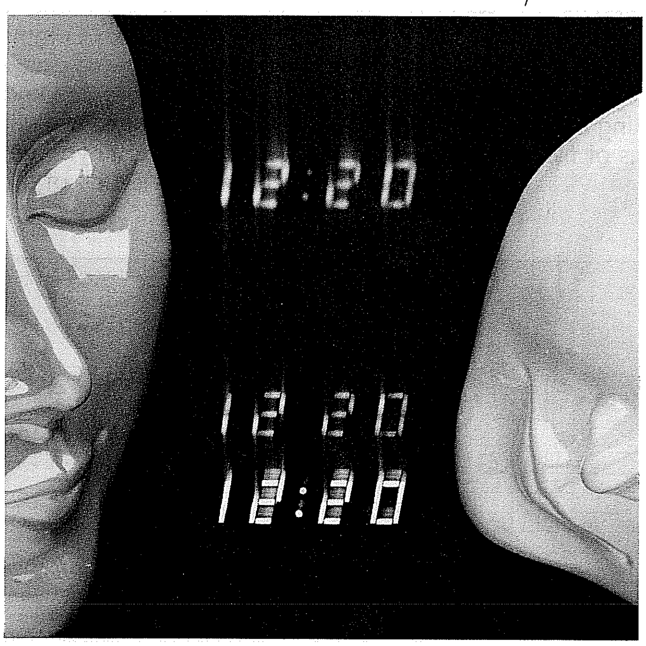
Part I

A real-time clock with voice synthesis!

What would you like? A real-time clock? A voice synthesizer? Info on using joysticks? Software without added hardware? A couple of hardware projects to dig into? Okay, get ready: I've got a something-for-everyone series beginning this month that includes two ways of producing voice from the Color Computer and two ways of adding a real-time clock. Depending on how you fold them together, you can have Basic programs that speak or a clock that talks.

This month I'll be describing how to do voice synthesis — actually, voice "recording" — in software, using your joystick input and cassette output ports. In the process, you will learn how the joystick input works, what sampling and timing mean, how the sound output works, and how you can force the high-speed POKE to work in RAM.

After you've had a chance to play with the vocal input/output, I'll create a purely software real-time clock. Since real time anything has to be totally independent of the vagaries of software, you'll learn about interrupts and video synchronization.



In the second part of this series I'll discuss interfacing and using the General Instrument vocal tract synthesizer (sold inexpensively by Radio Shack), and feeding it through the cartridge SND input. That will open the door to combining the vocal hardware with the interrupt-based real-time clock to create a talking timekeeper.

Finally, part three will offer an allophone (speech sound) vocal tract synthesizer that will give you an unlimited vocabulary. I'll provide it with inflection, too, and throw in one more twist to the plot — a battery-powered real-time clock. That will inevitably lead to a rewarding combination: a talking clock that keeps time even when the computer is off. It, too, will be interrupt-driven and completely transparent to other computer operations.

DIGITAL RECORDING. The basic principles of digital recording were explained in my article on Color Quaver (*TCCM*, November and December, 1983). Here's a review.

Sound is transmitted by variations in air pressure. Pressure variations are transformed by microphones into proportional electrical voltages. In analog recording (such as a phonograph record) the voltages are stored as wiggles in a record groove; in tape recording, the voltages become variations in magnetic intensity. By reversing the process and forcing voltages through a loudspeaker, air pressure variations can be reproduced; sound results. At all of its stages, analog recording "looks" like the original sound in some respect.

Digital recording also transforms the air pressure changes into electrical variations, at least to start. From that point, though, the process differs dramatically. The electrical variations are measured at regular intervals, and the voltage that has been measured is stored in computer memory, or on disk or tape. To play the sound, the stored numbers are converted back to voltages at exactly the same rate that they were originally measured.

The process of measuring the changes in incoming voltage is called sampling, and the frequency of this process is called the sampling rate. The accuracy of the measurement is called its resolution. *Sampling rate* is expressed in kilohertz (thousands of cycles per second, abbreviated KHz), and *resolution* is provided by the number of bits used to sample and store the incoming voltage. The sampling rate affects the fidelity of the sound (faster sampling means a wider frequency range), and the resolution affects the level of sound intensity and noise (higher resolution means wider dynamic range and a better signal-to-noise ratio).

In the demonstration I've got for you, the sampling rate is about 3.6 KHz (7.2 KHz on a 64K computer using the high-speed RAM mode), and the resolution is six bits. The resulting frequency range ends at 1.8 KHz (3.6 KHz on a high-speed 64K computer), and the signal-to-noise ratio is a poor but quite intelligible 36 dB. High fidelity it ain't. Again, you'll find details in the articles on Color Quaver.

The Color Computer has all the means to record and play back voice or music. You won't get much of it — 4.5 seconds is as long as it will last in 16K of memory — but it will start you on adding voice to your Basic programs or doing experiments with digital sound recording.

THE JOYSTICK. The key to digital sampling is joystick input. Unlike simple Atari-style joysticks, which are made up of just four switches, Color Computer joysticks are variable resistors like stereo volume controls. A voltage (actually +5 volts) is placed on one side of the resistor, and ground is hooked to the other side. The variable resistor's "wiper" sweeps from the ground side to the +5 volt side as you move the joystick, producing a variable voltage at the wiper contact. If the distance from ground to the wiper is labeled A and the distance from +5 volts to the wiper is called B, then the voltage at the wiper is 5*(A/A+B).

The Basic command PRINT JOYSTK(0) causes the computer to sample the voltage at the joystick wiper. The number displayed is 0 to 63, representing a range from $0/64$ths of five volts (zero volts) to $63/64$ths of five volts (4.92 volts). Try it; enter these lines:

```
10 CLS : A$="#.## VOLTS"
20 A=JOYSTK(0)
30 PRINT@269,USINGA$;5*A/64;
40 GOTO20
```

As you move the joystick, the voltage being sampled (within ½ bit, or $1/128$ volt) will be displayed.

The significant question is how the joysticks work. The voltage values don't just jump into the computer. And if you're looking for a sophisticated analog-to-digital converter chip, you won't find it.

The answer lies in Radio Shack's desire for a low-cost solution, and their cleverness in finding it. Converting a number to a voltage turns out to be an easier task than going the other way 'round. Converting a number to a proportional voltage goes this way: The binary value is fed through a group of resistors; each resistor contributes twice the previous amount of voltage (each bit to the right is twice the value of its neighbor to the left). Binary arithmetic prevails (I'm rushing here), and the resulting voltage ends up proportional to the original binary number. Figure 1 is the schematic of the digital-to-analog output stage of the Color Computer (the Color Computer 2 uses a custom integrated circuit for this).

If you can output a known voltage cheaply and easily, then you can measure any unknown voltage by comparing the two until you get a match. The known voltage feeds into one side of a commonplace electronic device

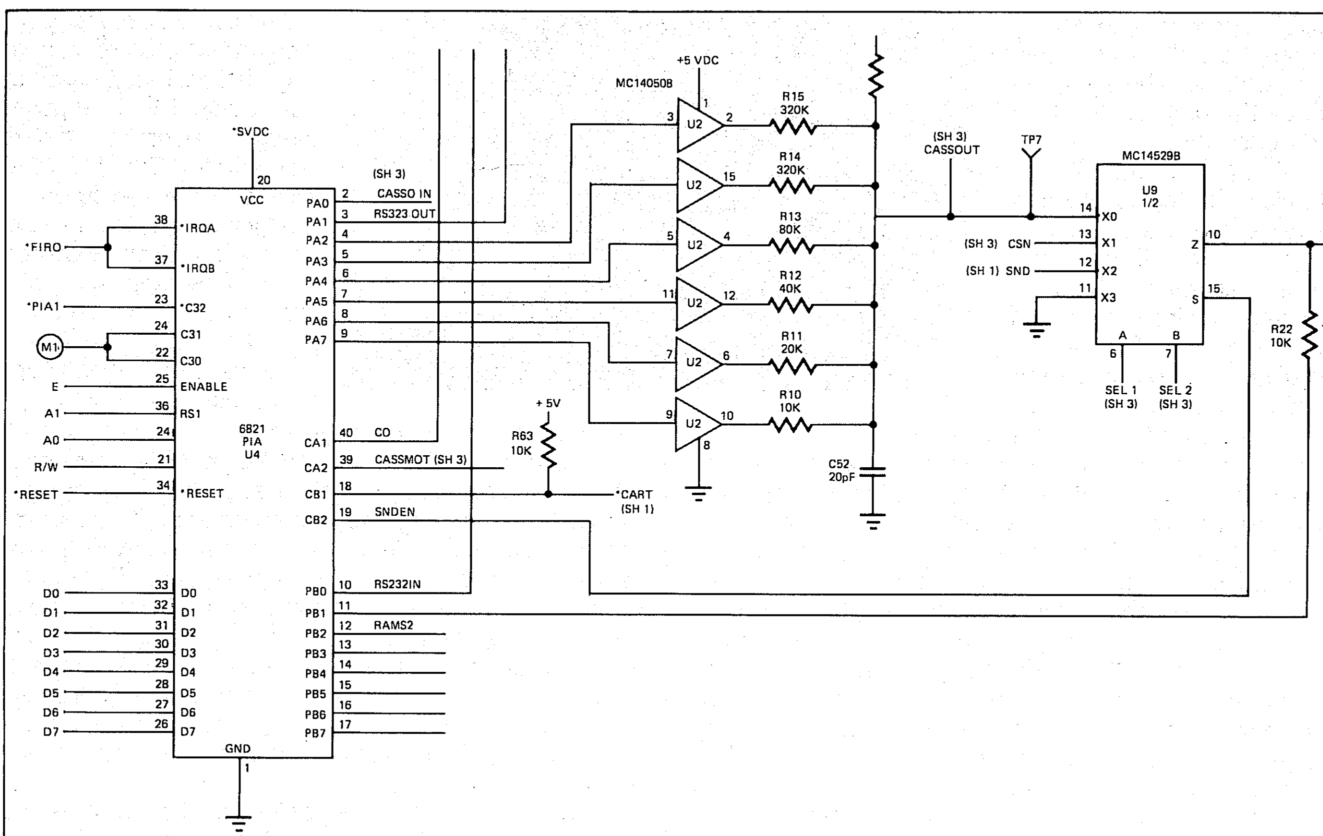


Figure 1. Digital-to-analog output stage. Reprinted with permission from the *TRS-80™ Color Computer Technical Reference Manual*, page 66.

