ENGINEERING NOTES
on
Radio Shack Color Computers

May 1986
Vol. 3 No. 4

$1.95

```
****    *   *   *   *     *     *   *    ***     ***
*   *   *   *   *   *    * *    ** **     *     *   *
*   *    * *    **  *   *   *   * * *     *     *
*   *     *     * * *   *****   * * *     *     *
*   *     *     *  **   *   *   *   *     *     *
*   *     *     *   *   *   *   *   *     *     *   *
****      *     *   *   *   *   *   *    ***     ***


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


            *   *   *****   *   *    ***
            *   *   *       *   *   *   *
            **  *   *       *   *   *
            * * *   ***     * * *    ***
            *  **   *       * * *       *
            *   *   *       ** **   *   *
            *   *   *****   *   *    ***
```

PROGRAMS

* ROULETTE (Game)
* GRAPHICS DRAW PROGRAM
* RESTORE BASIC PROGRAMS

INSTRUCTIONAL SERIES

* LARGE MEMORY PROGRAMS (Final)
* COMPUTER GRAPHICS (Part 15)
* WRITING PROGRAMS (Part 14)
* INTERFACING COMPUTERS (PART 4)
* MACHINE LANGUAGE PROGRAMMING
  (NEW SERIES)

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 3564Ø, phone (2Ø5) 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, $3Ø 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    *
*                             *
*          May 1986           *
*                             *
*    Editor and Publisher     *
*        Bill Chapple         *
*                             *
*          Secretary          *
*         Dean Chapple        *
*                             *
*******************************
```

CONTENTS

256K & 512K MEMORY UPGRADES

If you have a 64K computer with sockets for the SAM and 4164 chips then you can update it to 256K or 512K. The ramdisk allows programs to be retain within your computer and loaded as needed. Features include:

* 4Ø Track Single Disk Swap Can serves as second drive.
* Fast 35/4Ø Track Ramdisk (2 Ramdisks with 512K).
* 32K to 2ØØK printer spooler (4ØØK with 512K RAM).
* More then 3Ø PMODE 4 screens at once.
* Pager configures computer for 8 (16 with 512K) 32K pages.
* OS-9 Ram Disk 35-4Ø track single sided or 4Ø track double sided with 512K.
* Memory is protected when the computer is reset.
* Solderless installation.
* Miniature toggle switch can force computer into 64K mode.
* Compatible with all software.

Software is supplied on tape or disk execept OS-9 is not available on tape. Specify your choice when ordering. Assemblies are complete ready to install with memories and 64K mode switch. Order ME-16 for 256K assembly, ME-14B provides extra 256K for ME-16. ME-16A for 512K assembly.

ME-16	- 256K RAM	$89.95
ME-14B	- Second 256K for ME-16	69.95
ME-16A	- 512K RAM	159.95

128K UPGRADES

ME-1ØA Upgrades 64K Korean Computers to 128K. $49.95

ME-12 - Upgrades all 64K computers with 4164 memory chips to 128K. $49.95

VIDEO REVERSER

An integrated circuit that mounts on the 6847 and reverses the video reducing eye strain. Minor soldering for CC-2. $9.95

MEMORY SAVER
(Uninterrupted Power Source)

Our UPS saves your programs from being lost due to power failures by providing power to the memories from its battery. The assembly consists of a control circuit, battery, miniature toggle switch and a light emitting diode (LED). The control circuit and battery mount under the keyboard or can be mounted outside. The switch enables the UPS and the LED glows when power is available. For all computers with 5 volt memories. $59.95

MEMORY MANAGER
(New Product)

A complete set of software for managing the second 32K memory bank for 64K and larger computers. Run Basic programs in both banks, continue a basic program from one bank to the other, use the second bank for a RAM DISK, configure the computer for the all RAM mode and store programs in the upper memory. $27.95 cassette, $29.95 disk.

24 hr phone. Checks, VISA & MC cards. Add $3 ship.

DYNAMIC ELECTRONICS INC.
P. O. Box 896 (2Ø5) 773-2758
HARTSELLE, AL 3564Ø

WRITING PROGRAMS
(Part 14)

In this series we have been explaining how to write basic programs. For the last few months we have been looking at creating seperate files. This can get very involved but is a good application for the programming material we have covered. We want to continue with this, but first let's take some new material.

EDITING BASIC PROGRAMS

The extended basic edit command is very powerful. We want to show how to use this for editing our basic statements. We will cover a few of these each month. Extended basic is required for using the edit feature.

INSERT & DELETE

These two commands will almost handle the edit requirements. Let's take an example and show how to use them.

```
 10 PRINT"THIS S A XCOMPUTER."
```

We want to add an I before the S to form the word "IS" and eliminate the X preceeding COMPUTER. To get into the edit mode type "EDIT 10 <ENTER>". The following will appear.

```
 10 PRINT"THIS S A XCOMPUTER."
```

The cursor will appear on the first character which is the P in PRINT. Move the cursor with the space bar to the S where we want to insert the I. Then do the following:

1. Type "I" to enter the insert mode.
2. Type another "I" to insert an I. This will appear on the screen.
3. Hold the shift key down and press the "UP ARROW". This exits the insert mode.

We have inserted the I for IS. You can now press the "L" key and the line will be displayed in its present form and the cursor will appear on the first character. The "I" should be inserted. You are still in the EDIT mode.

Now move to the X by using the space bar several times. Then press the "D" key once. This will delete one character which is the X. You can move back and forth with the space bar and left arrow to verify that the X has been deleted. Pressing "L" lists the line and puts the cursor in the first position. If the statement is like you want it, then press the "ENTER" key to complete the editing process.

Let's summarize these commands.

1. To enter the edit mode type EDIT LINE NUMBER ENTER. Example: EDIT 10 <ENTER>
2. The cursor is over the first character. Move to the character to be edited by using the space bar and left arrow.
3. Pressing "D" will delete the character under the cursor. Press "D" for each character you want to delete.
4. Press "I" to insert characters. Type in the characters you want to insert. Hold the shift key down and press the up arrow key to exit the insert mode.
5. Typing "L" will list the program and leave you in the edit mode.
6. Press <ENTER> to exit the edit mode.

We will cover more on using the EDIT command next month.

PROGRAMMING

Last month we gave some examples on editing a text file. We showed what would be involved in deleting characters or inserting characters. To delete N characters we have to move all characters past those deleted forward in memory N locations. If we insert N characters then we have to move all characters back in memory by N locations. Let's look at moving blocks of text.

BLOCK MOVING

Why would we want to move a block of text? Suppose you have an address file of customers. You would like to rearrange the file so that all of the customers in the same state or city are ordered together. Or maybe you would like to update your mailing list and move less active customers to the end of the list. Let's look at the pointers we would need first.

1. X=Beginning of the block
2. Y=Ending of the block
3. Z=New location for the block.

There are two methods we can use. Let's demonstrate these with example programs.

