ENGINEERING NOTES
on
Radio Shack Color Computers

March 1986
Vol. 3 No. 2

$1.95

DYNACOMP COLOR NEWS

PROGRAMS

* ELECTRONIC BILLBOARD
* DRAW DEMO PROGRAM
* RAMDISK SUBROUTINES

INSTRUCTIONAL SERIES

* LARGE MEMORY PROGRAMS (Part 13
* COMPUTER GRAPHICS (Part 13)
* WRITING PROGRAMS (Part 12)
* INTERFACING COMPUTERS (PART 2)
* BASIC BASIC

SERVICES

* NEW PRODUCTS
* PRODUCT REVIEWS
* QUESTIONS & ANSWERS
* OERATING HINTS

DYNAMIC COLOR NEWS is published monthly by DYNAMIC ELECTRONICS, INC., P.O. Box 896, Hartselle, AL 35640, phone (205) 773-2758. Bill Chapple, President; Dean Chapple, Sec. & Treas.; John Pearson, Ph. D. Consultant; Bob Morgan, Ph. D., Consultant.

 Subscriptions are $15/yr for U.S.A. & Canada, $30 other foreign.

The purpose of this newsletter is to provide instruction on Basic & Machine Language programming, Computer theory, operating techniques, computer expansion, plus provide answers to questions from our subscribers.

The submission of questions, operating hints, and solutions to problems to be published in this newsletter are encouraged. All submissions become the property of Dynamic Electronics if the material is used. We reserve the right to edit all material used and not to use material which we determine is unsuited for publication.

We encourage the submission of Basic and Machine Language Programs as well as articles. All Programs must be well documented so the readers can understand how the program works. We will pay for programs and articles based upon their value to the newsletter. Material sent will not be returned unless return postage is included. Basic & ML programs should be sent on a tape or disk & comments should be sent as a DAT or BIN file.

****
* *
* DYNAMIC COLOR NEWS *
* *
* March 1986 *
* *
* Editor and Publisher *
* Bill Chapple *
* *
* Secretary *
* Dean Chapple *
* *
****

CONTENTS

EDITOR'S COMMENTS

As we begin our third year of publication, we are encouraged by the responses we are receiving. As I reflect back on our first isssue, I can see quite a difference although our editorial philosophy is the same. We are now reviewing products and publishing information on new products. We also have advertisers and editorial and software contributors.

Let's look at the progress that has been made in computer technology. First the cost of computers has dropped considerably. This month Radio Shack has their 16K Color Computer on sale for $69.95. The going price a couple of years ago was around $200. The prices of printers have fallen. For less than $200 a good printer can be obtained.

The prices of integrated circuits have fallen too. Now you can purchase a 256K (32K bytes) EPROM for less than $10. Within this chip the Basic, Extended Basic, Disk Basic, plus 8K more can be contained. The cost of memory chips has also dropped. You can purchase a 256K Ramdisk for less than $100. These were not available 2 years ago.

Expandability with the new color computers is not as easy as it was with the older ones. Most of the sockets have been eliminated. This of course reduced the production costs of the computers, but makes internal hardware upgrades impossible without soldering. There is now more software which will run on all versions.

It is impossible to predict what we will have in two more years. If the present trent continues then there will be many more exciting products. As I have stated previously, there is much more that can be developed for computers and I am looking forward to being a part of these new products.

I want to thank those who have supported us. Without your encouragement and support we would not have been able to continue. The future for Dynamic Color News looks very good and we are looking forward to expanding by adding more editorials and programs.

I am pleased to announce that my new wife Dean will be my secretary. For the past few months I have not had a secretary and this has really been a handicap. She will be keeping things in order which will leave me with much more free time to write and work on programs. Dean and I were married on February 15 and are looking forward to working together. She is experienced with our printing facility and is handling our orders and doing mailing. If you have any problems contact her.

We received a letter from John Gordon Reid. He gave us many suggestion about the newsletter and mentioned subjects he would like for us to cover. We are printing his letter and would like to hear from you. I am starting a file on subjects for DCN and want to know what you would like for us to cover.

INTERFACING COMPUTERS
(PART 2)

Last month we started this new series and asked the following question. How do you get information into and out of your computer? We can print results on our TV screen or an external printer. We can save or load programs to or from a cassette tape or a disk. We can also obtain information from another computer across town or in another city by using a modem or radio link.

A computer can also be used for control applications.

Weather sensors, burglar alarms and the control of electrical appliances and motors are other examples. Software is included within color computers to control the cassette motor with the "MOTOR ON" and "MOTOR OFF" commands. Cassette motor control is accomplished by the built in relay within the computer. A relay is an electrical switch that closes when a current flows through its coil. Relays have contacts that close when the relay is energized. The contacts can apply power to appliances, lights, or motors. For color computers, a relay could be connected in the place of the cassette motor for controlling 110 volt electrical devices.

These are just a few examples of the uses of computers. In this series we want to explain what a color computer can do and give software and hardware examples of how to use it for controlling devices.

ASCII

ASCII is an abreviation of the American Standard Code for Information Interchange. It serves two purposes. The first is to give a numerical value to each keyboard character. Microsoft basic recognizes the ASCII code with the following command.

```
X=ASC(X$)
```

If X$ is the key pressed then X will be the ASCII value of the key. The following program will allow you to observe the ASCII values of the keys.

```
10 INPUT X$: IF X$="" THEN 10
20 X=ASC(X$): ?X;X$
30 GO TO 10
```

The second use for the ASCII code is the serial transmission of data by an RS-232 port. The color computers have an RS-232 port which is normally used for the printer. This port can also be used for exchanging data with another computer either with a direct connection or through a modem.

SERIAL DATA

By serial data we mean that one bit at a time is sent. Computers work on parallel bits. An 8 bit word is called a byte.

A good example of serial data is computer programs stored on a cassette recorder. As you listen to a program being saved or loaded you can hear the changes in pitch. The higher pitch sounds are a "1" and the lower pitch sounds are "0".

SERIAL ASCII BYTE

When dealing with computer related concepts, we generally have two options. A memory bit can have only one of two possibilities.

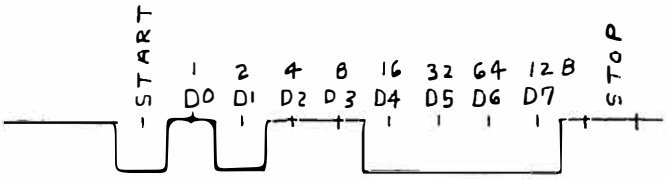
If it is a "0" then it is cleared and if it is a "1" it is set. To simplify this we revert to binary arithmetic where we have only a "0" or "1". If we send ASCII data between two devices in the same physical location, then the "0" and "1" are represented by different voltage levels. A negative voltage represents a "1" and a positive voltage represents a "0". The "1" is called a MARK and the "0" is called a SPACE. How do we know when we are receiving data bits? When no data is being sent the output is in the "1" state. To begin sending data we look for a change from a "1" to a "0". The first "0" bit is called the start pulse. After the start pulse, the 8 data bits follow. At the end of the data bits are one or two stop bits. Stop bits are a "1" and the output can remain at the "1" state until another data byte is to be sent.