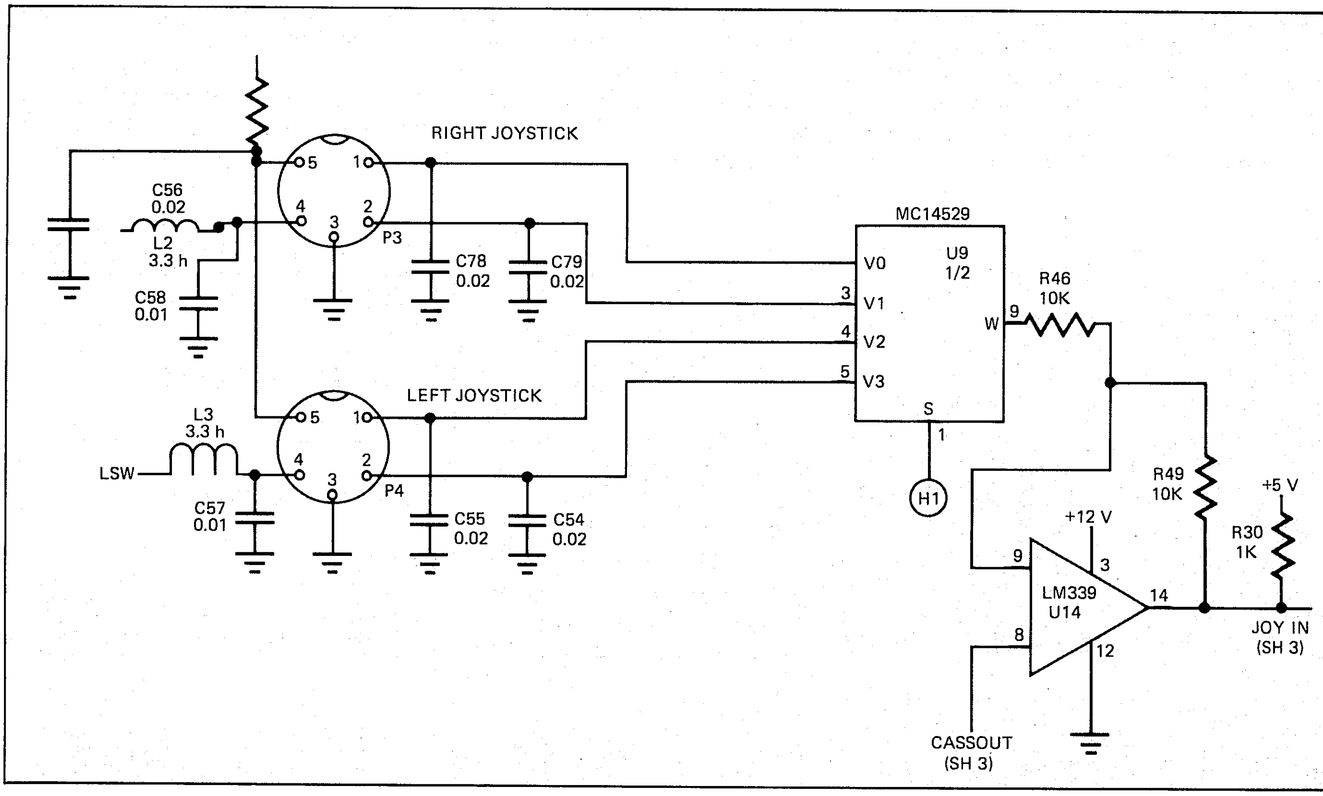


Figure 2. Joystick input stage. Reprinted with permission from the *TRS-80™ Color Computer Technical Reference Manual*, page 67.

known as a *comparator*; the unknown voltage feeds into the other side. The comparator flips from 0 to 1 when the unknown input equals or exceeds the known input. By reading the state of the comparator, the precise moment of match can be discovered. Figure 2 is the schematic of the joystick input section of the Color Computer.

Look at Listing 1, a Basic version of what Color Basic does when it checks the joystick. The digital-to-analog converter is found at address $FF20; the comparator bit is found at $FF00. The technique shown here is a *binary search*, also called *successive approximation*.

Control information is sent to the computer's ports (Lines 5 and 6) to get things set up and ready to read a joystick input. Variable A starts at the midway point (Line 7) and its value is sent to the binary-to-voltage converter (Line 10). Variable B receives the information from the comparator byte; bit 7 is the actual comparator result, and bits 0 through 6 contain other computer information. By ANDing value B with $80 (binary 10000000, Line 12), all but bit 7 are masked out, leaving the comparator information alone. Figure 3 shows that bit going into a 6821 peripheral interface adaptor, marked JOYIN.

If the comparator shows that the known outgoing value is less than the unknown incoming value, the binary search continues with a higher outgoing test value (Lines 14 – 16). Otherwise, it uses a lesser outgoing test value (Lines 17 – 18). The counter is decremented (Line 19), and the loop repeats until the process homes in on the unknown value. The result is divided by four to put it in the range 0 to 63, and then displayed.

Run the program. You'll see that — without ever using a JOYSTK command — the joystick value is printed correctly on the screen. Granted, the process is slow in Basic, but it demonstrates that analog-to-digital conversion is not mysterious.

If it works in Basic, it zips along in machine code. Look at Listing 2. The same values are sent to locations $FF01 and $FF03 to perform the set-up. The values 6 and $80 are stored in memory locations TEMP1 and TEMP2 as variable information. The value in the A accumulator goes to location DAC ($FF20).

The process varies just slightly, since a rotate command is available in 6809 commands, but not in Basic. Bit 7 of ADC ($FF00) is rotated into the carry flag; a branch-on-carry follows. The variable TEMP2 is divided by two (using a logical shift right, LSR). The rest of the listing is patterned after the Basic program.

Here's an interesting comparison: The Basic listing can do about three samples per second; using the JOYSTK command increases the sample to about 40 per second; the machine code routine can make over three and a half *thousand* samples each second.

DIGITAL RECORDER. So how does knowing how the joystick input works help you make a digital recording of your voice? Remember that the first stage of digital recording is the transformation of air pressure variations into proportional voltages. If the joystick input can measure joystick voltages, then shouldn't it follow that it can measure voltages from, say, a cassette player or microphone amplifier? Yes, it can. Basic can't do it because it's too slow. But machine code — at 3,600 samples per second — can pull in a reasonable representation of a voice.

Once you've got the voice stored as a series of numbers, reproducing it is easy. You swing through memory, taking

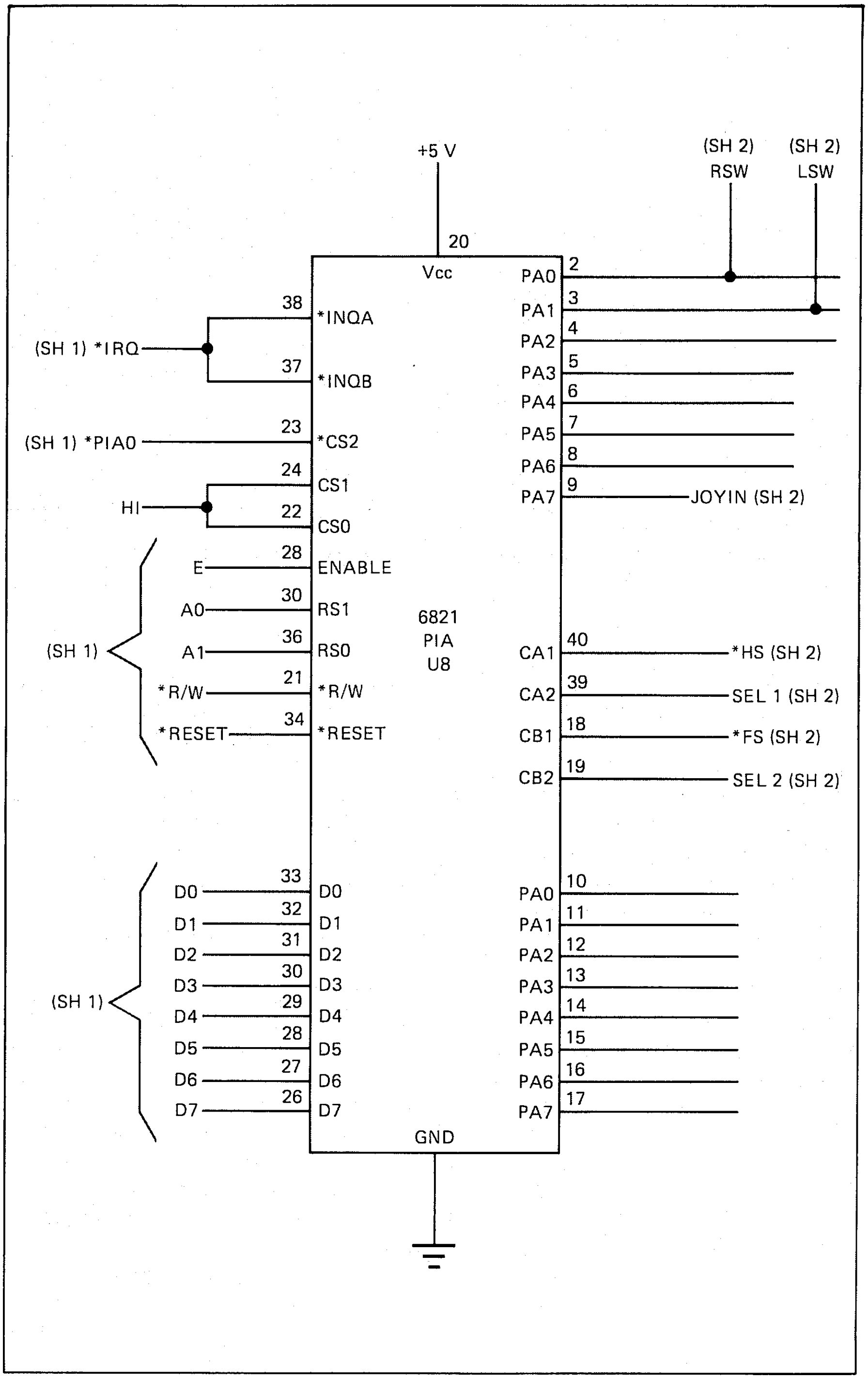


Figure 3. Joystick input PIA, showing the bit assignments of various peripherals; interrupt handling is also visible. Reprinted with permission from the *TRS-80™ Color Computer Technical Reference Manual*, page 68.