```
5 ?"BLOCK MOVE SUBROUTINE
10 FOR J=1 TO Y-X
15 'PUT CHARACTER IN A
20 A=PEEK(J)
25 'MOVE ALL CHARACTERS UP ONE
   LOCATION
30 FOR K=X+1 TO Z
40 B=PEEK(K):POKE B-1,K
50 NEXT K
55 'STORE THE CHARACTER IN THE
    NEW LOCATION
60 POKE Z,A
70 NEXT J
```

The preceeding program would be very time consuming. It holds the first character in A and then moves all characters to create a space at Z. Then the value A is stored in Z. It then takes the next character and continues until all characters have been moved.

MEMORY BUFFER

The time can be greatly reduced by using a memory buffer. A good place to put the buffer would be at the end of the text file. Lets look at a program for using this method of moving a block.

```
10 'MOVE A TEXT BLOCK TO END OF
   FILE-EF
15 P=EF:BS=Y-X
20 FOR J=X TO Y
30 A=PEEK(J)
40 POKE P,A:P=P+1
50 NEXT J
60 'MOVE TEXT FORWARD TO FILL
    IN WHERE THE BLOCK WAS
70 FOR J=Y TO Z
80 A=PEEK(J)
90 POKE J-BS,A
100 NEXT J
110 'MOVE THE BLOCK TO THE NEW
     LOCATION
120 FOR J=0 TO BS-1
130 A=PEEK(J+EF)
140 POKE Y-BS+J,A
150 NEXT J
```

ML SUBROUTINES

Because of the slowness of basic for quickly moving blocks of information, we will use machine language (ML) subroutines for moving information in our files. We are starting a series on ML Programming in this issue and suggest that you read these so you can understand how a ML Subroutine works. A ML subroutine can be linked to basic with the EXEC command. A number follows the EXEC command. This number is the memory location where the ML subroutine starts. When the ML subroutine is completed, it returns to basic. So as far as basic is concerned, it thinks that the ML subroutine was a basic subroutine.

The example basic subroutines we are giving show how to move information. The procedure is the same when using ML Subroutines. Techniques for programming are similar even when different programming languages are used.

We have been using an address or text file as an example. We also want to show how to create a file similar to a spread sheet so that mathematical operations can be performed. Other file example would be a monthly record of expenses,inventory file, and depreciation schedule for capital equipment.

The Basic commands that we will be using have been covered in this series. So now we are giving examples of how to use the programming skills we have developed.

ML PROGRAMMING
(Part 1)

Due to the requests of our readers we are starting this series. As an introduction let's define our objectives. First of all we want to look at the microprocessor's architecture. We will show in general terms how the microprocessor and its support circuitry works. The second thing we want to show is how to hand assemble machine language codes. This is good practice because it will really give you an insight into the operation of the microprocessor. We can expand this and design a simple assembler that you can use.

Our third objective will be to show how to merge machine language subroutines with basic programs. This is the technique we use for most of our programming. This is a very powerful tool because you have the ease of programming in basic and the speed of machine language subroutines.

Also we will define the operations that can be performed. If you have an assembler then this will make it easier for you to understand how to use it. Where possible we will relate ideas to basic.

68Ø9 STRUCTURE

A microprocessor has internal registers that hold values. A register can operate on another register or external memory. As an example the A and B registers are generally used for calculations. These are 8 bit registers and can hold a value up to 256. We will be using binary and hexidecimal arithmetic. Hexidecimal consists of 4 binary bits and has a decimal value from Ø to 15. It uses the numbers from Ø to 9 and the letters from A to F to indicate values from 1Ø to 15 respectively. Other 8 bit registers are the direct page register and the conditional code register. Each of these registers contains useful information which we will explain later. There are four 16 bit registers. These are generally used as pointers and are labeled PC,X,Y,U, and S. PC stands for Program Counter. This points to the memory that contains the next instruction. X and Y are index registers. We can use them as a reference for doing operation. As an example for moving a block of data, the X register can point to the next data byte to get and the Y can point to the memory location to which the byte is going. U and S are called stack registers. A stack is a portion of RAM reserved for the microprocessor. The following is a summary of the microprocessor's registers.

X-Index Register
Y-Index Register
U-User Stack Pointer

SPECTROGRAM

MAGAZINE FOR COLOR COMPUTER USERS.

- * Programs for business, home management, self-improvement, games, and utilities
- * Reviews of Color Computer products
- * Tutorials on programming in Assembly, C, Pascal, and Basic
- * Contests

As an introductory offer, you can order the first year of SPECTROGRAM Magazine at 40% off the cover price. For $18, you will receive 12 issues of the magazine that could become the most informative addition to your Color Computer system.

We want to establish a line of two-way communication between our staff and our readers as an aid in serving your needs. Please enclose any comments or special requests with your subscription form.

GROUP RATES: $15 with orders of five or more subscriptions!

PLEASE SEND ME 12 ISSUES OF SPECTROGRAM MAGAZINE FOR $18 *(40% OFF THE COVER PRICE)*.

Name: ____________________

Address: ____________________

City: ____________________

State: ____________ Zip: ____________

( ) Check Enclosed ( ) Visa ( ) Mastercard

Card # ______________ Exp. Date: ________

Mail to: SPECTROGRAM Magazine
P.O. Box 138
Rockford, IL 61105
(815)968-9600

COMPUTER TYPE:
( ) 64K Color Computer
( ) 32K Color Computer
( ) 16K Color Computer
( ) 4K Color Computer
( ) Other--Specify ________

LANGUAGES:
( ) Extended Basic
( ) Color Basic
( ) Disk Basic
( ) Basic09
( ) Pascal
( ) C Compiler

PERIPHERALS:
( ) Printer Type ________
( ) Modem Type ________
( ) Disk Drive (0) (1) (2) (3)
( ) Multi-Pak Interface
( ) Other--Specify ________

PROGRAM PREFERENCE:
( ) Business
( ) Games
( ) Graphics
( ) Tutorials
( ) Utilities
( ) Home Management
( ) Self-Improvement

BEST

WE'VE CHOSEN THE BEST OF OVER 450 PROGRAMS AND PACKAGED THEM FOR YOU! 10 TO 12 PROGRAMS EACH PACKAGE. COLOR COMPUTER ONLY.

#1 Home Management I

NEW!

Budget
Checkbook Balancer
Cost of Living
Tinycalc
Electronic Datebook
Account Manager
Stock Market
Word Processor
Lottery Analyst
Coco Database
Coco Terminal
Bartender

#2 Home Management II

Video Cassette Organizer
Home Product Evaluation
Electric, Gas & Water Cost
Baseball Manager
Car Manager
Ham Radio Log
Home Inventory
Personal Directory
Recipe Machine
Disk Labeler
Password Scrambler
Disk Directory Print

#3 Educaltion

NEW!

Flash Card
Spanish Lessons
Typing Tutor
Creativity Test
Arithmetic Football
Cost of Living
Math Tutors 1, 2
Trigonometry Tutor
Typing Game
Word Tests
Talking Alphabet
Clown Dunk Math

#4 Adventures

Treasures of Barsoom
Killer Mansion
College Adventure
Coco-Terrestrial
Escape
Zector
Skid Row
Quest
Naughide
Haunted House

#5 Games

Trek
Galactic Conquest
Warlords
The Power Sword
Steps
Robot Bomber
Force Field
Rat Attack
Caterpillar Cave
Meteor

#6 Utilities

Disk Dir Prot
Dir Pack & Sort
Disk Zapper
Roll Out
Doss Boss
Disk Backup
51'24 Editor
51'24 Screen
Autocopy
Fastsort
I O Error Ignorer
Text Screen Print

Some of these programs above can sell for $29.95 each

VISA

RAINBOW CERTIFICATION SEAL

$^{\$}29^{95}$each
TAPE OR DISK
★★ SPECIAL ★★
BUY 2 PACKAGES AND GET THIRD ONE FREE

THE GREATEST SOFTWARE DEAL ON EARTH!

GET 12 DISKS OR TAPES A YEAR CONTAINING OVER 120 QUALITY PROGRAMS. A SUBSCRIPTION TO T & D SOFTWARE CONSISTS OF 10 READY-TO-LOAD PROGRAMS DELIVERED BY FIRST CLASS MAIL EVERY MONTH.

NO, WE ARE NOT THE SAME AS THE RAINBOW ON TAPE. IN FACT, MANY SUBSCRIBERS HAVE WRITTEN IN AND SAID THAT WE ARE<u>MUCH</u> <u>BETTER THAN RAINBOW ON TAPE!</u>