A sketch showing several characters is shown in Figure 1. Notice that the least

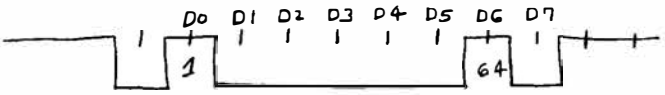
significant bit is the first bit after the start bit and the most significant bit is the last bit.

Next month we will continue looking at the ASCII code. We suggest you study the characters in Figure 1 to gain a better understanding of how serial ASCII works.

+ + +



CARRIAGE RETURN (13)



A (65) 1+64

S(53) 1+4+16+32

SERIAL DATA USING ASCII

FIGURE 1

ELECTRONIC BILLLBOARD

Have you ever wished you could leave messages on your computer? This program allows several lines to be entered and continuously scrolled from the bottom to the top of your TV screen. You can select the color of each line for a varied display. These are large characters and make a very impressive display.

This program is provided by courtesy of T & D Software (advertisement on page 7) and is used by permission.

```
1 REM COPYRIGHT (C) T&D SOFTWARE
 1985  * ELECTRONIC BILLBOARD *
10 PMODE0:GOTO60000
20 CLEAR1000,14499:CLS0:PRINT@23
6,"working";:FORI=14500 TO 15399
:READD:POKEI,D:NEXT
22 FORI=15400 TO 15411:READD:POK
EI,D:NEXT
30 DIM L$(50),CL(50):FORI=1 TO 2
2:READ L$(I),CL(I):NEXT
40 NL=22:GOSUB1000
50 CLS:PRINT:PRINT" ALL YOU NEED
 TO DO IS ENTER      THE LINES YO
U WANT DISPLAYED    AND SELECT T
HEIR COLORS.":PRINT
60 PRINT" YOU MAY ENTER UP TO 50
 LINES.   TO SIGNAL THE END OF E
NTRY,     ENTER '*QUIT*'.":PRINT
70 PT=0
80 LINEINPUT">";I$:IFLEN(I$)>16
THENPRINT" THAT LINE IS TOO LONG.
 MAXIMUM  IS 16 CHARACTERS PER LI
NE.       PLEASE RE-ENTER.":GOTO80
90 IFI$="*QUIT*" THEN200
100 PT=PT+1:L$(PT)=I$
110 PRINT"  gREEN yELLOW bLUE
rED         wHITE cYAN   mAGENTA o
RANGE"
120 K$=INKEY$:IFK$="" THEN120
130 P=INSTR("GYBRWCMO",K$):IFP=0
THEN120
140 CL(PT)=P-1:IFPT<50 THEN80
200 IFPT=0 THENPRINT" YOU DIDN'T
ENTER ANY LINES.":GOTO999
210 NL=PT
215 GOSUB1000
220 CLS:PRINT@230,"B)ACK TO DISPL
```

```
AY                    E)ND PROGRAM"
230 K$=INKEY$:IFK$="B" THEN215
240 IFK$="E" THEN999
250 GOTO230
999 END
1000 PT=1
1010 A$=L$(PT):AD=VARPTR(A$):POK
E277,LEN(A$):POKE278,PEEK(AD+2):
POKE279,PEEK(AD+3):POKE281,CL(PT
)*16+128:Z=LEN(A$):P=(16-Z)/2:PO
KE280,P*2:EXEC 14500
1020 PT=PT+1:IFPT>NL THENPT=1
1030 IFINKEY$="" THEN1010
1040 RETURN
50000 DATA 95,231,141,0,111,141,
86,111,141,0,106,&HBD,&H3C,&H28,
190,1,22,93,39,61
50010 DATA 166,128,52,20,198,8,1
28,32,61,48,141,0,85,48,139,230,
141,0,77,134
50020 DATA 2,61,16,174,133,31,32
,186,1,25,250,1,25,31,2,142,5,22
4,246,1
50030 DATA 24,235,141,0,52,16,17
5,133,108,141,0,45,108,141,0,41,
53,20,90,32
50040 DATA 192,230,141,0,31,92,1
93,4,45,167,141,1,57,142,4,0,236
,136,32,237
50050 DATA 129,140,5,224,45,246,
198,128,231,128,140,6,0,45,249,5
7,0,0,0,0
50060 DATA 0,0,0,0,0,0,0,2,0,10,
0,10,0,2,1,1,5,5,0,0
50070 DATA 0,0,1,1,7,7,5,5,13,13
,0,2,5,14,4,15,4,14,1,1
50080 DATA 0,6,0,10,5,1,0,2,4,6,
5,9,4,7,0,2,0,10,0,0
50090 DATA 0,0,0,1,1,8,5,0,0,9,1
,0,0,9,0,5,1,8,0,0
50100 DATA 1,1,1,9,0,0,0,0,0,10,
4,14,0,8,0,0,0,0,0,0
50110 DATA 0,10,0,0,0,0,4,12,0,0
,0,0,0,0,0,0,1,0,0,1
50120 DATA 0,6,0,10,5,0,0,2,5,5,
5,5,4,6,0,2,4,10,0,10
50130 DATA 1,11,0,2,4,5,0,10,5,3
,0,2,4,5,0,9,4,6,1,0
50140 DATA 5,5,0,13,0,5,1,3,5,0,
4,9,1,6,0,2,5,4,5,9
50150 DATA 4,6,1,3,0,5,0,10,5,0,
0,2,5,5,1,9,4,6,0,2
50160 DATA 5,5,0,13,4,6,0,0,0,0,
0,8,0,8,0,0,0,0,0,8
50170 DATA 0,10,0,0,0,6,4,2,0,4,
0,0,1,3,1,3,0,0,0,0
50180 DATA 4,2,0,6,4,0,0,2,4,5,0
,10,0,2,0,2,4,5,1,13
50190 DATA 4,6,0,2,5,5,5,13,5,5,
1,2,5,5,5,9,5,6,0,2
50200 DATA 5,4,5,0,4,6,1,2,5,5,5
,5,5,6,1,3,5,0,5,8
50210 DATA 5,3,1,3,5,0,5,8,5,0,0
,2,5,4,5,3,4,6,1,1
50220 DATA 5,5,5,13,5,5,1,3,0,10
,0,10,1,11,0,1,0,5,0,5
50230 DATA 4,6,1,1,5,5,5,10,5,5,
1,0,5,0,5,0,5,3,1,1
50240 DATA 5,13,5,5,5,5,1,1,5,7,
5,5,5,5,0,2,5,5,5,5
50250 DATA 4,6,1,2,5,5,5,8,5,0,0
,2,5,5,5,5,4,6,1,2
50260 DATA 5,5,5,9,5,5,0,2,5,4,0
,9,4,6,1,3,0,10,0,10
50270 DATA 0,10,1,1,5,5,5,5,4,6,
1,1,5,5,5,5,0,10,1,1
50280 DATA 5,5,5,5,5,13,1,1,5,5,
1,9,5,5,1,1,5,5,0,10
50290 DATA 0,10,1,3,0,5,1,8,5,3,
1,3,5,0,5,0,5,3,1,0
50300 DATA 4,2,0,10,0,5,1,3,0,5,
0,5,1,7,0,2,4,14,0,10
50310 DATA 0,10,0,0,0,6,4,14,0,4
,0,2,4,5,1,13,4,6,0,0
50320 DATA 0,0,1,13,4,7,1,0,5,0,
5,9,4,6,0,0,0,0,1,12
50330 DATA 4,3,0,1,0,5,1,13,4,7,
0,0,0,0,1,9,5,11,0,2
50340 DATA 5,4,5,8,5,0,0,0,0,0,1
,9,4,7,1,0,5,0,5,9
50350 DATA 5,5,0,0,0,8,0,10,0,10
,0,0,0,4,0,5,0,5,1,0
50360 DATA 5,0,5,1,5,9,1,2,0,10,
0,10,1,11,0,0,0,0,5,7
50370 DATA 5,5,0,0,0,0,1,9,5,5,0
,0,0,0,1,9,4,6,0,0
50380 DATA 0,0,5,9,5,6,0,0,0,0,1
,9,4,7,0,0,0,0,5,6
50390 DATA 5,0,0,0,0,2,4,2,0,6,0
,0,0,2,4,14,0,9,0,0
50400 DATA 0,0,5,5,4,6,0,0,0,0,5
,5,0,10,0,0,0,0,5,5
50410 DATA 5,13,0,0,0,0,4,6,1,9,
0,0,0,0,5,5,4,7,0,0
50420 DATA 1,3,0,6,5,3,1,3,5,0,5
,0,5,3,1,0,4,2,0,10
50430 DATA 0,5,1,3,0,5,0,5,1,7,0
,2,4,14,0,10,0,10,0,0
50440 DATA 0,0,0,0,0,0,48,71,71,
71,71,71,71,71,71,71,71,71,71,71
54000 DATA&H10,&H8E,&H40,&H00
54010 DATA&H31,&H3F,&H26,&HFC
54020 DATA&HF6,&H01,&H15,&H39
55000 DATA*****,1,ELECTRONIC,2,B
ILLBOARD,3,*****,4
55010 DATAPRESS ANY KEY,5,TO EXI
```