PROGRAM LISTING 1

Basic emulation of a joystick input routine.

16K Extended Color Basic

```
1 CLS
2 X=&HFF00                '* PORT VALUE
3 DAC = &HFF20            '* D/A CONVT.
4 ADC = &HFF00            '* COMPARATOR
5 POKE &HFF01,&H34        '* CTRL. INFO
6 POKE &HFF03,&H37        '* CTRL. INFO
7 A = &H80                '* B 10000000
8 T1 = 6                  '* ITERATIONS
9 T2 = &H80               '* B 10000000
10 POKE DAC,A             '* SEND VALUE
11 B = PEEK(ADC)          '* BYTE BIT 7
12 B = B AND &H80         '* TEST BIT 7
13 IF B = 0 THEN 17       '* IF FLIPPED
14 T2 = T2/2              '* ELSE  LESS
15 A = A + T2             '* ADD  VALUE
16 GOTO 19                '* GO TO NEXT
17 T2 = T2/2              '* IT IS MORE
18 A = A - T2             '* ELSE  MORE
19 T1 = T1 - 1            '* SUB. VALUE
20 IF T1>0 THEN 10        '* DONE  YET?
21 PRINT@0,"JOY =";       '* A  MESSAGE
22 PRINTINT(A/4)          '* A JOYSTICK
23 GOTO 7                 '* & DO AGAIN
```

PROGRAM LISTING 2

Machine code version of a joystick input routine, using a binary search.

```
        LDA     #$34
        STA     $FF01
        LDA     #$37
        STA     $FF03
*
NEXTIN  LDA     #$06
        STA     <TEMP1
        LDA     #$80
        STA     <TEMP2
*
AGAIN   STA     DAC
        ROL     ADC
        BCC     BIGGER
LESSER  LSR     <TEMP2
        ADDA    <TEMP2
        BRA     JUMP
BIGGER  LSR     <TEMP2
        SUBA    <TEMP2
        BRA     JUMP
JUMP    DEC     <TEMP1
        BNE     AGAIN
        STA     ,X+
*
```

each byte and feeding it to the digital-to-analog converter. The reproduced sound appears at the output. The only thing you need to keep in mind is that homing in on the sound sample takes longer than reproducing the sound by outputting a sample value...so you've got to add a little delay to keep the input and output rates matched.

Look at Listing 3, a complete record-reproduce program for the Color Computer. The heart of the joystick input program you've already seen is found in Lines 240–520. The output routine is in 620–690. You can see that Lines 650–670 add the delay needed to keep input and output matched.

There are only a few other items to note. First of all, direct addressing is used to speed the program along (Lines 220 – 230). Next, interrupts are turned off to keep the sampling rate rock solid (Line 200). And finally, the "fast RAM" mode is used during the input and output sections (Line 210).

What is *fast RAM*? What are the mysterious high-speed POKEs all about? Here's the deal. The innards of your Color Computer are created from off-the-shelf components suited to many purposes. One of these is called the SAM (Synchronous Address Multiplexer). The SAM can support several computer *configurations* — that is, several combinations of types and sizes of RAM. The Color Computer uses several memory sizes (4K, 16K, 32K, 64K) but supports only one RAM type: dynamic.

Dynamic memory, through creative development and lots of luck, is low in power required and physically compact. Best of all, it is inexpensive. On the other hand, its special construction requires a continual *refresh* of its contents. This refresh is accomplished by reading 128 sequential memory locations at least every microsecond. The refresh complicates computer design because it has to be squeezed in between sequential instructions carried out by the central processing unit (CPU).

Designers at Motorola solved the problem by developing a combination of three sophisticated parts: the 6809 CPU itself, the SAM, and the 6847 video display generator. These work together to execute computer instructions, refresh the dynamic memory, and provide a video display.

Imagine the master clock of the Color Computer as a sequence of regular on-off pulses. During pulses 1, 3, 5, 7, 9 and so forth, the CPU executes its instructions from memory. During even-numbered pulses 2, 4, 6, 8 and so on, the memory is freed from the CPU to create the video display or to carry out the dynamic memory refresh. Since the video display generator also uses sequential addresses when it draws the screen, any video display has the effect of refreshing the memory.

When you turn your Color Computer on, it creates those alternating pulses — first one for the CPU, then one for the video or refresh. It provides the pulses to all of memory. When you POKE 65495,0 (the usual so-called "high-speed" mode), you keep the refresh and video to the dynamic memory, but drop the extra pulse out when the computer uses the Basic ROM. Basic ROM is static and needs no refresh.

PROGRAM LISTING 3

Continuous record/reproduce program.

32K Extended Color Basic

```
          FF20    00100 DAC     EQU    $FF20
          FF00    00110 ADC     EQU    $FF00
          FFD9    00120 FAST    EQU    $FFD9
          FFD8    00130 SLOW    EQU    $FFD8
          4000    00140 MEMBOT  EQU    $4000
          8000    00150 MEMTOP  EQU    $8000
                  00160 *
3F00              00170         ORG    $3F00
          3F      00180         SETDP  $3F
                  00190 *
3F00 1A   50      00200 START   ORCC   #$50
3F02 B7   FFD9    00210         STA    FAST
3F05 86   3F      00220         LDA    #$3F
3F07 1F   8B      00230         TFR    A,DP
                  00240 *
3F09 8E   4000    00250         LDX    #MEMBOT
                  00260 *
3F0C 86   34      00270         LDA    #$34
3F0E B7   FF01    00280         STA    $FF01
3F11 86   37      00290         LDA    #$37
3F13 B7   FF03    00300         STA    $FF03
                  00310 *
3F16 86   06      00320 NEXTIN  LDA    #$06
3F18 97   9B      00330         STA    <TEMP1
3F1A 86   80      00340         LDA    #$80
3F1C 97   9C      00350         STA    <TEMP2
                  00360 *
3F1E B7   FF20    00370 AGAIN   STA    DAC
3F21 79   FF00    00380         ROL    ADC
3F24 24   06      00390         BCC    BIGGER
3F26 04   9C      00400 LESSER  LSR    <TEMP2
3F28 9B   9C      00410         ADDA   <TEMP2
3F2A 20   06      00420         BRA    JUMP
3F2C 04   9C      00430 BIGGER  LSR    <TEMP2
3F2E 90   9C      00440         SUBA   <TEMP2
3F30 20   00      00450         BRA    JUMP
3F32 0A   9B      00460 JUMP    DEC    <TEMP1
3F34 26   E8      00470         BNE    AGAIN
3F36 A7   80      00480         STA    ,X+
                  00490 *
3F38 8C   8000    00500         CMPX   #MEMTOP
3F3B 26   D9      00510         BNE    NEXTIN
                  00520 *
3F3D 4F           00530         CLRA
3F3E 1F   8B      00540         TFR    A,DP
3F40 B7   FFD8    00550         STA    SLOW
3F43 39           00560         RTS
                  00570 *
3F80              00580         ORG    $3F80
                  00590 *
3F80 1A   50      00600 PLAYBK  ORCC   #$50
3F82 B7   FFD9    00610         STA    FAST
3F85 8E   4000    00620         LDX    #MEMBOT
3F88 A6   80      00630 SPEAK   LDA    ,X+
3F8A B7   FF20    00640         STA    DAC
3F8D C6   2C      00650         LDB    #$2C
3F8F 5A           00660 LOOP    DECB
3F90 26   FD      00670         BNE    LOOP
3F92 8C   8000    00680         CMPX   #MEMTOP
3F95 26   F1      00690         BNE    SPEAK
3F97 B7   FFD8    00700         STA    SLOW
3F9A 39           00710         RTS
                  00720 *
3F9B              00730 TEMP1   RMB    1
3F9C              00740 TEMP2   RMB    1
                  00750 *
          0000    00760
00000 TOTAL ERRORS

ADC     FF00        MEMTOP  8000
AGAIN   3F1E        NEXTIN  3F16
BIGGER  3F2C        PLAYBK  3F80
DAC     FF20        SLOW    FFD8
FAST    FFD9        SPEAK   3F88
JUMP    3F32        START   3F00
LESSER  3F26        TEMP1   3F9B
LOOP    3F8F        TEMP2   3F9C
MEMBOT  4000
```

The "super high speed" mode is POKE 65497,0. If you try it, though, your video display will go into a frenzy, and you might lose your programs. That's because this POKE tells the SAM to bother with neither refresh nor video, but rather to consider the computer a machine with static RAM and no display output. No refresh, no video.

In this mode you might lose your programs because the refresh cycle is missing and the memory gradually "forgets" its contents. The mode conditions are summarized in Table 1.

But why does the super high speed work for program Listing 3? In this case, sequential memory is being filled with voice information, and then the voice information is played back. At this program's speed, the very act of filling and reading back the sequential memory acts as a memory refresh! The contents remain intact, undisturbed... though the video display is useless.

Once you've saved the source program in Listing 3 (W VOICE), you can assemble it to memory (A/IM/AO), quit the editor/assembler (Q), and protect Basic memory (CLEAR 200,&H3F00). To use it, you'll need a joystick cable adaptor (see Figure 4). Turn the volume up on your television or monitor, pop a voice or music cassette into the recorder, and run the following lines:

```
10 EXEC &H3F00
20 EXEC &H3F80
30 GOTO 10
```

The screen will go haywire, but the computer will record and play back the sound every two seconds. If you want to see the actual sound displayed as a sound wave, tap the Break key and enter and run Listing 4.

USING VOICE WITH BASIC. Since Basic programs have to run at normal speed to refresh the RAM, the voice you use with Basic won't have the quality you can hear at the higher speed. No matter; for some programs, any voice at all is better than none, so drop Lines 120, 130, 210, 550, 610 and 700 from Listing 3. You'll be able to store four seconds of sound in the 16K of memory above location $3FFF.