PRICES

	TAPE OR DISK
1 YEAR (12Issues)	70.00
6 MO. (6 issues)	40.00
1 ISSUE	9.00

Michigan Residents Add 4%
Overseas Add $10 to Subscription Price
Persona! Checks Welcome!

- ★ 16k-64k Color Computer
- ★ Over 3800 Satisfied Customers
- ★ Back Issues Available From
- ★ July '82 (Over 450 Programs)

★ THIS MONTH ONLY ★

SUBSCRIBE FOR A YEAR AND RECEIVE A FREE PACKAGE OF YOUR CHOICE. INDICATE WHICH ONE

1. Home Man I
2. Home Man II
3. Education
4. Adventures
5. Games
6. Utilities

OUR LATEST ISSUE CONTAINED

1. INCOME PROPERTY MANAGEMENT
2. BILL BOARD 2
3. MOUNTAIN BATTLE
4. THE TEN ROUND FIGHT
5. COCO - KEENO
6. HIGH RESOLUTION HOCKEY
7. LOGIC
8. ON SCREEN SCALE
9. LIBERTY SHIP
10. SINGLE STEP RUN

RAINBOW CERTIFICATION SEAL

GET ISSUE #45 ABOVE FOR ONLY $3.00

MasterCard

T & D SUBSCRIPTION SOFTWARE, P.O. BOX 256C, HOLLAND, MI 49423 (616) 396-7577

S-Hardware Stack Pointer
PC-Program counter
A-Accumulator Register
B-Accumulator Register
DP-Direct Page Register
CC-Conditional Code Register

The 6809E microprocessor was developed by Motorola. The instructions are confusing if you are not familar with microprocessor terminology. We want to explain what the terms mean so that you can read the data sheet and have a good idea of what it says. The registers contain the important information that the microprocessor uses. The microprocessor either uses the information in the registers or goes to a location designated by a register. The next machine code to which the PC is pointing determines what is to happen next.

It is not hard to write simple routines with a 6809E microprocessor. However it is necessary to understand how the microprocessor operates and the tools we have available.

Interrupts

Interrupts stop the microprocessor from doing the task it is doing, and assigns it another task. Depending upon the type of interrupt, it is possible to continue with the first task after the interrupt. We are all familiar with the rear reset push button. When this button is pushed an interrupt called RESET is enabled which causes the computer to go through its initialization routine. When the computer is first turned on, the RESET interrupt is activated. This forces the microprocessor to go to memory location 65534 ($FFFE) for the address of the RESET routine. This address is contained in the basic ROM along with the reset routine.

Next month we will continue and take a few of the microprocessor's commands with examples. We will also discuss more of the computer's architecture.

INTERFACING COMPUTERS (Part 3)

You have a complete color computer system with disk drives, printer, and numerous software. You have had your eye on a second computer for some time, and finally you purchased it. Wouldn't it be nice if information could be transferred between the two computers? What if the second computer is a different type such as an IBM PC, Apple, or a Commodore 64?

Can basic programs be transferred between two computers of different types? The purpose of this series is to answer these questions and show what can be done. We have been looking at serial ASCII. This involves transferring one bit at a time from one device to another. This can be slow or relatively fast depending upon the transfer or baud rate. The baud rate is the number of bits per second and can go as high as 9600. The cassette works at about 1500 baud so 9600 baud is over 6 times as fast as the cassette.

Information can be transferred between computers in different cities by using a modem and a telephone line. A modem converts electrical voltages to tones and tones to electrical voltages. Tones are required to pass information through the telephone lines. These are usually slow and run from 300 to 1200 baud.

Terminal programs are used to transfer programs and data between two computers. We mentioned transferring basic programs. For two color computers we can transfer each byte in the program designated by the vectors in locations 25- 28. For other computers that use Microsoft Basic, programs can be trnasferred in ASCII format. However, the commands must be compatible. Some computers use different commands than others although they essentially use Microsoft Basic. As an example the color computers use the command PRINT#-2 to print to a printer while many computers use LPRINT.

For transferring text files you do not need to be concerned about compatibility because ASCII is the same for all computers. Basic programs can be transferred in ASCII format and the computer reconstructs the Basic tokens or codes so the program can be run.

INITIALIZATION

To use serial ASCII both the sending computer and the receiving computers must be in agreement. They both must know the baud rate, word length, number of stop bits, and the type of parity used. A start pulse is sent to synchronize the transfer. The data bits, the parity bit, and the stop bits follow. The receiving computer reconstructs the byte and processes it before the next byte or word is sent.

COMPUTER to PRINTER

When a serial printer is to be connected to a color computer, the parameters mentioned earlier have to be set. Most printers just require a baud rate adjustment. For color computers, the values in locations 149 and 150 determine the baud rates. These values can be determined from your manual. Common values are 1, 187 for 300 baud; 0, 87 for 600 baud; and 0,1 for 9200 baud.

HANDSHAKING

Handshaking allows the computer to be stopped by the output device. If data is being sent too fast for the printer, it can stop the computer until the data in its buffer has been printed. Then the printer can ask for more characters to print. If a printer has a large buffer or spooler, then it can receive a block of characters into its buffer and release the computer to do work while the characters are being printed. Most printers have at least a buffer equivalent to one line of characters. Some have buffers up to 8K and larger.

25 Pin RS-232

Most RS-232 devices contain a 25 pin plug. A mating connector with proper wiring to mate with the 4 pin connector on the rear of the color computer is required. The wiring of the plug is standard except for the handshake. The following is a chart of a standard RS-232 pin assignment for the most common pins. This is what you can expect to find on computers, printers or modems.

Pin	Description
1	Protective ground
2	Transmitted Data
3	Received Data
4	Request to Send
5	Clear to Send
6	Data Set Ready
7	Signal Ground
20	Data Terminal Ready

For a color computer the pins are:

1 -	Yellow-	CD or Handshake
2 -	Green-	Receive data
3 -	Red -	Ground
4 -	White-	Send Data

Two color computers can be

connected together without handshaking as follows:

Pin (CC-1)	Pin (CC-2)
1 - Yellow (not used)	
2 - Green	3 - Red
3 - Red	2 - Green
4 - White	4 - White

RS-232 to CC

To interface RS-232 devices a rule of thumb is to exchange the wires going to the input and output of one of the plugs. In other words the output from the second device should go to the input of the first device and the input to the second device should be connected to the output of the first device. These are pins 2 and 3 of the 4 pin color computer connector and also pins 2 and 3 of the 25 pin RS 232 connector. The other pin is ground which is pin 4 for the color computer and pin 7 for the RS 232 connector. Therefore the following is the wiring connections:

CC	RS-232
2 Transmit data	3 Receive data
3 Receive data	2 Transmit data
4 Ground	7 Ground

+ + +

RENEWAL TIME?

The date beside your name on the address label indcates the last issue you will receive. Send in your renewal if you want to continue receiving technical information on Color Computers. This is the last issue for those with 5/86.

ROULETTE
(GAME)

This exciting game challanges you to beat the odds. Start out with $1000 and place your bet. Bet on BLACK/RED, ODD/EVEN, HIGH /LOW, or all three. This program is provided by courtesy of T & D Software and is used by permission. See their advertisement on page 7.