OWLS NEST SOFTWARE

' WE GIVE A HOOT '

NEW! CHILDRENS GRAPHIC STORYBOOK SERIES
These delightful children's storys will amuse, amaze and educate your children with words and pictures.

THE STAR LIGHTER presents an entertaining story and illustrates star constellations in graphic displays. Requires 64K EXT Tape or 32K EXT Disk.

THE DRAGON STORY presents a story about a girl and a dragon in words and graphic pictures. Requires 64K EXT tape or 32K EXT Disk.

A VISIT TO THE BEACH presents an easy to read story about two children and their pets on a visit to the beach. Requires 32K EXT Tape or Disk.

The storybooks are $15.00 Tape or $17.00 Disk. Take any two on tape or Disk for $25.00 or all three on Tape or Disk for $30.00

LABEL64 - LABEL64 is a name and address file/print system that takes advantage of your 64K. You can develop and maintain a mailing list. Print lists or mailing labels in your choice of 1, 2, or 3 wide. Supports 3 or 4 line addresses with phone optional. You can sort by last name, first name, and/or zip code. You can work with up to 300 records in memory at a time. We include a second copy for back up at no additional charge. Take advantage of your 64K with LABEL64.
Cassette - 64K EXT Postpaid $24.95

FILE64 - FILE64 is a data management system designed to take advantage of a 64K machine. You can create and maintain records on anything you choose. Recipes, coupons, household records, financial records - you name it. You create records containing up to five fields you define. You can search, sort, modify, add, delete, save on tape, display on the screen and print on a printer. The program could cost you much more and we include a back up copy at no additional charge.
Cassette - 64K EXT Postpaid $24.95

SAVE $$ Take both our LABEL64 and FILE64 for only $40.00 Postpaid. Don't miss this special offer!

ALCATRAZ ADVENTURE Our newest and we think most involved adventure. You have been unjustly imprisioned and sentenced to death. You must escape to prove your innocence. You will face many unique problems as you work on your goal. If you liked our BASHAN adventure you will love ALCATRAZ. Your adventure contains a large vocabulary and some unique features. This is a tough one recommended for advanced players.
32K EXT Postpaid Disk $20.95 Cassette $17.95

CUBE ADVENTURE - Cube is a non violent adventure for a minimum 16K EXT system. You must locate and enter the "CUBE" gathering treasures along the way. You will encounter some unique problems as you work on your goal. CUBE is an intermediate to hard adventure suitable for everyone.
16K EXT postpaid Disk $20.95 Cassette $17.95

In Canada remit to - Kelly Software Dist. LTD.
P.O. Box 11932 Edmonton Alberta T5J 3L1

In Australia remit to - Computer Hut Software 21 Williams St. Bowen QLD 4805

NOW LABELIII IS AVAILABLE ON DISK!

LABELIII - (Reviewed in Nov 83 Rainbow) With LABELIII you can develop and maintain a mailing list. Print lists or mailing labels in your choice of 1, 2, or 3 wide. Supports 3 or 4 line addresses with phone optional. Sort by last name, first name or zip code.
16K EXT Postpaid Disk $21.95 Cassette $19.95

FILEIII - Data management system. With FILEIII you can create and maintain records on anything you choose. Recipes, coupons, household records, financial records - you name it. You create records containing up to five fields that you define. You can search, sort, add, delete, modify, display on the screen or send to a printer. The program is user friendly and user proof. Prompting is extensive. A comparable program could cost you much more. This one is a bargain!
16K EXT Postpaid Disk $21.95 Cassette $19.95

PROGRAM FILE - (rev Oct 83 Rainbow) Organize your programs. With PROGRAM FILE you create a file of your computer programs. You can search, sort, add, modify, delete, save to tape and display on the screen or send to a printer.
16K EXT Postpaid Disk $16.95 Cassette $14.95

DATA MANAGEMENT PACKAGE - Save $$ Take the three above on disk or tape (specify) for only $40.00 Postpaid