DIGITAL PUNCH IN THE NOSE. Now I'll turn away from digital voice recording to explore the first of two techniques for keeping the time of day. Keeping the "real time" is tricky because computer time is relative to its own master clock. The Color Computer's master clock (running at 894,886 pulses per second) is meaningful in just one way — you know exactly how much time each processor instruction takes. That's it; there's no easy way a program can know, so to speak, when one second has passed. Something has to knock on its door, interrupt its reverie, and punch it in the nose. That digital punch is known as an *interrupt*.

The Color Computer was built with interrupts in mind. The cursor flashes to the rhythm of interrupts, the Sound command counts them out to learn when it's done, and the timer runs with them. Fancy games depend on their presence.

Using interrupts isn't difficult, though it does call for a slight change of perspective. When you think of a program, you probably think of it as a logical series of commands followed one at a time without distraction. Interrupts seem to defy this concept, forcing the program away from its appointed tasks to work on something else.

That scenario is basically correct. The surprise is that the program in progress seems to have amnesia — it never remembers that it has been distracted from its work, can never recall what has happened during those lost microseconds.

The ideal interrupt happens like this: a signal (say, one synchronized with the video display) appears on the CPU's interrupt-request (IRQ) connection. If the machine code program in progress (such as the Basic interpreter) has *enabled* the interrupt process, then this signal will be accepted as soon as the CPU is finished with its present instruction. The machine state (all the registers, program counter, and flags) gets stashed, and the program counter is given a new value. The new value in the program counter points to an "interrupt service routine."

The interrupt service routine takes care of whatever the programmer had in mind for that particular occurrence of the interrupt. When the interrupt service routine is finished, the original machine state (group of registers and flags) is restored and the main program continues exactly where it left off. A little time has mysteriously disappeared during the course of the main program.

A real-time clock is a perfect example of interrupt use.

VIDEO SYNCHRONIZATION. The Color Computer invites interrupt exploitation because both vertical and horizontal video signals can be used to generate interrupts. Television images require electrical pulses to cause each frame to begin at the top of the screen, and other pulses to cause each scanning line to begin at the left side of the screen. The signal to begin each frame is called vertical synchronization, and the signal to begin each line is called horizontal synchronization.

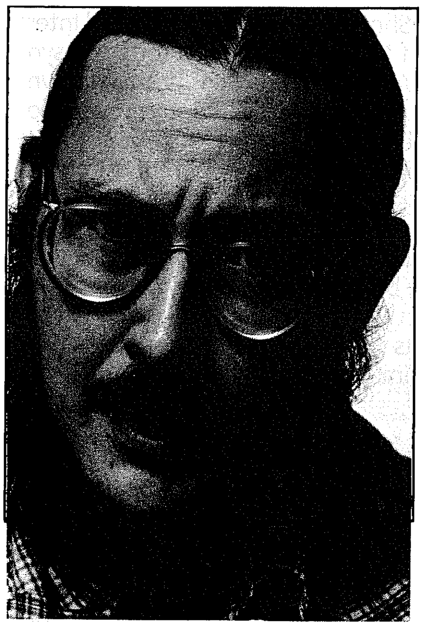
Vertical sync, which occurs 60 times every second, is an ideal candidate for interrupt use. After six pulses have occurred, then, $1/10$ second has passed; after 60 pulses, one full second has passed. By keeping strict track of the pulse count, you can have a real-time clock.

Look at Listing 5, beginning at Line 200. At each interrupt pulse, the CPU will be directed there. Register X points to the final character in the clock image stored in memory, in ASCII, which starts at 00:00:00.00. The final digit is the $1/60$ of a second counter. This value is incremented (Line 220), and checked to see if it has reached six. If it's less than six, the image is transferred to the screen by the exit routine beginning at Line 490. If the counter has reached six, it is reset to zero and the $1/10$ of a second counter is incremented. If it hasn't reached ten, the routine exits; otherwise, the one-second counter is incremented. And so on, up to a total count of 99:59:59.95.

The exit routine transfers ten characters of the image to the screen. The $1/60$ of a second digit is not displayed.

POKE ADDRESSES		MODE	CONDITION
65494	65496	00	NORMAL
65495	65496	01	FAST ROM
65494	65497	10	FAST RAM
65495	65497	11	ALL FAST

Table 1. Memory clock speed control conditions within the Synchronous Address Multiplexer (SAM).



PROGRAM LISTING 4

Short Basic routine to display the sound wave representation of the sound recorded by Listing 3.

32K Extended Color Basic

```
10 CLS0
20 FORX = &H4000 TO &H8000
30 SET (Y, PEEK(X)/8, 5)
40 Y=Y+1: IF Y>63 THEN CLS0: Y=0
50 NEXT
```

PROGRAM LISTING 5

Real-time clock using the internal interrupts which occur 60 times each second.

32K Extended Color Basic

```
3F00                          ORG     $3F00
                       *
3F00 1A      50        INTOFF ORCC    #$50      * TURN INTERRUPTS OFF
3F02 8E      3F10             LDX     #START    * POINT X TO SERVICE ROUTINE
3F05 BF      010D             STX     $010D     * STORE ROUTINE TO IRQ VECTOR
3F08 86      37               LDA     #$37      * VALUE 00110111 FOR MASKING
3F0A B7      FF03             STA     $FF03     * TURN ON VERTICAL SYNC
3F0D 1C      EF               ANDCC   #$EF      * TURN INTERRUPTS ON
3F0F 39                       RTS               * AND BACK TO BASIC "OK"
                       *
3F10 8E      3F77      START  LDX     #IMAGE+10     * POINT X TO 1/10 SEC.
3F13 C6      30               LDB     #$30      * B BECOMES ASCII OFFSET
3F15 6C      84               INC     ,X        * INCREMENT 1/10 SECONDS
3F17 A6      84               LDA     ,X        * GET 1/10 SECONDS VALUE
3F19 81      36               CMPA    #$36      * IS 6/10 SECONDS COUNTED?
3F1B 2D      2C               BLT     OUT       * IF NOT 6/10 SECONDS, OUT
3F1D 8D      40               BSR     DEC1      * ELSE BAC UP 1 MEM. LOCATION
3F1F 81      3A               CMPA    #$3A      * IS IT 1 SECOND YET?
3F21 2D      26               BLT     OUT       * IF NOT 1 SECOND, OUT
3F23 8D      41               BSR     DEC2      * ELSE BACK UP 2 MEM. LOCNS.
3F25 81      3A               CMPA    #$3A      * IS IT 10 SECONDS YET?
3F27 2D      20               BLT     OUT       * IF NOT 10 SECONDS, OUT
3F29 8D      34               BSR     DEC1      * BACK UP 1 MEM. LOCATION
3F2B 81      36               CMPA    #$36      * IS IT 60 SECONDS YET?
3F2D 2D      1A               BLT     OUT       * IF NOT 60 SECONDS, OUT
3F2F 8D      35               BSR     DEC2      * ELSE BACK UP 2 MEM. LOCNS.
3F31 81      3A               CMPA    #$3A      * IS IT 10 MINUTES YET?
3F33 2D      14               BLT     OUT       * IF NOT 10 MINUTES, OUT
3F35 8D      28               BSR     DEC1      * ELSE BACK UP 1 MEM. LOCATION
3F37 81      36               CMPA    #$36      * IS IT 60 MINUTES YET?
3F39 2D      0E               BLT     OUT       * IF NOT 60 MINUTES, OUT
3F3B 8D      29               BSR     DEC2      * ELSE BACK UP 2 MEM. LOCNS.
3F3D 81      3A               CMPA    #$3A      * IS IT 10 HOURS YET?
3F3F 2D      08               BLT     OUT       * IF NOT 10 HOURS, OUT
3F41 8D      1C               BSR     DEC1      * ELSE BACK UP 1 MEM. LOCATION
3F43 81      3A               CMPA    #$3A      * IS IT 100 HOURS YET?
3F45 2D      02               BLT     OUT       * IF NOT 100 HOURS, OUT
3F47 E7      84               STB     ,X        * PLACE $30 (ASCII ZERO)
                       *
3F49 108E    0416      OUT    LDY     #$0416    * POINT TO RIGHT SCREEN
3F4D 8E      3F6D             LDX     #IMAGE    * POINT X TO CLOCK IMAGE
3F50 C6      0A               LDB     #$0A      * COUNT 10 SCREEN POSITIONS
3F52 A6      80        LOOP   LDA     ,X+       * GET CHARACTER FROM CLOCK
3F54 A7      A0               STA     ,Y+       * AND PLACE IT ON THE SCREEN
3F56 5A                       DECB              * DONE WITH IMAGE YET?
3F57 26      F9               BNE     LOOP      * IF NOT, THEN GET NEXT CHAR.
                       *
3F59 B6      FF02             LDA     $FF02     * CLEAR VERT. SYNC LATCH
3F5C 7E      894C             JMP     $894C     * AND TO BASIC TO DO RTI
                       *
3F5F E7      84        DEC1   STB     ,X        * PLACE $30 (ASCII ZERO)
3F61 6C      82               INC     ,-X       * BACK UP ONE MEM. LOCATION
3F63 A6      84               LDA     ,X        * GET VALUE FROM IMAGE
3F65 39                       RTS               * BACK TO MAIN PROGRAM
                       *
3F66 E7      84        DEC2   STB     ,X        * PLACE $30 (ASCII ZERO)
3F68 6C      83               INC     ,--X      * BACK UP TWO MEM. LOCATIONS
3F6A A6      84               LDA     ,X        * GET VALUE FROM IMAGE
3F6C 39                       RTS               * BACK TO MAIN PROGRAM
                       *
3F6D         30        IMAGE  FCC     /00:00:00.00/
             30
             3A
             30
             30
             3A
             30
             30
             2E
             30
             30
                       *
             3F00             END     INTOFF
00000 TOTAL ERRORS

DEC1    3F5F
DEC2    3F66
IMAGE   3F6D
INTOFF  3F00
LOOP    3F52
OUT     3F49
START   3F10
```

The amazing thing about this 116-byte routine is that, as programs go, it is totally mundane. It's nothing more than increment, test, proceed, move memory, and exit. The only item to note is found at Line 570. The command JMP $894C lets Basic complete its part of the interrupt service routine chain (cursor, timer, sound) before returning to the program in progress.