```
1 REM COPYRIGHT (C) T&D SOFTWARE
 1986    roulette
2 PMODE0:GOTO60000
3 LINE(128,96)-(128+X,96-Y),PSET
:RETURN
4 LINE(128,96)-(128+X,96-Y),PRES
ET:RETURN
5 FORI=1 TO 32:X=CX(I):Y=CY(I):G
OSUB4:X1=FNQ(AG):Y1=FNR(AG)
6 X=X1:Y=Y1:GOSUB3:CX(I)=X
7 CY(I)=Y:NEXT:CIRCLE(128,96),70
,2,.95:CIRCLE(128,96),90,2,.95:R
ETURN
50 CLEAR200:DIM CX(32),CY(32):DE
FFNQ(Z6)=X*COS(Z6)+Y*SIN(Z6):DEF
FNR(Z7)=-X*SIN(Z7)+Y*COS(Z7):PLA
Y"V31T200L200"
52 FORI=1 TO 9:READ CX(I),CY(I):
NEXT
54 FORI=1 TO 8:CX(I+9)=-CX(I):CY
(I+9)=CY(I):NEXT
56 FORI=2 TO 9:CX(I+16)=CX(I):CY
(I+16)=-CY(I):NEXT
58 FORI=11 TO 17:CX(I+15)=CX(I):
CY(I+15)=-CY(I):NEXT
60 AG=3.5:AG=AG/57.79
61 CLS:PRINT:PRINT:PRINT"
    roulette":PRINT:PRINT
62 PRINT" PLAY A GAME OF COMPUTE
R ROUL-    ETTE. YOU WILL START O
UT WITH    $1000. SEVERAL BETTING
 OPTIONS   ARE AVAILABLE."
63 PRINT" THERE ARE 32 NUMBERS O
N THE      WHEEL. ALL NUMBERS EVE
NLY DI-    VISIBLE BY 3 ARE RED,
THE REST   ARE BLACK."
64 GOSUB9000:CLS:PRINT:PRINT
65 PRINT" YOU CAN BET ON THE COL
OR OF      THE WINNING NUMBER. PI
CK THE     CORRECT COLOR AND WIN
THREE      TIMES YOUR BET. PICK T
HE WRONG   COLOR AND YOU LOSE DOU
BLE YOUR   BET."
66 PRINT" YOU CAN ALSO BET ON WH
ETHER      THE WINNING NUMBER IS
ODD OR     EVEN. A WIN GETS YOU D
OUBLE      YOUR BET. IF YOU BET I
NCORRECT-  LY, THEN YOU LOSE DOUB
LE YOUR    BET.":GOSUB9000:CLS:PR
INT@64
67 PRINT" THE LAST BETTING OPTIO
N ALLOWS   YOU TO BET ON HIGH OR
LOW. ALL   NUMBERS 1-16 ARE LOW.
NUMBERS    17-32 ARE HIGH. EVEN M
ONEY IS    WON OR LOST ON THIS BE
T."
68 PRINT" NOTE THAT YOU DON'T HA
VE TO      BET ALL THREE OPTIONS.
 YOU CAN   BET ON ANY ONE, TWO, O
R ALL      THREE BETTING OPTIONS.
"
69 GOSUB9000:GOTO100
80 PMODE1:PCLS:FORI=1 TO 32:X=CX
(I):Y=CY(I):GOSUB3:NEXT:CIRCLE(1
28,96),70,2,.95:CIRCLE(128,96),9
0,2,.95
81 FORI=3 TO 7 STEP2:PMODE1,I:PC
LS:NEXT
82 PMODE1,3:GOSUB5:PMODE1,5:GOSU
B5:PMODE1,7:GOSUB5
89 RETURN
100 PM=1000:CLS:PRINT@235,"worki
ng"
101 GOSUB80
110 CLS:PRINT@136,"you have  $";
PM
120 PRINT:PRINT"   enter amount
of bet";:INPUT BT:IFBT>PM THEN11
0
130 PRINT:PRINT"   SELECT rED, b
LACK, nO BET"
132 K$=INKEY$:IFK$="N" THENCF=0:
GOTO140
134 IFK$="R" THENCF=1:GOTO140
136 IFK$="B" THENCF=-1:GOTO140
139 GOTO132
140 PRINT:PRINT"   SELECT oDD, e
VEN, nO BET"
142 K$=INKEY$:IFK$="N" THEN NF=0
:GOTO150
144 IFK$="E" THEN NF=1:GOTO150
146 IFK$="O" THEN NF=-1:GOTO150
149 GOTO142
150 PRINT:PRINT"   SELECT hIGH,
lOW, nO BET"
152 K$=INKEY$:IFK$="N" THEN HL=0
:GOTO160
154 IFK$="H" THENHL=1:GOTO160
156 IFK$="L" THENHL=-1:GOTO160
159 GOTO152
160 SP=0
```