ESPIONAGE ISLAND ADVENTURE - (reviewed in June 84 Rainbow) You have been dropped off on an island by submarine. You must recover a top secret microfilm and signal the sub to pick you up. Problems abound in this 32K EXT adventure.
32K EXT Postpaid Disk $20.95 Cassette $17.95

FOUR MILE ISLAND - You are trapped in a disabled nuclear power plant. The reactor is running away. You must bring the reactor to a cold shutdown and prevent the "China Syndrome". Can you save the plant (and yourself)? It's not easy!
16K EXT Postpaid Disk $20.95 Cassette $17.95

KINGDOM OF BASHAN - Our flagship adventure. Bashan has a very large vocabulary and some unique features. You must enter BASHAN (not easy), gather the ten treasures of the ancient kingdom (even harder) and return to the starting point (harder yet). If you can score the maximum 200 points in BASHAN you are an expert!
32K EXT Postpaid Disk $20.95 Cassette $17.95

ADVENTURE COMBO Save $$ The three above adventures on three cassettes or one disk (specify) postpaid for only $40.00

ATLANTIS ADVENTURE - This one is not easy - in fact we challenge you to complete it in 30 days. If you do we will send you any adventure we sell - postpaid - at absolutely no charge. You start on a disabled sub near the lost city of Atlantis. You must get the sub (and yourself) safely to the surface. Do you think Atlantians are friendly?
Postpaid 32K EXT Disk $24.95 16K EXT Cassette $21.95

ADVENTURE STARTER - Learn to play those adventures the painless way. You start with an easy adventure and move to an intermediate. Two complete seperate non violent adventures plus hints and tips on adventuring in general. Finish this and you will be ready for ATLANTIS!
16K EXT Postpaid Disk $20.95 Cassette $17.95

VISA

C.O.D. orders please add 1.50
No Delay For Personal Checks
In a Hurry? Call (615) 238-9458

OWLS NEST SOFTWARE
P O BOX 579c
OOLTEWAH, TN 37363

MasterCard

```
T,6,INSTRUCTIONS,7
55020 DATA*****,1,USE THIS PROGR
AM,2,TO GENERATE,3,COLORFUL LARG
E-,4,LETTER MOVING,5,DISPLAYS.,6
,*****,7
55030 DATAYOU CAN CONTROL,1,THE
COLOR OF,2,EACH LINE.,3,*****,4
55040 DATAWHEN DISPLAYED,5,EACH
 LINE IS,6,AUTOMATICALLY,7,CENT
ERED.,1
60000 PCLEAR1:GOTO20
```

+ + +

WRITING PROGRAMS
(Part 12)

In this series we have been covering the basic commands. We have given programs to demonstrate the principles covered. For those who have followed this series, you should be able to write and debug fairly difficult programs.

Last month we looked at writing seperate files. A seperate file works by storing data in a memory area not used by the normal basic program. The memory below the basic program is reserved by basic for graphics with the PCLEAR extended basic command. By using the PCLEAR command we can quickly reserve lower memory by moving our basic program up in memory. A graphics page is 1535 bytes. You can use PCLEAR N where N is multiplied by 1535 to indicate the amount of memory reserved. If you do not have extended basic then you can move your basic program up in memory by poking values in locations 25 and 26. In fact it is a good idea to peek 25 to see where basic is starting. Non extended basic has a 6 and a 1 in 25 and 26. It therefore starts at 256*6 + 1 or 1537. To simplify this we can use the notation MS,LS. For 6,1 we start at 1537. For 80,1 we start at 20481 or 80 * 256 + 1=20481. This works for basic and extended basic. Basic also requires a 0 before the first statement. So we need to poke a 0 into 20480 if we want the program to start at 80,1. The following program shows the procedure for conditioning the computer for basic to run in a different memory area.

```
 5 'PUT VALUES IN 25 & 26 FOR
   START OF BASIC
10 POKE 25,80: POKE 26,1
15 'PUT A 0 IN BYTE BEFORE
   FIRST BASIC COMMAND
20 POKE 80*256 ,0
30 NEW
```

You do not have to have a program. The commands can be entered directly from the keyboard. The program is given to demonstrate the procedure.

Last month we gave a program for writing characters to a file and saving the file to a cassette or disk. We considered the file as a machine language program. Let's discuss machine language programs since we have not covered them in this series.

MACHINE LANGUAGE

Machine language programs consists of codes recognized by the microprocessor. This is the fastest mode of operation for the computer. Basic commands have to be translated into machine language codes. This is automatically done by the basic software and is the reason for the slower speed of the computer. You might not think it is slow when you list a program and it is quickly displayed on the screen. However a machine language program to list you program would display it so fast that all you would see would be a flash.

WRITING MACHINE CODES

Learning to write machine codes is a study in itself. You can start by looking at the mi-

croprocessor's data sheet. Machine language codes can be assembled by hand by looking up the codes and poking them into memory. This is awkward and time consuming.

To make life easier you can use an assembler. With an assembler you enter mnemonics or symbols for the codes. This is similar to basic in that you write instructions using mnemonics or symbols. Then the assembler converts the mnemonics to machine code.

MACHINE LANGUAGE & BASIC

Machine language subroutines can be called from basic. If you have a basic operation that you want to perform that is slow then you can call a machine language subroutine to perform the operation. Last month we showed how a seperate file could be generated. Suppose we want to add something to the middle of the file. Then we need to make room for the new information by moving data. It is easy to write a basic subroutine to move data in memory. If you have a lot of data then basic is too slow. However a machine language subroutine can quickly move the data.

Another example is searching for data. To do this with basic would be very time consuming. A machine language subroutine can quickly search memory for a match. The machine language subroutine can be called from basic by using the EXEC command.

EXEC COMMAND

The EXEC command calls a machine language subroutine from basic. We are not going into detail on writing machine language subroutines now, but we do want to show how to use them. We will give you the machine codes for the subroutines we will be using in our example programs so that you can use them with your programs. Machine codes can be carried in data statements and poked into memory. Suppose we have a machine language subroutine at 500 then we can call this subroutine from basic by entering the following command:

```
EXEC 500
```

The EXEC command does not return a variable. Variables need to be stored or poked into memory before using EXEC. For example suppose we need to search our file for a word string with a maximum of 8 characters. The number of characters could be placed in 500. Locations 501 to 508 could contain the character string. Then the machine language subroutine will be designed to use the same memory for its data when it does the memory match search. If 509 and 510 is to contain the memory location of the match, then when the machine language subroutine finds a match it can put the memory location for the match in 509 and 510. The machine language subroutine can return to basic after the match is completed. From basic we can use the peek command and look at the values in 509 and 510 to determine where the match occurred.