Save the source code (W CLOCK), assemble this program into memory (A/IM/AO), quit the editor/assembler (Q), protect memory (CLEAR 200,&H3F00) and execute the machine code program (EXEC &H3F00). The clock will appear in the top corner of the screen, in reverse-video characters.

This software clock will survive resets and CLOADs (not necessarily CLOADMs), and will run under most program conditions. Naturally, if you have a 32/64K machine, you'll want to move it up in memory.

INTERRUPT THEORY. I passed by that interrupt theory pretty quickly to give you a chance to see it perform. There are some important things to understand before you put interrupts to work for yourself.

First of all, since interrupts come through at a regular pace, you've got to make sure you keep ahead of that pace. If you don't, you will not get back to the main course of your program properly.

Rule Number 1 is: make your interrupt routines short and efficient. When using the relatively laggard 60-per-second pulse of the vertical sync, that isn't much of a problem, because 1/60th of a second is 16,667 microseconds, or 14,915 Color Computer master clock pulses. The clock interrupt routine is lengthiest at the change back to 00 hours. Including the machine state saving at the interrupt itself, this requires 303 master pulses plus some more for Basic's timer, cursor and sound. The shortest execution for just a 1/60 second count is only 35 master clock pulses. Overall, the clock routine's time is minuscule with respect to the 14,915 master pulses available.

But what if you wanted to create a kind of super stopwatch? The Color Computer gives you a chance to do that, too. There isn't room to go into the details, but the horizontal synchronization from the video can also be used as an interrupt. This interrupt occurs 15,750 times each second, or once every 63.5 microseconds. This time, 63.5 microseconds is just over 56 computer clock cycles... hardly time to do much of anything. If you think 56 clock cycles isn't any time at all, though, turn to Steve Bjork's article on interrupts (Display Modes, December, 1983 *TCCM*); the pro speaks!

Figure 3 shows the schematic for interrupt handling in the Color Computer. Shown is the 6821 Peripheral Interface Adaptor (PIA). At the left the vertical (or field) synchronization is marked FS, and feeds CB1; horizontal synchronization is marked HS, and feeds CA1. By proper programming (refer to the 6821 data sheet for details), either can be fed through to the CPU's interrupt request pin via IRQA and IRQB (right side of the PIA).

Oh yes. Here's Rule Number 2: if you are using interrupts in an independent program that does not work with the Basic ROMs at all, be sure to reset the interrupt latch by reading the correct PIA location (LDA $FF02). Otherwise, further interrupts won't be passed through the PIA, and you'll lose all the timing.

WHAT DOES IT ALL MEAN? I won't tie all this together until later in this series, but keep in mind that all these concepts involve timing — the ultra-fast timing of digital recording, the hidden timing of interrupts, and (in the next part) the slow timing of interfacing with external electronics.

Ed's Note: Articles referred to, except Part I of this series, are found in back issues of The Color Computer Magazine, *unless otherwise noted.*

Memory Test

By Stephen P. Allen

Memory prices the way they are now make it easy for a lot of people to upgrade their computer to a full 64K. It's a great feeling of satisfaction—"Aah, now I've got more power!"

"Wait a minute, how do I know I've *really* got 64K? Radio Shack says I've got to get a disk drive and OS-9 to use it. I don't really know if I even have it! I wish there was some way to test."

Is that you? Run this program.

RAMCHECK will give your computer memory a proper workout. If your memory chips extend all the way to 64K, you'll return from this program running Basic in RAM. And instead of OK, you'll see GO as the prompt. If your RAMs flunk the test, RAMCHECK will tell you where your memory effectively ends.

The Program

The assembly language routine will alter the contents of all memory above 16K. The first job of the Basic program is to move the 6809's stack out of the way (leaving room for the routine). Don't change the addresses in Lines 30 and 40! If you move them higher there's the good chance the routine or the stack will be over-written during the test.

After the necessary space is cleared, the routine is POKEd in and called with a USR function. The value of AR, returned by the assembly language subroutine, will be zero if you have 64K, or it will be the address where memory "ceased to exist." GIVABF, at $B4F4, will treat any number given it that's larger than 32767 as negative. That's why 65536 is added if AR◄0.

Switching between ROM and RAM is done by the 6883, that 40-pin mystery responsible for so much behind-the-scenes footwork. In all those 40 pins there are no data lines—the 6883 is controlled by simply writing to one of its 32 memory addresses. What's written doesn't matter; it can't read data, only addresses. The addresses that matter to us here are $FFDE (select ROMs) and $FFDF (select equivalent RAMs).

The byte values used to test the RAMs are $55 and $AA. In binary these correspond to 01010101 and 10101010. The values were chosen because these are the most difficult for memory to store—a one next to a zero may "influence" that zero to leak up to one, and vice-versa. I say *may* influence; I haven't seen it happen yet. *(end)*

Program Listing. RAM Check (Basic)

```
10 '      RAMCHECK
20 '
30 ' TESTS FOR 64K RAMS
40 '  IF YOU HAVE THEM YOU RETURN
50 '  IN ALL RAM MODE
60 '  ELSE END OF MEMORY IS
70 '  DISPLAYED
80 '
90 CLS:PRINTSTRING$(32,42)
100 PRINT@75,"RAMCHECKER
110 CLEAR200,12200
120 AD=12202:DEFUSR0=AD
130 READ OP:IFOP=1000 THEN150
140 POKEAD,OP:AD=AD+1:GOTO130
150 PRINT"CHECKING NOW...":HR=USR0(0)
160 IFHR=0THEN190ELSEIFHR<0THENHR=HR+65536
170 PRINT"RAM TEST FAILS AT $"HEX$(HR)
180 CLEAR200, HR-1:END
190 PRINT"NOW IN RAM!":CLEAR200,&H7FFF:END
200 DATA 142,63,255,26,80,183,255
210 DATA 223,134,85,167,132,161,132
220 DATA 38,44,72,104,132,161,132
230 DATA 38,37,48,1,140,255,0
240 DATA 37,234,142,128,0,183,255
250 DATA 222,166,132,183,255,223,167
260 DATA 128,140,240,0,37,241,134
270 DATA 71,198,79,253,171,238,79
280 DATA 95,126,180,244,183,255,222
290 DATA 31,16,126,180,244,1000
```

Program Listing. RAM Checker (Assembly Language)

```
                    00100 * RAMCHECKER
                    00110
                    00120 * Tests for 64K rams
                    00130 * If you have them it copies
                    00140 * Basic to Ram and returns.
                    00150 * Else it tells where the Ram ends.
                    00160
                    00170
                    00180
0000 8E    3FFF     00190 START  LDX    #$3FFF  *Point to start of 32K
0003 1A    50       00200        ORCC   #$50    *Interrupts off
0005 B7    FFDF     00210        STA    $FFDF   *Select "All Ram" mode
                    00212
0008 86    55       00220 LOOP1  LDA    #$55    *Binary 01010101
000A A7    84       00230        STA    ,X      *Store it
000C A1    84       00240        CMPA   ,X      *Did we get it back?
000E 26    2C       00250        BNE    ERROR   *if not, this is end of Ram
0010 48             00260        ASLA           *binary 10101010
0011 68    84       00270        ASL    ,X      *Same to Ram
0013 A1    84       00280        CMPA   ,X      *Did the Ram follow?
0015 26    25       00290        BNE    ERROR   *if not, end of Ram
0017 30    01       00300        LEAX   1,X     *Test OK, move up one
0019 8C    FF00     00310        CMPX   #$FF00  *Start of PIAs?
001C 25    EA       00320        BLO    LOOP1   *not yet-- keep going
                    00330
                    00332 * TEST SUCCESSFUL:
                    00340 * COPY BASIC TO RAM
                    00350
001E 8E    8000     00360        LDX    #$8000  *Start of Basic
0021 B7    FFDE     00370 LOOP2  STA    $FFDE   *Select Roms
0024 A6    84       00380        LDA    ,X      *Grab a byte from Rom
0026 B7    FFDF     00390        STA    $FFDF   *Select Ram mode
0029 A7    80       00400        STA    ,X+     *Duplicate byte in Ram
002B 8C    F000     00410        CMPX   #$F000  *End of Roms?
002E 25    F1       00420        BLO    LOOP2   *Not yet
                    00430
                    00440 * CHANGE "OK" TO "GO"
                    00450
0030 86    47       00460        LDA    #'G
0032 C6    4F       00470        LDB    #'O
0034 FD    ABEE     00480        STD    $ABEE   *"OK" location
                    00490
                    00500 * GIVE A ZERO TO BASIC:
                    00510 * SHOW RAM TEST OK
                    00520
0037 4F             00530        CLRA
0038 5F             00540        CLRB
0039 7E    B4F4     00550        JMP    $B4F4   *GIVABF
                    00560
                    00570 * ERROR: GIVE ERROR ADDRESS
                    00580 * BACK TO BASIC
                    00590
003C B7    FFDE     00600 ERROR  STA    $FFDE   *Select Rom mode
003F 1F    10       00610        TFR    X,D     *Last Ram address to D
0041 7E    B4F4     00620        JMP    $B4F4   *and back to Basic's GIVABF
           0000     00630        END
00000 TOTAL ERRORS

ERROR   003C
LOOP1   0008
LOOP2   0021
START   0000
```

Who's Fibonacci?

Author's Note: The following information is based on true fact, believe it or not. (Anyone know a false fact?)

Leonardo Fibonacci was an Italian mathematician who was a leading influence in mathematics during the Middle Ages. Scholars and generally polite people refer to him as Leonardo of Pisa. His neighbors in Pisa, however, called him *Bigollone* or "the Blockhead." His family name was of no help, either. His father's name, Bonaaccio, meant "Simpleton." Hence, Fibonacci was "Son of Simpleton."

He spent his formative years in the North African city of Bujaia, where his father was stationed as a customs official. Inquisitive young Leonardo was educated by the Muslims there, who taught him the Arabic numeral system. He quickly realized that working with this system was far simpler than the Roman numeral system in use in Europe; any half-intelligent school child knows that 1984 is easier to deal with than MCMLXXXIV!