```
170 TD=1
180 FORL=1 TO 4
190 FORI=1 TO 7
200 FORQ=1 TO 7 STEP 2:PMODE1,Q:
SCREEN1,0:PLAY"O5E"
202 CIRCLE(128,96),80,2,.95,SP,S
P+.02
210 FORQA=1 TO TD:NEXT
212 CIRCLE(128,96),80,1,.95,SP,S
P+.02:SP=SP-.02:IFSP<0 THENSP=1
220 NEXT:NEXT:TD=TD*4:NEXT
230 WN=RND(32):IFRND(10)>7 THEN2
40
231 IFCF=0 THEN230
232 IFCF=1 THENWN=RND(32):IFWN/3
=INT(WN/3) THEN232 ELSE240
234 WN=RND(32):IFWN/3<>INT(WN/3)
 THEN234 ELSE240
240 IF WN/3 =INT(WN/3) THEN WC=1
 ELSEWC=-1
250 IF WN/2=INT(WN/2) THEN NN=1
ELSENN=-1
260 IF WN<17 THEN WW=-1 ELSEWW=1

270 CLS:PRINT@32
280 PRINT"        RESULTS OF BETTI
NG"
290 PRINT
300 BQ=1:PRINT" YOUR BET $";BT;:
IF CF=-1 THENPRINT"BLACK ";:BQ=0
 ELSEIFCF=1 THENPRINT"RED ";:BQ=
0
310 IF NF=-1 THENPRINT"ODD ";:BQ
=0 ELSEIFNF=1 THENPRINT"EVEN ";:
BQ=0
320 IF HL=-1 THENPRINT"LOW";:BQ=
0 ELSEIFHL=1 THENPRINT"HIGH";:BQ
=0
330 IFBQ=1 THENPRINT"NO BET";
340 PRINT
350 PRINT:PRINT"        WINNING NU
MBER: ";WN
360 PRINT
370 PRINT"  WINNING COLOR: ";:IF
WC=1 THENPRINT"RED  "; ELSEPRINT
"BLACK";
375 IFCF=0 THENPRINT" NO BET":GO
TO380
376 IFCF=WC THENPRINT" WIN!!!":P
M=PM+BT*3 ELSEPRINT" lose":PM=PM
-BT*2
380 PRINT"  ODD/EVEN      : ";:IF
NN=1 THENPRINT"EVEN "; ELSEPRINT
"ODD  ";
382 IFNF=0 THENPRINT" NO BET":GO
TO390
384 IFNF=NN THENPRINT" WIN!!!!":
PM=PM+BT*2 ELSEPRINT" lose":PM=P
M-BT*2
390 PRINT"  HIGH/LOW      : ";:IF
WW=1 THENPRINT"HIGH "; ELSEPRINT
"LOW  ";
392 IFHL=0 THENPRINT" NO BET":GO
TO400
394 IFHL=WW THENPRINT" WIN!!!!!"
:PM=PM+BT ELSEPRINT" lose":PM=PM
-BT
400 PRINT:PRINT"        YOU NOW HA
VE $";PM
410 IFPM>0 THENGOSUB9000:GOTO110

420 PRINT"   THE GAME IS OVER FO
R YOU."
999 END
9000 PRINT@484,"press [enter] to
 continue";
9010 IFINKEY$<>CHR$(13) THEN9010
 ELSERETURN
50000 DATA66,0
50002 DATA64,10
50004 DATA62,20
50006 DATA56,30
50008 DATA48,40
50010 DATA42,50
50012 DATA30,56
50014 DATA16,62
50016 DATA0,64
60000 PCLEAR8:GOTO50
```

PAGE -1
PROGRAM DEVELOPMENT

In our 1985 October issue we showed how to place a program in memory below the keyboard buffer. Since page 0 is normally considered the lowest beginning operating location above the keyboard buffer, we designated the lower operating location as page -1. The memory from 500 to about 720 is generally not used and can be used for data or a small program.

Have you ever been working on a program and for some unexplained reason the computer went through a hard reset and wiped out your program? You try listing the program and only get an OK on the screen. The program is still in the computer but the vectors have been modified so that the program appears to have been erased. If you have extended basic and a disk drive you will see the following values if you look at locations 25 - 28:

25	26	Value	27	28	Value
38	1	9729	38	3	9731

Many times we have discussed vectors or pointers. For review a vector takes two bytes and points to a memory location. The value of the vector is determined by multiplying the lower byte by 256 and adding the upper value. Thus a vector consisting of components 38,1 will have a value of 38*256 + 1 or 9729. The vector in location 25 points to the beginning of basic and the vector in location 27 points to the end of basic. The length of the program can be calculated by subtracting the values of the beginning and ending vectors.

BASIC PGM STRUCTURE

We can recover our basic program if we can restore the vectors. Let's look at the organization of a basic statement.

The first byte is a zero. The next two bytes contain a vector that points to the memory location of the next statement. The third and fourth bytes contain a vector which is the statement number. A chart showing these relationships follows:

MEMORY	VALUE	
M	0	Statement Follows
M+1,M+2	X	Vector to next stmt
M+3,M+4	Y	Statement Number
M+5 TO K		Commands
K+1	0	Statement Follows
K+2	Z	Next Statement

Now let's take an example with numbers.

MEMORY	VALUE	VECTOR VALUE
9728	0	
9729	38	9749
9730	21	
9731	0	10
9732	10	
XX		
XX		
9748	0	
9749	38	9773
9750	45	

Notice that the 0 has a dual purpose. If location 9728 is the first statement, then the 0 precedes the statement. The 0 in 9748 means the end of statement 10 and the beginning of the next statement.

After the computer went through the hard reset, the values in 9729 and 9730 would be 0. If we can restore this vector then we can list the program. If we are to do any editing or save the program, we will have to restore the end of program vector in locations 27 and 28.

RESTORING the FIRST VECTOR

Since all bytes, after the

first 4, have values greater than zero, we can look at the value stored in each location until we find a zero. The next memory location after the end of basic 0 will be the start of the next statement. The value of this vector should be in the first two bytes of the first statement. We can convert a memory location M into the two vector components with the following equation:

(1) MS=INT (M/256):LS=M-256*MS

To find the value of a vector from its components use the following equation:

(2) V= 256 * MS + LS

Our procedure will be to skip over the first 4 bytes at the beginning of the program and begin looking for a 0 to indicate the end of the first statement. The first vector in the first statement should point to the beginning of the second statement. The memory location after the 0 is the required memory for the vector in the first statement. Knowing this memory we can determine the MS and LS from equation (1) and poke these values into the first two bytes of our program. For the numerical example we gave, these would be the values in 9729 and 9730.

After the beginning vector is restored you can list or run the program, but you can not edit or save it until the end of program vector is restored.

RESTORING END of BASIC VECTOR

After the first vector is restored, continue looking at each statement until you find two zeros after the end of basic zero. In other words you will a vector that has a value of 0 for the next statement. This is the end of the program and this memory value can be converted into its MS and LS components by using equation (1).

BASIC PROGRAM RESTORER

This program will restore a basic program. A page -1 program must be less than 200 bytes. We had to remove all comments to be able to reduce the size. Since we have discussed the requirements for restoring a basic program after a hard reset, you should be able to follow what we did.

After the hard reset POKE 25,2 to use the page -1 location. Then load in the program. It will ask you for M. This is the most significant byte of the erased program. For disk basic this is 38 if you did not do any PCLEAR commands. It is always a good idea to know where your program is residing in memory. Do a PEEK (25) to find out where it starts. The starting memory will be S=256 * PEEK (25) + 1. A 1 is always in location 26 so you can just add 1 instead of doing a PEEK (26).

RESTORE
PAGE -1 PROGRAM

The following program will restore the vectors for a basic program after a hard reset. To use the program POKE 25,2 and load the program. Then run it and do the memory pokes printed on the screen. Each statement's memory location will be printed so you can see what is happening.