EXEC EXAMPLES

You can provide a hard reset for your computer by entering the following:

```
POKE 113,0: EXEC 40999
```

To tell what version of basic you have do the following:

```
EXEC 41175
```

Next month we will continue with our file examples. There are a lot of applications where seperate files are desireable. Word processors, spread sheets, check books are a few examples.

BASIC BASIC

We welcome questions and will print some of general interest. Time will not permit individual replys. Questions or tips may be sent to Norman R. Shelton C/O Owls Nest Software P.O. Box 579 Ooltewah, TN 37363. O.K. here goes -

Many of the questions we receive concern the "Baud Rate" and what it does. The BAUD rate is the rate that the computer outputs it's data. Sort of like giving the computer a speed limit. This feature is provided to make sure the computer does not operate too fast for the device it is "talking" to. Memory locations 149 and 150 contain the BAUD rate information. The computer on power up is set at 600 BAUD. You can verify this by "PEEKING" into the computer's memory. All information used by the computer is stored as numbers. (Even letters are given a number by the computer) Each number is stored in a different location. (A memory map provides this data.) If you know the location you can ask the computer to tell you what it has stored there with the PEEK command. Power up your computer and type - ?PEEK(149) <ENTER> (You get a 0 because nothing is there). Now type - ?PEEK(150) <ENTER> You should get an 88. Whenever the computer prints data it must go to locations 149 and 150 and wait the length of time specified by the numbers stored there. Think of it as if the computer must count to 88 between each bit of data it is sending. This is to allow the device that is receiving data time to digest the data and not get overloaded or fall behind. Now lets say your printer is capable of handling 9600 BAUD. If your computer is sending at only 600 BAUD and the printer is set for 600 BAUD everything will print O.K. (Like driving 40MPH in a 55MPH zone). The only problem is there is a lot of time wasted while the computer does it's counting and the printer is simply waiting. That is where the counterpart of PEEK comes in. Some of the memory locations can be changed with the POKE command. What you are doing with POKE is saying "put (the number you POKE) in location (the number you specify). The Syntax for POKE is - POKExxx,yyy <ENTER> where xxx is the location in memory you want to store a number and yyy is the number you want to store. Now lets type - POKE150,1 <ENTER> You have just changed your computer BAUD rate to 9600. If you have a printer that can handle 9600 BAUD set it for 9600 and try running a program listing. You will be amazed how much faster it is. Now instead of counting to 88 the computer only counted to 1 between sending bits of data. You can also verify the POKE worked by typing - ?PEEK(150) <ENTER>. On the reverse side you can slow down the data by POKING higher numbers. For example a 4 at location 149 and an 88 at location 150 gives you 50 BAUD. The computer starts at location 149, counts one, goes to location 150 counts to 88, returns to 149 for 2, back to 150 for another 88, back to 149 for 3, back to 150 for 88, back to 149 for 4, finally to 150 for 88 and at last sends it's next bit of data. If you try to send data too fast the printer will not be able to handle it. You may get nothing or you may get garbage. You will not hurt your printer but it may just sit there and say "WHAT WAS THAT?" instead of printing your data. The BAUD rate table in Radio Shack's manual is difficult to understand at best and is very limited. The following will give you BAUD rates most commonly used.

BAUD RATE	POKE 149	POKE 150
50	4	88
300	0	180
600	0	88
1200	0	40
1800	0	25
2400	0	18
3600	0	10
4800	0	7
7200	0	3
9600	0	1

If you haven't got a printer and want to try a PEEK and POKE then type - ?PEEK(282) <ENTER>. You should get a 255. Now type POKE282,0 <ENTER>. Hit a few keys and you will see the POKE turned on the lowercase. You can either use the SHIFT/0 to switch back or type - POKE282,255 <ENTER>. When there is a 1 at location 282 the computer uses uppercase. When there is a zero it uses lowercase.

You may have noticed when I had you "PEEK" into memory I had you use a "?" where the normal syntax would be "PRINT". The computer will accept a "?" in place of "PRINT". This little short cut can save you time. Try this short program -

```
10 CLS:?@64,"TEST"
```

Run the program. It works O.K. - now type - LIST <ENTER>. The computer changed the ? to PRINT. This will work in direct statements or program listings. You probably know you can use the "'" in place of REM or REMARK.

Lets look at another programming shortcut. Enter this short program -

```
10 CLS
15 ?@32,"TEST LINE ONE"
20 ?"TEST LINE TWO"
```

Now RUN the program. It works O.K. but lets shorten it. Type - NEW <ENTER> to erase the old program. Enter this new program -

```
10 CLS:?@32,"TEST LINE
ONE",,"TEST LINE TWO"
```

Run this and you have done the same thing but used less memory. Change line 10 to read -

```
10 CLS:?@32,"TEST LINE
ONE",,,,,,,,"TEST LINE TWO"
```

Now run the program. As you can see the extra commas spaced the printing down the screen. This can be a little tricky to use but in certain applications can save a lot of time. Print statments can be seperated by commas. The computer divides the screen into two columns of 16 characters each. The comma tells it to move to the next available column. The tricky part is if the first string printed was 16 characters or less the computer would move to the center of the screen for the next column. If the string printed was over 16 characters it would move to the next line.

Another character that can be inserted into a print line is the semi-colon. Type - NEW <ENTER> to erase the old program and enter this line.

```
10 CLS3:?"This is the first
half";
```

Run it and you can't see what the semi-colon did, however it left the cursor at the very end of the printed string. Add this program line -

```
20 ?" and this is the rest"
```

Run the program. As you can see the semi-colon left the cursor where it finished printing the first string and the next line started printing where the first left off. Delete the semi-colon from line 10 and run the program to see what you get without it. As you can see both the comma and semi-colon can be very usefull when you want to print data on the screen.

In our next column we will go further with the various screen printing methods and tips including ways to add graphic characters into our printed lines.

Until next month - - - -

COMPUTER GRAPHICS

(PART 13)

SCALLING

In this series we have been covering the extended color graphics commands. With these powerful commands almost any image or picture can be drawn on the screen. These pictures can be saved to a disk or cassette and later recalled. We can draw pictures by making memory pokes. When we began this series, we showed how the bytes were arranged in memory for the different graphics modes. In fact our character generator worked by poking values into memory. This was slow because basic had to assemble the bits before doing the memory pokes.

The draw commands are collections of machine language subroutines that are called from the basic and extended basic software. These are much easier to use than doing memory pokes.

LINE - DRAW COMMANDS

Before proceeding to new material let's review and compare the LINE and DRAW commands. The line command requires two points designated as (X1, Y1) and (X2, Y2). The line can be drawn between these two points. Also the command can be expanded to draw a rectangle or box with the two points as opposite vertices. This is similar to using rectangular coordinates in algebra.

```
LINE (X1,Y1) - (X2,Y2) , PSET, B
```

The preceeding command will draw a box outline with the points (X1,Y1) and (X2,Y2) at opposite angles. If we replace the B with BF then we will have a filled box.