After returning to Pisa as a young scholar, he spent several years contemplating this discovery. Leonardo was often seen wandering in a fog. When inspiration hit, he would grab a piece of chalk and absentmindedly scribble numbers on a nearby wall. At the age of 27, he published an historic manuscript, *Liber Abaci,* which introduced Arabic numerals to the European continent. This in itself was a major accomplishment, but one section of the book contained a theoretical problem that proved to be most interesting of all. The solution resulted in the discovery of the Fibonacci series.

The problem goes something like this: suppose someone placed a pair of rabbits in an enclosed area. If these rabbits were allowed to breed, how many pairs of rabbits would be born over the course of one year? Fibonacci figured that every month a pair of rabbits would be born, and that rabbits begin to bear young two months after their own birth. After this hare-raising experience, it was assumed that the original rabbits and their offspring would total 233 pairs. Fibonacci listed the number of pairs at the end of each month: 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233. When analyzing this group of numbers, he found that each number as the sum of the two preceding numbers. These numbers—1, 2, 3, 5, 8, 13 . . . —became known as the Fibonacci series. This series theoretically goes on forever, although you might end up with some thoroughly exhausted bunnies after awhile. By the way, there is no historical evidence that Fibonacci actually tried this experiment.

The series itself appears to be composed of seemingly random numbers, right? Well, any self-respecting Fibonacci scholar would tell you that each number has a special relationship to the numbers surrounding it. If you divide a number in the Fibonacci series by the next highest number, you will discover that the quotient is always about .6, or more precisely, .618034. This precise number works when the Fibonacci numbers are large enough to be precise (after about the 14th in the sequence). This number, .618034, is referred to as the Golden Mean. The ratio .618034 to one is the mathematical basis for eye-pleasing shapes in art, spiral galaxies, the curvature of a snail shell, and the shape of playing cards. Add to this the occurrence of Fibonacci numbers in botany and music, and you come up with some interesting patterns.

Plant life provides many examples of Fibonacci numbers. The number of pine needles that grow in a cluster on most species of pine trees tend to be 2, 3, or 5. Counting the number of petals on a daisy will most likely yield a Fibonacci number. Phyllotaxis, the arrangement of leaves on a stem, provide further evidence that these numbers are more than mere coincidence. The number of leaves and turns on a stem are almost always Fibonacci numbers.

Musicians are aware that an octave is composed of eight notes. On a keyboard instrument, this is represented by eight white keys and five black keys for flats and sharps, for a total of 13.

Being the skeptic I am, I decided to check some of the number relationships. The pineapple's scaly outer skin is supposed to be made up of three distinct groups of logarithmic spirals; five go sharply in one direction, 13 in another, and finally eight in the third spiral. A trip to the local Valu-King proved this to be true. After checking about three pineapples in the produce department, I looked up and found myself staring at a plump Italian lady with an impatient expression on her face. I could have sworn she mumbled "Bigollone" as she snatched a pineapple and left.

Fort Worth, take note. The ratio of the Color Computer 2's length to width dimensions is .7106896, substantially closer to the Golden Mean than the first Color Computer, at .93220. The esthetically ideal Color Computer would have cabinet dimensions of 10¼ × 16⅝. ■

Bibliography for further reading:

Hoffer, William, "A Magic Ratio Recurs Throughout Nature," *Smithsonian,* December, 1975, pp. 110–124.

Hoggatt, Verner E., *Fibonacci and Lucas Numbers,* Boston: Houghton Mifflin Co., 1969.

```
*****************************************************************
*** NEWS FROM UNDER COLOR **** NEWS FOR UNDER THE COLORS ***
**************** Compiled November 26, 1984 ****************
** Send news to: Under Color, Box 6809, Roxbury, VT 05669 **
****************** CompuServe 70136,1257 ******************
*****************************************************************
```

COCO GETS REDESIGNED AGAIN ... The Color Computer finally outstrips the venerable Model I in Tandy remakes. There are now NINE VERSIONS ... CoCo1-C, D, E, NC; TDP-100; CoCo2, 2-A, 2-B; Korean CoCo2, KC2-B ... All the reworkings since NC are "economically driven designs", as one of the key engineers told U.C. ... Repairs to the new units? Forget it. SWAP the board.

* * * * *

SOCKETS VANISHED in the latest two Korean CoCos; in the first version, all but the SAM and the RAM are soldered in place. In the second version, even the SAM IS SOLDERED in, but more surprising are the RAMs -- just two sockets and two extra connectors. What's up? These are TWO 16K x 4 RAMs and -- get this -- a PIGGYBACK BOARD is needed to upgrade to 64K (does this sound like a 1980 deja-vu?) ... U.C.'s call to Forth Worth engineers revealed that the PIGLET REDESIGN WASN'T COMMON KNOWLEDGE there ... But wait! There's more. Read on...

* * * * *

$199 64K COCOs continue to have eight 64K x 1 RAMs on board, but the 64K RAMs are SOLDERED IN PLACE! And, if that isn't enough to confuse users, reliable rumor (November 24) has it that Radio Shack's 26-3134B and 26-3136B computers are CoCos with LOWERCASE BUILT IN, and won't be released until AFTER THE HOLIDAYS. Using the 6847-T1 for a video display generator -- not Motorola's hot new RMS high-resolution chip set, as previously rumored -- the NEW COCO loses none of its previous features and compatibility, but doesn't gain in the high-resolution game. So when the warehouses are clear, LOOK FOR COCO REVISION #10 to hit the stores. Now, what about 256K MEMORIES, Tandy?

* * * * *

TECHNICAL DETAILS on the 16K Korean CoCo RAM setup: a 74LS785 version of the SAM, with TMS4416 RAMs, and Basic 1.3; two connectors and two sockets, with A0 through A7 from the SAM; D0 through D7 come from a new design enhancement, a 74LS623 octal bus tranceiver from Texas Instruments (is Tandy moving away from Motorola?); CAS, RAS, R/W*; +5v and ground. MAKE YOUR OWN 64K upgrade? U.C. will give you the details as soon as we get them, but one of our SoCal hardware gurus tells us IT ISN'T THAT SIMPLE.

* * * * *

The MAGAZINE SHUFFLE at Ziff-Davis is still in the news. After cutting loose The Color Computer Magazine and other small computer rags, Z-D sold a boxful of titles for over $100 million dollars ... that's NINE FIGURES. ... Meanwhile, on CompuServe's Color Special Interest Group (SIG), angry talk continues from subscribers who sent in the subscription in The Color Computer Magazine's last issue, and can't get a refund from Z-D. If you're a newcomer to CompuServe, GO PCS-126 and have a look around.

* * * * *

EVER WONDER WHAT IT COSTS to build one CoCo? How about 100 CoCos? U.C. did some shopping around for both retail and quantity pricing, and discovered some suprises ... THE DAYS OF BUILDING YOUR OWN MACHINE MAY BE OVER. Below are some retail and 100-piece quantities for Radio Shack's $199 64K Extended Basic Color Computer:

Item	Single	100's	Item	Single	100's
CPU	$ 9.95	$ 8.10	SAM	$25.00	$18.67
PIA (2)	9.00	3.60	64K RAMs	39.95	24.80
VDG	12.95	11.00	74LS273	1.80	.73
74LS244	1.80	.73	74LS02	.42	.21
74LS138	.80	.54	2 ROMs (N/A; see next item)		
555	.42	.18	2 EPROMs	30.00	8.60
crystal	4.95	1.80	switches	3.93	1.50
4 jacks	3.96	1.80	8 sockets	2.32	1.42

Item	Single	100's
2 special chips (based on similar parts)	5.50	1.00
47 capacitors (mixed, best prices)	14.69	4.43
13 diodes, 3 transistors	7.91	1.76
33 resistors, 2 inductors	4.43	.49
power transformer, cassette relay	7.90	5.50
video modulator, TV switch box, cable	13.90	4.80
cartridge connector, AC line cord	5.90	2.43
printed circuit board (quote)	(by hand)	22.00
keyboard	39.95	18.00
Basic programming books	(none)	8.00
THE TOTAL, NOT INCLUDING PHONE CALLS, SHIPPING, FCC INTERFERENCE APPROVAL, ADVERTISING, SCREWS, SPACERS, LABOR, PACKAGING, AND SUPPORT	$247.43	$152.09

BUY one complete CoCo at $199 and SAVE $48.43 over a hand-wired, caseless CoCo. BUILD 100 caseless CoCos for $15,209 save $4,691 over buying them (but who will PAY THE LICENSE FEE FOR COLOR BASIC?). How much DOES RADIO SHACK PAY for parts in 10,000 quantity to make a PROFIT AT $199?

* * * * *

****
*** NEWS FROM UNDER COLOR **** NEWS FOR UNDER THE COLORS ***
**************** Compiled November 26, 1984 ****************
** Send news to: Under Color, Box 6809, Roxbury, VT 05669 **
****************** CompuServe 70136,1257 *******************
****

A NEW COLOR BBS in Monterey, California, is on line, and SysOp Marc Rigas wants everyone to call, since he says "I need a LOT OF CALLERS if I am to keep it running." Sounds like a CHALLENGE. Weekdays 6-8pm, 10pm-7am; Saturday 12-4pm; Sunday 10pm-7am. Call (408) 646-1850 and help Marc keep his BBS running.

* * * * *

An NTSC COLOR COMPUTER is available by special order from Radio Shack ... for you COLOR VIDEO MONITOR users. Reports from the first few purchasers offer mixed reactions to the quality of the signal; U.C. will provide comparatives soon. In the meantime, Radio Shack will NOT RETROFIT existing CoCos with the monitor modification.

* * * * *

SOFTWARE PROTECTION (AGAIN) ... Controversy has been raging around DELIBERATE SECTOR DAMAGE BY LASER BEAM used for PROTECTION of disk software. A few suppliers have been using the protection scheme, which introduces a deliberate pit in one specific location of the disk's oxide coating -- making it UNWRITABLE. The software tries to write information to that special sector ... it assumes the diskette is an original ONLY IF IT GETS AN ERROR. If not -- WHAMMO! It either bombs out (the nice version) or else it ERASES EVERYTHING IT CAN FIND ON THE DISK. Some distributors have RETREATED from this scheme, others defend it. Apparently no DISK HEAD DAMAGE will occur, since the laser pit system is carefully designed to resemble a normal, microscopic disk blemish.