```
1' PGM 5-1-86. RESTORES A BASIC
   PROGRAM AFTER A HARD RESET.
   REMOVE REMARKS FOR PAGE -1.
   COPYRIGHT (c) 1986 BY DYNAMIC
   ELECTRONICS INC.
2 INPUT"M";X:B=256*X+1:M=B+4:GOS
   UB5:POKEB,S:POKEB+1,T
3 PRINT"M="M:M=M+4:GOSUB5:IFV>0T
   HEN3
4 M=M+2:GOSUB5:PRINT"POKE27"S,"P
```

```
  OKE28"T-1,"POKE25"X:END
5 A=PEEK(M):M=M+1:V=256*PEEK(M)+
  PEEK(M+1):S=INT(M/256):T=M-256
  *S:IFA>0THEN5ELSERETURN
```

To use the program remove statement 1 and (C)SAVE "RESTORE". Then when you loose a program POKE 25,2 and LOAD and RUN "RESTORE". You will have to enter the most significant byte in location 25 of the program that was erased. The program corrects the vectors for the erased program and prints values for you to poke to correct the basic vectors. You can list the program and it is a good idea to save it before trying to run it again.

You can test RESTORE by loading a program. Then POKE 113,0 and push the rear reset button. The program is lost. You can now use RESTORE to recover the program.

+ + +

```
* * * * * * * ** * * * * * * * *
*                               *
* Would you like to write a     *
* page -1 program for us? We    *
* will pay you $5 if we pub-    *
* lish your program. Here are   *
* the rules.                    *
*                               *
* 1. It must be less than 200   *
*    bytes.                     *
* 2. It must be able to oper-   *
* ate at the page -1 desig-     *
* nated memory area POKE 25,2.  *
* 3. Write a short paragraph    *
* about what your program does. *
* 4. Send Program listing.      *
*                               *
* * * * * * * ** * * * * * * * *
```

EDITOR'S COMMENTS

This month we are concluding our large memory series. In this series we showed how to use the extra 32K memory bank in 64K computer. The second bank can be initiallized for basic and a program placed it it. You can then place a program in the first bank and go from one program to the other by exchanging banks. We also developed a ramdisk program for storing programs in the second bank and gave machine language subroutines for transferring data between the two banks. Also we covered the all ram mode and showed how to store a program in the upper memory.

This month we are starting a series on machine language (ML) programming. If you have wanted to know how to program in ML, now is you chance. We have been very successful in writing our programs in basic and using ML subroutines when we need something that basic does not offer such as speed. We will define the terms, talk about the microprocessor's architecture, and show how to assemble the codes. This will help if you have an assembler and are confused about how to use it.

We are looking at some hardware projects. This is not as easy as software. For example you can not damage your computer with a bad statement, but you can damage it if you drop some solder across pins on one of the integrated circuits. Hardware projects that plug into the cassette, printer, joystick jacks, do not require going inside the computer.

We appreciate the letters we have received and are using them as guide lines for future subjects. If you have a subject you would like for us to cover or a question you would like for us to answer, please write us a note.

Last Fall we discussed using Page -1. This is unused memory below the keyboard buffer from 512 to about 720. In this issue we have a page -1 utility that will restore a basic program after a hard reset. We are interested in receiving page -1 programs, and will pay $5 for each we publish. They should be less than 200 bytes long. So if you have a short program, see if it will work in page -1 and send it to us.

Also in this issue we are including a summary of all articles and programs we have published. This can serve as a quick reference for finding material on a subject.

LARGE MEMORY PROGRAMS (FINAL)

In this series we have shown how to write programs for memories larger than 32K. Most programs are designed for a 32K computer. If you have a 64K, 128K, or larger computer then you have to modify or write your own programs. We used the second memory bank in a 64K computer as our example for program development.

2 BANK MODE

We gave machine language subroutines for partitioning the computer for two bank operation. If the first bank is transferred into the second bank, then basic can be run in the second bank. A 32K program can reside in each of the two banks with both being accessible. The basic, extended basic, and disk basic read only memories (ROM) are available with each bank. They occupy the upper 32K of memory. With this configuration you have a total of 96K that can be accessed with a 64K computer.

We showed how to store data in the second bank. If you have a program that requires a lot of data, then this is a good procedure. We gave an address file program as an example. Depending upon the organization of the program, a total of 32K of data can be stored in the second bank. It would be possible to store 18K to 20K in the first bank for a total of around 50K or more of data.

The principles we presented could be used with larger memories. For example suppose you have a 256K memory that has a ramdisk. You can quickly move blocks of data from the ramdisk into your operating memory to do data search or manipulations. This would be the same as loading and storing files on a disk but would be many times faster. We developed a ramdisk for using the second 32K memory bank for program storage.

ALL RAM MODE

The all RAM mode configures the computer so that all of the memory is RAM. The ROM information is copied to RAM. In this mode the operating system can be modified. For a disk system the upper 8K of memory is available for use. Data can be stored in this memory or a program can be placed there. We showed how to do this last month. This mode only allows about 8K more of memory. Therefore the two bank mode is generally desireable over the all ram mode.

+ + +

COMPUTER GRAPHICS
(Part 15)

In this series we have been covering the Extended Basic Graphic Commands. Last month we looked at the GET and PUT commands. These allow us to move a figure from one part of the screen to another. We gave a program to demonstrate this.

This month we want to look at developing a drawing program. With it we will be able to move to different parts of the screen with the arrows and draw the figures we have covered. The basic figures are a line, box, filled boxed, erased box, and a circle. We want the capability of getting a figure and putting it into another location as we demonstrated last month. Also we would like to have the option of saving a drawing on a disk or cassette. If you have a graphics printer with a graphics printer ML subroutine, then you can make a hard copy of your drawing.

PROGRAM DEVELOPMENT

We will have a menu that does all of our required operations. While in the graphics mode we want to move in any of the 4 directions designated by the arrows. If we press any other key, we want to go to a subroutine and do the operation assigned to that key. Our keys will cause the following operations to be performed.

Key	Operation
B	Draw a box
C	Draw a circle with the first point as center and the second the radius.
E	Erase the box defined by the two points.
F	Draw a filled box
I	Initialize first point
L	Draw a line
M	Menu
P	Print Coordinates of the points
N	Clear the screen

Our menu should allow us to do the following:

C	Change the color
L	Load a drawing
S	Save a drawing
H	List all options

This month we are doing the first part. We will be able to draw the function indicated by the keys or perform the operation. This will allow us to make and modify a drawing. When the program is run you are asked to enter the initial point for X1 and Y1. The second point is made equal to the first by letting X2 = X1 and Y2 = Y1. You are then taken to the graphics mode where you can begin drawing your figure.

We did not add the menu and will add it next month. The drawing is a little slow so we added sound to let you know that the computer received your pressed key. Also we set and reset the points so you can see where they are located. If you want to move the first point P1 then press the "I" key and it will be moved to P2 or the second point. This is like picking up your pencil and moving to another location. Comments are included so you can see what each section does.

GRAPHICS DRAW PROGRAM