The draw command requires a direction and length. Several draw directions may be combined into one draw command. The

letters that designate the draw directions are easy to remember. R=right, U=up, L=left, and D=down.

Four more directions starting at 45 degrees and with 90 degree separation are defined by E(45), F(135) ,G(225), and H(315).

STRINGS

A string is a series of characters. These characters can be contained within a string variable which can be composed of 1 or two letters followered by a $ sign. Examples are X$, WA$, etc. Since the draw commands use strings, we can make up a seperate string with our draw information. For example A$= "U5; L8; D5; R8" will draw a box. Now whenever you need to draw this box just enter DRAW "A$".

SCALING

Scaling is used to increase or decrease the size of an object generated by the draw command. With scaling we can write the draw command for the object and make it larger by scaling. Since the draw commands require strings, we will need to convert a number to a string. This can be done by the STR$ command. If we have a variable X, we can convert it to a string by the following command:

```
X$=STR$(X)
```

Remember that we can do numerical operations with variables such as X. Then convert the variable to a string to indicate a length for the draw command. We are familiar with FOR - NEXT loops. These can be used if we want to continuously draw a figure and have it change in size. Suppose we just want to define a figure by a string variable such as A$. If A$ can draw the same figure with different sizes then we have implemented scaling.

BOX EXAMPLE

A box is a simple figure and we will use it to demonstrate scalling. If H$ represents the height or vertical length and W$ represents the width or horizontal length, then the string to represent the box is:

```
A$="U"+H$+"L"+W$+"D"+H$+"R"+W$
```

If H$="20" and W$="40 we would have:

```
A$="U20"+"L40"+D20"+"R40.
```

Now the next thing we must do is define W$ and H$. If we let S represent the scale factor then we can write W = 4 * S and H=2 * S for a box with the width twice as much as the height. Now to find the strings H$ and W$ we can write H$ = STR$(H) and W$ = STR$(W).

DRAW PROGRAM
(DEMONSTRATING SCALING)

The following program demonstrates scaling by drawing a box of different sizes. The first part continuously draws boxes of increasing sizes. The second part allows you to enter a scale factor for the box.

```
10 PRINT"DRAW PROGRAM TO
   DEMONSTRATE SCALING
20 PRINT"cOPYRIGHT (c) 1986
30 PRINT"dYNAMIC eLECTRONICS
   iNC
40 PRINT"PROGRAM 3-1-86
50 PRINT"THIS DRAWS BOXES OF
   DIFFERENT":PRINT"SIZES. THIS
   STARTS IN THE RIGHT HAND
   CORNER OF THE SCREEN.
60 FOR K=1 TO 800: NEXT K
70 'SET UP FOR GRAPHICS
80 PMODE 3,1:SCREEN 1,0:PCLS
90 'DRAW BOXES WITH INCREASING
    SCALE FACTOR
```

```
100 FOR S=2 TO 50
110 GO SUB 180
120 NEXT S
130 PRINT"THIS ALLOWS YOU TO
    DRAM":PRINT"INDIVIDUAL
    BOXES WITH A WIDTH TO
    HEIGHT RATIO OF 2 TO  1.
    ENTER THE SCALE FACTOR S
    FROM 1 TO 50. AFTER THE BOX
    IS DRAWN THEN PRESS A KEY
    TO RETURN HERE.
140 INPUT S: PCLS
150 GO SUB 180
160 D$=INKEY$:IF D$=""THEN 160
170 GO TO 130
180 'MOVE TO RIGHT HAND BOTTOM
     CORNER
190 'THIS IS THE DRAW
     SUBROUTINE
200 PMODE 3,1: SCREEN 1,0
220 DRAW "BM255,191;
230 W=4*S: H=2*S: W$=STR$(W):
    H$=STR$(H)
240 FOR K=1 TO 100:NEXT K
250 A$ ="U" + H$+ "L"+W$+ "D"
    + H$+ "R" + W$
260 DRAW A$
270 RETURN
```


LARGE MEMORY PROGRAMS

(PART 13)

RAMDISK IMPROVEMENTS

In this series we have been showing how to use the second 32K memory bank in 64K computers. The principles covered here will also apply to computers with larger memories. In our NOV/DEC issue we presented a RAMDISK program that allows programs to be saved in the second memory bank. In January and February we gave some improvements to the program. Last month we showed how to delete a program from the menu and move programs in the second bank to free extra memory.

This month we want to show how to handle machine language programs. Machine language programs are a little more complex than basic. Basic programs require beginning and ending pointers or vectors. These are in locations 25-28. Machine language programs require three vectors which are the beginning, ending, and execution addresses. These are in 487-8, 126-7, and 157-8 respectively.

For our program directory we allowed 20 bytes for each program. It occupied memory below the basic program in the first bank. The directory was organized as follows:

BYTES	INFORMATION
0-7	Program Name
8	0-BAS, 1-BIN
9-10	Beginning of PGM
11-12	Ending of PGM
13-14	Beginning of ML PGM
15-16	EXEC. add. of ML PGM

The vectors in bytes 9 and 11 contain the beginning and ending location of the program in the second bank. The vector in byte

13 shows where the machine language program goes in the first bank. The vector in byte 15 contains the execution address of the machine language program.

Handling machine language programs requires two modifications to our RAMDISK program. We have to modifiy our save a program and our load a program subroutines. The original RAMDISK program was printed in our NOV/ DEC issue and these modifications will apply to it. The subroutine for saving a program was location in lines 2000-2999 and the subroutine for loading a program was in lines 6000-. We are including these subroutines in the next section with comments. Remember the RAMDISK requires machine language subroutines which were presented in an earlier issue.

This finishes our discussion on the RAMDISK. We have some more information and applications for using the second 32K memory bank and will start on them next month.