* * * * *

HARDWARE PROTECTION is going on, too. A video digitizer is being released before the holidays with ALL THE ELECTRONICS POTTED IN EPOXY and NO SCHEMATIC provided. Apparently the item is LOW COST and the manufacturer is worried about gnomes running off to some HI-TECH SWEATSHOP and copying it. Repairs? Modifications? Better ask before you buy. With the small-computer SHAKEOUT underway now, you can't tell who will still be here NEXT CHRISTMAS.

* * * * *

OVERHEARD on the Color SIG: "Meanwhile back at primeval swamp where the original CoCo still lurks, it is seen as slowly succumbing to a sickly diet of arcade games -- seen as a computer with no redeeming social value."

* * * * *

COLOR BASIC 1.3 IS HERE ... but don't worry. It's only change is to accommodate the UPGRADED SAM in the new Korean CoCo. Everything else is (for a change) ENTIRELY COMPATIBLE with the Color Basic 1.2 that's been standard for about a year.

* * * * *

For future reference......

26-3001	4K Color Computer, Color Basic, grey case
26-3002	16K Color Computer, Extended Basic, grey case
26-3003	32K Color Computer, Extended Basic, grey case
26-3003A	64K Color Computer, Extended Basic, white case
26-3004	16K Color Computer, Color Basic, grey case
26-3004A	16K Color Computer, Color Basic, white case
26-3005	??
26-3006	Color Computer Dust Cover
26-3007	Color Computer Carrying Case
26-3008	Joysticks, original version
26-3009	4-pin to 5-pin cable
26-3010	13" Color TV Receiver
26-3011	4K MC-10 Micro Color Computer
26-3012	Deluxe Joysticks
26-3013	16K RAM cartridge for MC-10 Computer
26-3014	4-pin DIN to 25-pin DB-25 serial connector
26-3015	16K RAM kit for original Color Computer
26-3016	New Keyboard Upgrade, keyboard version #2
26-3017	64K RAM kit for Color Computer 1 & 2
26-3018	Extended Basic ROM kit
26-3019	Diagnostic ROM cartridge for Color 1
26-3020	4-pin DIN to 4-pin DIN cable
26-3021	??
26-3022	Color Disk #0, 5/12-volt version
26-3023	Color Disk #1, 2, 3
26-3024	Multi-Pak Interface
26-3025	Color Mouse
26-3026	16K Color Computer 2, Color Basic
26-3027	16K Color Computer 2, Extended Basic
26-3028	??
26-3029	Color Disk Drive #0, 5-volt version
26-3030	OS-9 with EDTASM+ Disk Version
26-3036	BASIC-09 Disk Version
26-3134	16K Color Computer 2, new keyboard
26-3136	16K Color Computer 2, Extended, new keyboard
26-3137	64K Color Computer 2, Extended, new keyboard
26-3134A	16K "Korean" 16K Color Computer 2
26-3136A	16K "Korean" 16K Color 2, Extended
26-5000	4K Videotex Terminal
26-5001	16K Videotex Terminal

WIN!

FIRST PRIZE: BILL BARDEN

By Bill Barden

RULES:

Ready to win *Big Money?* Do you like fame and notoriety? Would you like to be feted and dined in every major Color Computer center from Appleton, Maine to Mission Viejo, California? If this appeals to you, enter the First Great International *UnderColor* Magazine Programming and Pie Eating Contest!

In all seriousness, readers, this contest does give you a chance to win a prize. The grand prize consists of two hours of free advice from Bill Barden. If you win the contest, Bill will give you a call at a mutually agreed upon time to discuss your hopes, dreams, and aspirations regarding your pet Color Computer project—anything from graphics to assembly language. You can use this consulting time in any way you'd like, ask any questions you'd like, or even get tutorial advice. It's Bill's nickel on the phone charges, too!

Runnerups will be listed in the pages of *Under Color.*

Ready for the contest question? You don't have to be a super programmer to answer it—all it takes is a little research, a little hard work, and some originality. Here it is:

Write an Extended Basic program that draws a spiral figure on graphics screen 0 on PMODE 4 (256 × 192 resolution). The figure, when done, should look like Figure 1. That's it! You can use any combination of Basic commands you'd like, but no assembly language please. The winner will be the reader whose program executes in the shortest possible time. (Entries that simply list a series of Line commands will be disqualified.) When you've finished the program, send a listing and either a cassette tape or disk file to: William, Barden, Jr., P.O. Box 3568, Mission Viejo, CA 92692.

Along with your entry also indicate the approximate time the program takes to execute, and be sure to say it's an *Under Color* contest entry!

THE SPIRAL:

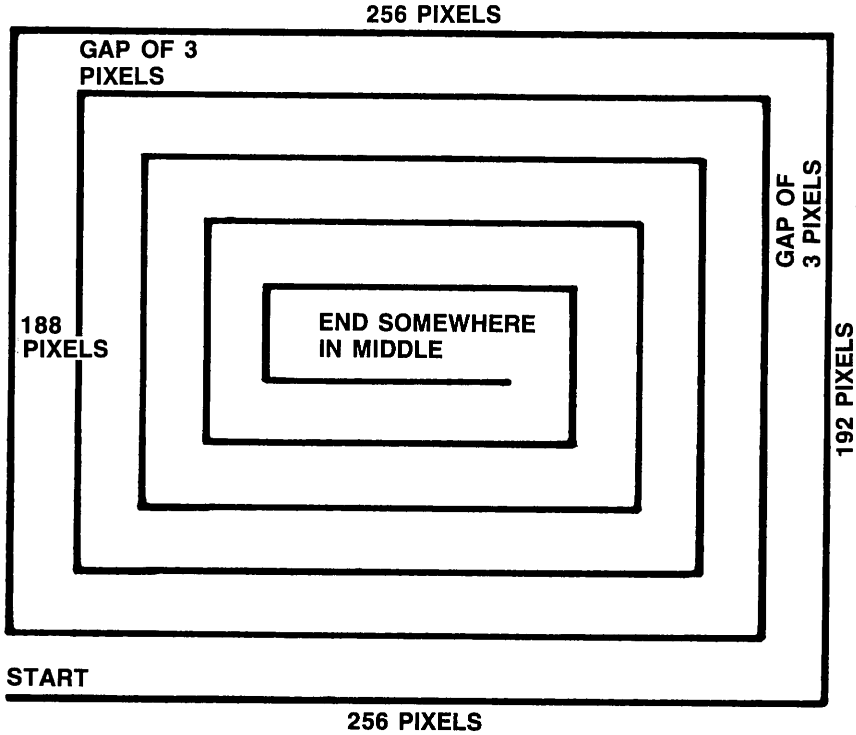


Figure 1. The Spiral

NOTES:

1. LINES MUST BE SEPARATED BY 3 PIXELS.

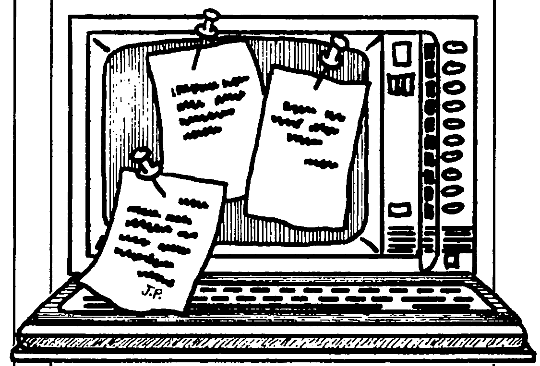
2. THIS FIGURE NOT ACCURATE AS FAR AS TOTAL NUMBER OF LINES.

JUDGING:

Bill will judge the entries on these criteria: he will first look at the approximate times for execution, and discard those entries that are obviously too slow. He will then test the remaining entries to see that they produce the figure, measure the execution time, and select a winner. The winner will then be notified via *Under Color.*

CLOSING DATES:

The closing date for entries is January 15, 1985 . The winner and runners-up will be announced early in 1985. Good luck!



Bulletin Boards

Evansville, IN
The Evansville Connection is on-line Saturdays and Sundays from 9 a.m.–9 p.m. Email, download, graphics, news: call 812-476-9453.

Tracy, CA
The Tracy BBS is available 24 hours a day: call 209-835-6496.

Joplin, MO
David Morgan is Sysop, 7 a.m.–12:30 a.m. Call 417-781-4020.

Yonkers, NY
Colorama is available 24 hours. Up- and down-load, want ads, boutique; Fred Siudym, Sysop. Call 914-965-7600.

Salem, OR
Les Taylor, Sysop; 300 baud, public access. Call 503-362-1351.

Reston, VA
Samoht is available Saturday and Sunday 6 a.m.–10 a.m.; M–F 6 p.m.–7 p.m. Mail, newsletters, downloads. Call 703-476-1147.

Camarillo, CA
Communicom-80: Beginners, D&D players. Call 805-484-5040.

Crawford, NE
Colorama; Rick Hungerford, Sysop. Available M–F 5 p.m.–8 a.m.; Saturday and Sunday 24 hours. Call 308-665-1526.

Long Beach, CA
Fineboard; Kenneth Fine, Sysop. Available Wednesdays 5 p.m.–12 a.m. Call 213-424-1803.

Port Mouton, Nova Scotia
Colorama; Jeff Pyne, Sysop. Available 24 hours; up- and down-loads, messages, on-line games. Call 902-683-7086.

Reviews

The TRS-80's User's Encyclopedia
(Color Computer and MC-10)
by Gary Phillips and Guier Wright III
The Book Company
11223 South Hindry Avenue
Los Angeles, CA 90045
241 pages
$14.95—8½" × 11" paperback

By Terry Kepner

At first glance I had high hopes for this book; unfortunately, it let me down. The name proclaims it's an *encyclopedia*: as such a combination computer dictionary, software/hardware directory, user group listing, and general purpose resource. It covers practically everything about the Color Computer and MC-10, including their respective Basic commands (with plain English descriptions and examples). And all is organized in alphabetical order. So, how could it let me down?

Trying to review a book like this is difficult; there isn't much to say about the entries, except to note their completeness and accuracy. And with so many entries, it's easier to point out errors than to praise good aspects. I've got to say that I'm unaware of so complete a listing of software and hardware suppliers and their products available anywhere else. The same can be said of the User Group listings. And the dictionary entries are tailored to the Color Computer market, leaving out many terms that have nothing to do with TRS-80 computers.