```
10 PRINT"GRAPHICS PROGRAM
20 PRINT"PROGRAM 5-2-86
30 PRINT"cOPYRITE (c) 1986 BY
   dYNAMIC eLECTRONICS iNC.
40 PRINT"THIS IS DESIGNED
   AROUND USING TWO POINTS
   P1=X1,Y1 AND P2=X2,Y2
50 INPUT"FIRST POINT";X1,Y1
60 X2=X1:Y2=Y1:PCLS
70 CLS:PRINT @0,"X1="X1;"
   Y1="Y1;"X2="X2;"Y2="Y2
80 INPUT"THE TWO POINTS ARE ON
   TOP OF EACH OTHER.  USE THE
   ARROW KEYS TO MOVE THE
   SECOND POINT. PRESS A
   KEY";Y$
90 PMODE 3,1: SCREEN 1,0 'SET
   UP FOR GRAPHICS
100 X$=INKEY$:GO SUB 300:GO SUB
    310:IF X$="" THEN 100
    'FORCE THE COMPUTER TO WAIT
    FOR A PRESSED KEY
110 SOUND 150,1:X=ASC (X$)
120 IF X=9 THEN X2=X2+1 'RIGHT
    ARROW
130 IF X=8 THEN X2=X2-1 'LEFT
    ARROW
140 IF X=10 THEN Y2=Y2+1 'DOWN
    ARROW
150 IF X=94 THEN Y2=Y2-1 'UP
    ARROW
160 GOSUB 300
170 IF X$="C" THEN GO SUB 320
    'DRAW A CIRCLE WITH X1,Y1
    CENTER AND RADIUS EQUAL TO
    DISTANCE FROM P1 TO P2
180 IF X$="L" THEN GOSUB 260
    'DRAW A LINE
190 IF X$="N" THEN PCLS 'ERASE
    DRAWING
200 IF X$="I" THEN X1=X2:Y1=Y2
    'INITIALIZE FIRST POINT TO
    EQUAL THE SECOND
210 IF X$="B" THEN GOSUB 270
    'DRAW A BOX
220 IF X$="F" THEN GOSUB 280
    'DRAW A FILLED BOX
230 IF X$="E" THEN GOSUB 290
    'ERASE THE BOX
240 IF X$="P" THEN 70 'PRINT
    THE COORDINATES OF THE
    POINTS
250 GO TO 100
260 LINE (X1,Y1) -(X2,Y2),
    PSET:RETURN 'DRAW A LINE
270 LINE (X1,Y1)-(X2,Y2),PSET,
    B:RETURN 'DRAW A BOX
280 LINE (X1,Y1)-(X2,Y2),PSET,
    BF:RETURN 'DRAW A FILLED IN
    BOX
290 LINE (X1,Y1)-(X2,Y2),
    PRESET,BF:RETURN 'ERASE BOX
    OUTLINE
300 PSET (X1,Y1):PSET
    (X2,Y2):RETURN 'THIS SETS
    THE POINTS
310 PRESET (X1,Y1):PRESET
    (X2,Y2) :RETURN 'THIS
    RESETS THE POINTS
320 'SET UP FOR DRAWING CIRCLE
330 G=ABS(X2-X1):H=ABS(Y2-Y1):
    W=SQR (G*G+H*H)
340 W=INT (W+.5) 'ROUND OFF
    RADIUS
350 CIRCLE (X1,Y1),W:RETURN
```

PRODUCT REVIEWS

This section is open to all producers and dealers of color computer products. We will review your 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.

STRING VARIABLE EQUATION SOLVER

M. F. Estes Softhead Co. has a program that will solve an algebra equation. For example the program will solve the equation A*X+B=C for X without any numerical variables being given. The program will solve equations composed of the letters A-Z and integers from 0-9.

The program is loaded and run. Options are given for selecting a PRINTER (1), FAST (2), FASTER (3) OR NOTHING (<<ENTER>>). If you just want to solve the equation press "ENTER". The following appears on the screen.

```
 INPUT EQUATION THAT I SHOULD
         SOLVE---
```

The following was entered:

```
 Y+X-3Z=-8
```

The computer prints

```
 FOLLOWING IS THE EQUATION  THAT
IS TO BE SOLVED Y+X-3Z=0-8

WHAT VARIABLE DO YOU WANT TO SOL
VE FOR?
```

When Z is entered the following appears:

```
SOLVING THE FOLLOWING-
    Y+X-3*Z=0-8
     - FOR Z. MY SOLUTION OF THE
EQUATION IS
   Z=+(8+X+Y)/3
```

If you will go through the algebra you can verify that the solution is correct. Let's try another equation.

```
  5W-8P+3Q-5=0
```

Solving for P the computer's result is

```
  P=-(5-3Q-5W)/8
```

This is fun let's try another one.