RAMDISK SUBROUTINES

```
10 'PGM 3-2-86
20 'cOPYRIGHT (c) 1986
30 'dYNAMIC eLECTRONICS iNC
40 'RAMDISK SUBROUTINES
50'
2000 PRINT"THIS SAVES A PROGRAM
2010 PRINT"PRESS B FOR BASIC OR
     M FOR ML":GO SUB 950
2012 'GET PROGRAM NUMBER
2015 PN = PEEK(32000): PX=S+20
     * PN+8
2017 IF P$="M" THEN GO SUB
     2200:GO TO 2025
2020 IF P$="B" THEN GO SUB 2500
2025 EXEC 32090
2026 '2025 PUTS THE PGM IN BK 2

2030 GO SUB 2900
2035 'INCREASE PN IN 32000
2040 POKE 32000, PN + 1:GO
     SUB 1000
2050 RETURN
2200 PRINT"MACHINE LANGUAGE
     PROGRAM
2205 INPUT "BEGINNING"; BE:
     INPUT "ENDING"; EN: INPUT
     "EXECUTION ADDRESS";EX
2209 'THE FOLLOWING PUTS BE,
      EN, AND EX ADD IN PGM
      DIRECTORY
2210 M=474:V=BE:GO SUB 970:POKE
     PX + 5,MS: POKE PX+6, LS:
     V=EN:GO SUB 970
2220 M=PX+7: V=EX: GO SUB 970:
     POKEPX,1:PX=PX+1
2230 GO TO 2540
2500 PRINT"THIS HANDLES BASIC
     PROGRAMS
2510 POKE PX,0:PX=PX+1
2520 FOR J=0 TO 3
2530 A=PEEK(32165+J):POKE 474 +
     J,A :NEXT J
2540 M=32001:GO SUB 990:GO SUB
     980: POKE 478,MS: POKE
     479, LS:NL=V
2542 GO SUB 2590
2545 'CALCULATE LENGTH OF PGM
2550 X1=PEEK(476)-PEEK(474):X2
     =PEEK(477)-PEEK(475): PL=
     256*X1+X2
2552 'CHECK TO SEE IF PGM TOO
      LARGE
2555 N=0:GO SUB 1012:SA=G-PL:
     IF SA<0 THEN PRINT
     "PROGRAN TOO LARGE BY
     "SA"BYTES":RUN
2558 'UPDATE END OF RAM VECTOR
2560 V=1+PL+NL: GO SUB 980:
     POKE32001,MS:POKE32002,LS
2590 POKE PX,MS
2595 PX=PX+1:POKEPX,LS
2596 PX=PX+1
2600 RETURN
2900 INPUT"PROGRAM NAME";N$
2905 A=LEN(N$):B=PEEK(32000)
2910 M=S+20*(B)
2912 'PUT PGM NAME IN DIR
2915 FOR J=0 TO 7:POKEC(M+J),
     32:NEXT J
2920 FOR J=1 TO A:X$=MID$ (N$,
     J,1): X=ASC(X$): POKE
     M+J-1,X
2927 NEXT J
2930 RETURN
2940 '
6000 PRINT"THIS LOADS A PROGRAM
6010 GO SUB 1000
6020 PRINT"PRESS NUMBER FOR
     DESIRED PROGRAM":GO SUB
     950:Q=VAL(P$)
6030 Q=Q-1:Z=S+20*Q+9
6040 FOR J=0 TO 3:A=PEEK(Z+J):
```

```
     POKE 474+J,A:NEXT J
6042 'CALCULATE PGM LENGTH
6045 X1=PEEK(476)-PEEK(474):
     X2=PEEK(477)-PEEK(475):
     PL=256*X1+X2
6046 IF PL=0 THEN PRINT "NO
     PROGRAM- PRESS A KEY.": GO
     SUB 950:RUN
6047 IF PEEK(Z-1)>0 THEN 6200
6048 'LOAD PGM INTO AREA INDI-
      CATED BY VECTOR IN 32165
6050 A=PEEK(32165):B=PEEK
     (32166): POKE478,A:
     POKE479,B
6060 'PUT END OF PGM VECTOR
      IN 32167
6070 PB=256*A+B:V=PB+PL:GO SUB
     980: POKE32167,MS:
     POKE32168,LS
6080 PRINT"PROGRAM IS LOADED
     AND RUNNING
6085 '32115 MOVES DATA AND
      32170 EXCHANGES VECTOR
      AND RUNS THE PROGRAM
6090 EXEC 32115:EXEC 32170
6100 '
6150 'THIS SETS UP BE, EN, EX
      VECTORS FOR ML PGM
6200 M=Z+4:GO SUB 990:A=PEEK(M)
     :B=PEEK(M+1): POKE478,A:
     POKE479,B: POKE487,A:
     POKE488,B
6210 EP=V+PL:V=EP-1:GO SUB 980:
     POKE126,MS :POKE127,LS
     'PUT END OF PROGRAM VECTOR
      IN 126
6220 A=PEEK(Z+6): B=PEEK(Z+7):
     POKE 157,A:POKE158,B
6230 EXEC 32115 'LOAD ML PGM
6240 PRINT"THE ML PGM IS NOW
     LOADED. TO RUN IT ENTER
     EXEC. TO RETURN TO THIS
     MENU TYPE <RUN>.
```


PRODUCT REVIEWS

This section is open to all producers and dealers of color computer products. We will review you product free of charge and write an editorial on the product. We do not use a rating system but will explain what the product does, and what can be expected from it. Any comments about the review from the firm submitting the product will be printed in a later issue.

* * * * * * * * * * * * * * * * *

ASSEMBLY LANGUAGE PROGRAMMING

A new book by Laurence A. Tepolt is available for those who want to learn assembly language programming. An assembler is a tool for writing machine language programs. One of the problems with assemblers is understanding how to use them.

The book is written for color computer users and on a level that is understandable to a nontechnical person. Also included is information about the operation of color computers. The 6809E microprocessor's architecture is discussed in detail. Each of the machine codes are defined plus a discussion on stacks, subroutines, addressing modes, plus much more.

A section is included on assembly language programming using EDTASM+. Included are the text editor, assembler, and ZBUG debugging aids.

The chapter devoted to assembly language programming has numerous program examples. Subroutines, stacks, and interrupts are covered as well as examples on how to organized and develop your program.

There is a chapter on using assembly programs with basic. These can be called with the EXEC and USR commands. Also examples are given for using machine language surroutines resident in the basic, extended basic, and disk ROMS.

A chapter is devoted to internal control and graphics. This includes the Synchronous Address Multiplexer (SAM), Peripheral Interface Adapters (PIA) and Videw Display Generator (VDG).

The last chapter covers technical details. It explains how sound is generated, how the keyboard matrix works, the joystick ports, D to A converter, cassette operation, and the cartridge port.

We found the book to be very informative, useful and complete. The examples on assembled programs were easy to follow. If you have an assembler and are having trouble with it, then the examples should be helpful. It is written in a style that is easy to read and understand. Since there is much material on the internal structure or architecture of color computers, the book also makes an excellent reference book.

TEPCO, 30 Water St., Portsmouth, RI 02871, $16 +$1.50 S/H.

+ + + DCN STAFF + + +

LETTERS

The following letter was sent to us by John Gordon Reid. He had many suggestions and we thought it would be of interest to our readers. Would you like for us to cover some of the subjects he mentioned? We usually print the answers to questions asked in letters in our Question and Answer section. We are considering adding additional editorial subjects and your letters will help us decide which subjects will be of the most interest to our readers. So if you have a subject you want us to cover, please write a letter and tell us about it. We will publish letters that we feel are of interest to our readers.