The disadvantage that cripples this book is that it is very poorly cross-referenced. For example, page 129 lists five magazines that might be of interest to Color Computer owners; it doesn't list *Color Computer News* and gives the wrong name for *Hot CoCo* (calling it "Hot Color Computer" instead). If you look under the heading "Color Computer" (page 44), you find *Color Computer News* listed, but not the *Color Computer Magazine* (listed on page 129).

Another cross-reference error is under the headings about the Forth language: under the entry "Forth," only Frank Hoqq Labs is mentioned; but under "Programming Languages" four of the five Forths are listed; and all five are listed under their individual names (scattered throughout the book in alphabetical order).

Some entries are mysteriously left out; for example, the two hardware products that convert Color Computer serial printer output to parallel printer output are listed, but a description of parallel printers isn't included. And under the entry "Printers" is a list of about a dozen, but by no means is it a complete list of the printers mentioned in the book.

In spite of this, the *Encyclopedia* is a useful reference book that covers a wide field. Careful thought by the user will guide you to most areas in the book where you might find the information you desire. And if you're looking for programs, this is a source worth having.

If the book were adequately cross-referenced, it would be a "must have" item. As it is, I must rate the book as interesting, well worth buying if you have a problem it can solve (such as trying to locate software, user groups, or manufacturers). You should bear in mind as with any reference of this type, many areas are already outdated; any product, user group, and/or manufacturer making an appearance after approximately January 1984 couldn't be listed in this particular reference encyclopedia. *(end)*

Grobot
Children's Computer Workshop
Tandy/Radio Shack Corporation
Fort Worth, Texas 76102
Extended Basic, Joysticks,
Cassette Recorder Req. *$19.95.*

By Dennis Peterson

If you remember, The Children's Computer Workshop budded from Children's Television Workshop, the creators of *Sesame Street.* Grobot, designed for players ten years or older, is a relatively recent bloom from CCW's garden of good ideas. For this age group the manufacturer has its games stress "creative exploration." This is true for Grobot. You should understand that the first purpose of the software is to be a game—it isn't an educational program in a game format.

Most children's software manufacturers softpeddle the violence aspect of computer games. This holds true in Grobot, keeping close to what appears to be the philosophy of CCW. Even what is commonly referred to as the Joystick's "fire" button is referred to as the "red action button" in the very clearly written user's manual.

Play the game: you are a gardener on the strange and wonderful Planet Kaldas, on which 16 plants, with names like Kalidobean, Gaboink and Shishkabud, can live. These plants grow best in certain climates, and coincidentally there just happen to be four distinct climates on Kaldas. In the Kaldasian Farmer's Almanac, located near the end of the user's manual, you'll find the point value for each Plant, a value which varies with the climatic area in which it is grown. For example, a harvested Arroid is worth only 25 points in a hot and dry climate, but its value climbs to a whopping 800 points harvested in a cold and wet climate, where is it more difficult to grow. A few pieces of information are missing in the Almanac for some plants, giving the player a chance to discover some facts on his own.

As gardner, using the information from the Almanac and a weather forecast, you determine which plants to grow and determine strategy to harvest the maximum number of points. To help with your gardening, you control an energetic little gardening robot, called (of course) Grobot, who rides around in a flying-saucerish harvesting, picking plants as they bloom on your command. This isn't as easy as it sounds; the garden, like any real garden, is afflicted by pests. These are the voracious, fast-flying, flower-eating Astro-Gnats, and the slithering, root-munching Verms. In keeping with the non-violence theme, Grobot can be used to scare these Pests from the screen, rather than spray or swat them. You and Grobot will have some very busy moments.

Grobot should capture the interest of most family members, not only because of the actual game play, but because of the pre-play planning. It might be the most refreshing investment in your video game library. *(end)*

DISK DRIVES DISK DRIVES DISK DRIVES DISK DRIVES DISK DRIVES DISK DRIVES DISK DRIVES

PRICE BREAKTHROUGH

Super Sale on New Disk Drives

Starting at ~~$169.00~~! CALL FOR NEW LOW PRICE

Tandon — Siemens — Remex — MPI — Teac — Shugart — Tabor

40 or 80 Tracks — Single or Dual Head — New 3½" Drivette ™

Our Disk Drives are Capable of Single **and** Dual Density Operation

The **NEWEST** Technology Capable of Operating on Most Popular Computers

Drive a Hard Bargain!! ™ **For your TRS/80, Color Computer, IBM, Apple, Franklin, and Max/80 Complete Systems......from $699.95**

Diskette Breakthrough — 10 Pack — ~~$18.95~~ CALL FOR NEW LOW PRICE

PLEASE CALL FOR OUR MOST CURRENT PRICE REDUCTIONS.

TOLL FREE ORDERING 1-800-343-8841

GENERAL AND TECHNICAL 1-617-872-9090

Disk Drives (0123) TRS/80-IBM-Apple-TI-Franklin-Max/80-LNW	◄	
Model I/III/IV Upgrade (Disk Drives - Memory)	◄	CALL
Printers—Daisywheel/Dot Matrix	◄	
Percom Double Density Controller (Model I)	◄	TOLL
Color Computer Printer Interfaces	◄	
Disk Drive Operating Systems	◄	FREE
Repair Services Now Offered—FAST Turn-a-Round	◄	
Apple/Franklin Compatible Add-On Drives with Case & Cable	◄	FOR
Diskettes	◄	NEW
DISK DRIVE CASES AND POWER SUPPLIES (5 yr. warranty)	◄	
Printer **Buffers 8K to 512K**	**starting at $143.95**	PRICES
Holmes Model I/III Speed-up Mod	**starting at $90.00**	
Cables—Printer/Disk Drive	**starting at $16.00**	

Warranty on Disk Drives — 1 Full Year Parts and Labor

SOFTWARE SUPPORT, INC.

One Edgell Road, Framingham, MA 01701 (617) 872-9090 Telex-383425

Hours: Mon. thru Fri. 9:30 am to 5:30 (E.S.T.) Sat. 10 am to 4:30 pm

DEALER INQUIRIES INVITED.

TERMS:
M.C./Visa/Amex and personal checks accepted at no extra charge.
C.O.D., please add $3.00.
Shipping: Please call for amount.
Not responsible for typographical errors.
Prices subject to change.

CANADA
MICRO R.G.S. INC.
751, CARRE VICTORIA, SUITE 403
MONTREAL, QUEBEC, CANADA, H2Y 2J3
Regular Tel. (514) 287-1563
Canadian Toll Free 800-361-5155

Service! Service!
All in stock products are shipped within 24 hours of order. Repair/Warranty service is performed within 24 hours of receipt unless otherwise noted. We accept C.O.D., foreign and APO orders. School and D&B corporate P.O.s accepted.

TRS/80 Registered Trademark Tandy Corp. Apple Registered Trademark Apple Computer Corp
IBM-PC Registered IBM Corp. Franklin Registered Trademark Franklin Corp. Max/80 Registered Trademark Lobo Int.
LDOS Reg. Logical System Inc. Dosplus - Micro Systems Software Newdos/80 - Apparat Inc.

DISK DRIVES DISK DRIVES DISK DRIVES DISK DRIVES DISK DRIVES DISK DRIVES DISK DRIVES

DYNACALC®

CoCo's Best & Fastest Spreadsheet System

ACCLAIMED BY THE EXPERTS

"DYNACALC is my choice for a CoCo spreadsheet."
Dan Downard, RAINBOW, September, 1984.

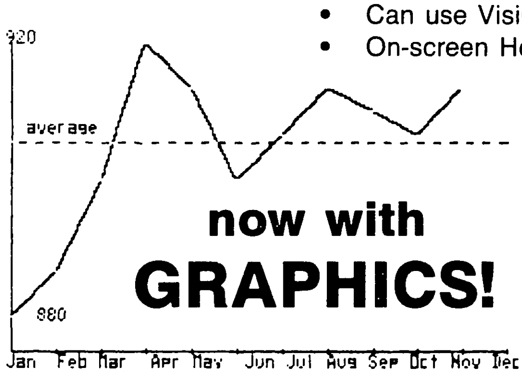
"Eat your heart out, Lotus 1-2-3!"
Scott Norman, HOT CoCo, October, 1984.

Built-in Features:

- 51 x 24 Display with Lower Case
- Super-fast Smart Screen Refresh
- Auto-Repeat Keyboard Driver
- Keysaver (TM) repeats last command x times
- Disk Operating System (works just like ROM DOS)
- Easy communication with BASIC/DOS programs
- Fast 16-Digit Arithmetic with Scientific Functions
- Summation, Mean, and Standard Deviation Functions
- Logical Functions with String & Numeric Comparison
- String locate command to navigate large worksheets
- Sort full or partial worksheet by columns or rows
- Line, Bar, Hi/Lo/Close, Circle Graphs
- Full Graphics captioning and overlay facility
- Graphics Drivers for all popular Printers
- Joystick/Mouse Driver for Cursor Movement
- Works with any ROM versions, even JDOS
- 33k Available Worksheet Space
- Up to 256 Columns or 256 Rows
- Can use VisiCalc worksheets & training materials
- On-screen Help Messages

NOW ONLY $99^{95}

RAINBOW
CERTIFICATION SEAL



now with
GRAPHICS!

FOR 64K DISK SYSTEMS
available from

COMPUTER SYSTEMS CENTER
13461 Olive Blvd.
Chesterfield, MO 63017 USA
(314) 576-5020

CSC

10 to 6 Mon.-Fri.

or your local DYNACALC dealer

Price US$99.95
Outside North America add $10 postage

CANADA- CDN $129.95
RGS MICRO INC.
759, VICTORIA SQUARE 405
MONTREAL H2Y 2J3
TEL.: (514) 287-1563
ORDER LINE ONLY ★ ★ ★
QUEBEC-ONTARIO-MARITIMES
800-361-5338
WESTERN CANADA 800-361-5155

MasterCard
VISA

DYNACALC Reg. U. S. Pat. Off.
Lotus and 1-2-3 are trademarks of Lotus Development Corp.
VisiCalc is a trademark of VisiCorp