```
 AX-3Y+6=0
```

and the solution for Y is:

```
   Y=+(6+AX)/3
```

The program will also arrange a higher order equation into descending powers from left to right. For example the equation

```
   5X-8X^3+2X^2-9=0
```

is rearranged by the computer as

```
  -8X^3+2X^2+5X-9=0
```

when asked to solve for X.

We tried putting a variable in the denominator with the following equation:

```
  A+B/C*(A+1)=0
```

When asked to solve for C the computer's result was:

```
   C=-B(1+A)/A
```

The computer can generate an equation. For example we ran the program and selected the option for the computer to generate an equation. It generated the following equation:

```
    E*(F+B)-E*A+B=0
```

The solution for A is

```
A=+(B+EB+EF)/E
```

The computer will continue after a pause and solve for another variable or generate a new equation.

We found the program to be excellent for solving equations within its limits. The computer takes a long time to solve some equations. It is fairly fast for the simple equations though and makes a good learning program for those studying algebra.

M. F. Estes Softhead Co., P. O. Box 335, Elba, NY 14058, $19.95 + $3 S/H.

+ + + DCN STAFF + + +

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.

PRINTER RIBBON INK SPRAY

As the demand for printed copies from data increases, so is the cost of fabric ribbon replacement. EBONIZE can greatly reduce these costs by extending the ribbons' useful life. One can of EBONIZE can re-ink up to 20 ribbons. Simply remove the toop of the ribbon cassette and spray evenly and lightly over the coiled ribbon. Its specially formulated ink permeates the ribbon surface by osmosis. After the ribbon is dry, replace the top and you're ready for re-use. Get longer life from your fabric ribbon with EBONIZE.

For more information contact Gold Plug 80-E.A.P. Co., P. O. Box 14, Keller, TX 76248 (817) 498-4242. $14.95

QUESTIONS & ANSWERS

These are questions that have been asked us through letters. If you have a question, please write and we will print your answer here. For $10 we will give you a personal reply to your question. Letters that are of interest to others will be printed here.

Question: I purchased a printer that uses a different type parity than my computer uses. About half of the characters are erroneously printed. What can I do to correct this?

Answer: Most printers have switches for selecting the baud rate and parity if used. I assume you looked for switches. This can be corrected in your computer by modifying the output printer routine to give the parity your printer needs. A machine language subroutine would have to be made and loaded whenever you needed to use your printer. Another approach would be to contact the printer manufacturer and see if they or one of their representatives can tell you how to change the parity.

OPERATING HINT

You can break a program, do memory peeks or pokes, or list part of your program, and then continue the program by typing "CONT ENTER". You do not loose your variables with this procedure.

DCN SUBJECT INDEX

We have listed our subjects by Volume and Issue. Our first issue, Vol 1-1, was February 1984. The first and second year we printed 11 issues each. This listing is complete through Volume 3-4 or May 1986.

Basic Programming

Immediate mode, Vectors 1-1
Variables 1-2
Arrays, Read, Data 1-3
Data Handling Techniques 1-8
Memory Searching 1-9
Random Numbers 1-10, 1-11
FOR- NEXT Loops 2-5
DIM, Arrays, IF-THEN 2-7
Branching, ASCII, Strings, Peeks & Pokes 2-8
Word Processor Development 2-9
LEFT$, RIGHT$, MID$, LEN, VAL commands 2-10
Seperate Data Files 3-1
EXEC Command 3-2
Deleting & Inserting Data in Files 3-3
Editing Statements 3-4

ML Programming

Microprocessor, EXEC 1-1
Indexed Addressing 1-2
Data Relocation & Branching 1-3
Sound Subroutine 1-10,1-11
Bank Switching Subroutine 2-2
Block Move Subroutine 2-3
64K All RAM 2-6
2-Bank Subroutines 2-9
Move Basic Program to Upper Mem. 3-3
ML Programming (Part 1) 3-4

Articles

Memory Expansion 1-2
ASCII 1-3, ASCII & BASIC 1-4,
Interfacing ASCII Devices 1-5
Powerful Remarks-Word Processing 1-5
Uninterrupted Power Sources 1-5
Word Processing 1-6
Computer Generated Sound 1-9, 1-10
Large Memory Programs 2-1 thru 3-4
Computer Graphics 2-1
Writing Programs 2-2
CoCo Heat Problem 2-6
Graphics, Lines, Bar Graphs, 2-8
Large Memory Pgms, Basic Vectors 2-8
Using Page -1 2-9
Circle Command 2-10
Draw Command 3-1
Interfacing Computers 3-2,3-3,3-4
Basic Basic 3-1, 3-2
Graphics Scalling 3-2
Ramdisk Improvements 3-2
Page -1 Program Development 3-4
Developing a Drawing Program 3-4

Programs

Multiprogram Manager 1-1
Utility 1-4
Remark Print (Word Processing) 1-5
Check Book with Data in Remarks 1-6
Memory Search 1-8
Ball Team Sort 1-9
Sound Generator 1-10
Card Shuffling 1-10
Sound Learning 1-11
Bank Switching Program 2-3
Gas Mileage 2-4
Graphics Demo 2-4
Grade Book 2-5
Character Generator 2-6
Alarm Clock 2-6
Address File 2-7
Student Study 2-7
Line Demo 2-7
Vector Corrector 2-8
Fast Food 2-8
Draw Bar Graphs 2-8
Word Processing 2-9
Bar Graph with Character Generator 2-9
Ram Disk 2-10
Recipe 2-10
Electric Cost 2-10
Circle Demo 2-10
Check Book 2-10
Inventory (Strings for Data) 2-11
ARC & Circle Demo 2-11
Ship War Game 2-11
Ram Delete Subroutine 3-1
Draw Demo 3-1, 3-2
Bouncing Ball Game 3-1
File Demo (Seperate Data File) 3-1
Electronic Billboard 3-2
RamDisk Subroutines 3-2
Tanks (game) 3-3
Draw Demo (GET & PUT) 3-3
Move Programs to Upper RAM 3-3
ROULETTE (game) 3-4
RESTORE - Restores erased programs 3-4
Graphic Draw Program 3-4

Hardware Projects

Installing an interrupt Switch 1-4
Video Reverser 2-1
Add a Second Port 2-9

Product Reviews

Spectrum DOS 1.0 2-6
Thunder RAM 2-7
Telewriter Enhancer (Telepatch) 2-8
Lowercase Character Generator (Hardware) 2-8
Basic + 2-9
COCO Calender 2-11
Assembly Language Programming (Book Review) 3-2
Schematic Drafting Processor 3-3
String Variable Equation Solver 3-4

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
1/3	19	17	15
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.

DCN PROGRAMS on Tape or DISK

This is our second collection of programs from Dynamic Color News. This collection includes:

1. Check book program. Data in remark statements. Prints to screen or printer.

2. Ball Team Sort Program. with information on sorting.

3. Card Shuffling Program. (Using Random Numbers)

4. Student Study Program. Randomly picks questions and answers.

5. Address File Program. Print mailing labels, search for address by name, zip code, city, or state.

Order DCN-2

Tape $9.95, Disk $11.95
Add $2 shipping, Foreign $3

Please sign me up for one year for the DYNAMIC COLOR NEWS SERVICE. I Want to receive instruction on programming, Computer Theory, Operating 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