- Editor -

Dynamic Color News
Dynamic Electronics Inc.
P.O.Box 896
Hartselle, Al 35640-0896

To the DCN Staff:

I have just subscribed and bought all back issues of your newsletter...YUM..YUM.

I am using the new Telepatch. I must strongly dissagree with your review. First the reset button brings Telewriter back to EDIT. And second, all the tape functions are there but the menu is blank. Go to the part of the menu that contains (D)isk and you will see blank spaces above and below. Enter (R) or (S) etc and the tape functions will operate. I love it. Perhaps you have an earlier version.

I would like to know whether the other enhacements like Wizard etc will work on top of Telepatch.

Here are some things I would like you to cover.

(1) An eprom vdg chip that would give a choice of 51 or default c/line.

(2) An internal seperate ram spooler that will work with TW64.

(3) Projects on expanding the eproms in disk controllers. I would like to have R/S 1.0, 1.1, Ados, Spectrom dos, no dos, and more. I can think of about 8 doses.

(4) A run down on the varies hard disk drives.

(5) More info about combining Basic and Extended Basic on one chip and installing an eprom in the empty socket. Like disk to disk to tape programs.

(6) A project to hardware (eprom?) program the four programed keys on the newer boards.

I would also like to be able to increase the buffer in my Epson serial board. I have 2K and would like to have 8K. Also I am confused as to what Grafrax chip to use in my board, Graftrax 80?,Graftrax+. Also does a reference card exist for the Epson printers. I understand that you have an Epson printer so that is why I ask.

Anyway I love the hardware projects and ML instruction for those of us who are thick. I have had a CoCo since Feb'81 (D board) and still get confused with ML.

I know I have left a lot of suggestions and questions and though I can't afford to pay the ten dollar fee for answers I hope you will answer a few each month in 'letters'. I will leave msgs (ads) on all the BBS's of which I am addicted.

Thanks again for much enjoyment and knowledge.

Bye for now

John Gordon Reid

OPERATING HINTS

Disk Programs - You can quickly remove disk programs from a disk by typing "DIR" to display the programs. Then chain kill commands for the programs you don't want. Example: KILL "FIRST/BAS" :KILL"PGM /BIN": KILL"LAST/DAT <ENTER>. This saves having to type DIR after deleting each program.

Tape Programs - You can check tape programs for errors by using the SKIPF command. Load the tape and rewind it. Then type SKIPF"X where X is a file that is not on the tape. Then start the tape and the name of each file will be displayed on the screen as it is found on the tape. If there is an error the tape player will stop and an error message will appear on the screen. All programs before the player stopped are good. If the tape reaches the end with no errors then the tape is good.

QUESTIONS & ANSWERS

Question: You mentioned using remarks for data. How is this done?

Answer: We covered this subject in detail in our first few editions. The check book program in our DCN-2 collection of programs uses remarks for data. The advantage of this method is that you can say "Go to statement #X for your data". With read and data statements you have to take the data in sequence.

Question: I am trying to run the assembly programs in the "large Program" series and am confused by programs such as :4076 STA E 65493 'STORE A IN 65493. What is the "E" in there for? Can you operate with these lines, as written, in EDTASM or MACRO C?

Answer: The "E" stands for extended addressing. We use our decimal assembler "DISASM" for writing our machine language programs. We use the following letters for addressing modes:

E - Extended I - Immediate
N - Indexed D - Direct

4076 STA E 65493 means at memory location 4076 we have the command to store in 65493 the value in A. We covered this in our earlier editions. If there is enough interest we will be glad to start a series on assembly language programming. If you have an assembler, we suggest you enter our machine codes by using pokes and then use your

disassembler if you are having trouble with our notation.

Question: In your review of Telepatch you said that the reset did not work. The reset works fine for me.

Answer: We had several people to point this out to us. Our version does not reset and it may be that we have an earlier version of Telewriter. Anyway we are glad that it does work. We can only report what we observe.

CLASSIFIED ADS

1. 10 cents a word, $3 minimum.
2. Name, Address, & Telephone listed free.
3. Send payment with ad.
4. Closing date 1st of the preceeding month. Ex. Nov ad closing is Oct. 1.

* * * * * * * * * * * * * * * * * *

DISPLAY ADS

(Rate sheet 2 - March 1986)

Closing 1st of preceeding month.

Pages	1 time	2 times	3 times
*2	25	23	22
1	30	27	25
1/2	23	20	18
1/4	15	13	12

* We can use colored paper at no extra charge if ads are on both sides.

We can do ads in Red, Blue, or Brown. No all one color ads will be accepted. For color ads send artwork for each color. Add 40% for each color. Example: One page black and red for 3 times costs $25 + 10.00 = $35.00 each month.


```
+ ++ + ++ + ++ ++ ++ + ++ + ++ +
+                               +
+        RENEWAL TIME?          +
+                               +
+ The date beside your name on +
+ the address label indcates +
+ the last issue you will re- +
+ ceive. Send in your renewal +
+ if you want to continue re- +
+ ceiving technical informa- +
+ tion on Color Computers. +
+ This is the last issue for +
+ those with 3/86.          +
+                               +
+ + ++ + ++ + ++ ++ ++ + ++ + ++
```

NEW PRODUCTS

This section is available free for producers and dealers of color computer products. These products have not been reviewed by us but are included for our reader's information. Send a description of new products to:

New Products
Dynamic Electronics Inc.
P. O. Box 896
Hartselle, AL 35640

The Zellerbach Group has a new catalog on color computer products. Included are Software Programs, Technical Manuals, Hardware Equipment, and Computer Supplies. A "Z-RING" is a helix coil that twists into the pin-feed holes of printer paper to form a rigid binding. The Z-RING costs $.50 each or $5.00 a dozen. For more information contact the Zellerback Group, 1335 Pacific Ave. Unit #216, San Francisco, CA 94109 (415) 673-3485.

****
* Please sign me up for one year for the DYNAMIC COLOR NEWS SERVICE. I *
* Want to receive instruction on programming, Computer Theory, Operat- *
* ing Techniques, Computer Expansion plus the Individual Reply to my *
* Computer problems for a special of $10 each. Also I understand that *
* there will be no charge for letters printed with answers in the *
* Newsletter. Cost $15 USA & Canada, $30 foreign. *
* *
* Name _____ Mail payment to *
* Address _____ Dynamic Electronics Inc *
* City _____ P. O. Box 896 *
* State & Zip _____ Hartselle, AL 35640 *
* Enclosed is a check ___ *
* charge to VISA ___ MC ___ Number _____Exp._____ *
* *
****

DYNAMIC ELECTRONICS INC.
P. O. Box 896 (205) 773-2758
Hartselle, AL 35640

FIRST CLASS