Radio Shack

TRS-80™ Microcomputer NEWS

April, 1981

P.O. Box 2910, Fort Worth, Texas 76101

THE MICROCOMPUTER NEWSLETTER PUBLISHED FOR TRS-80 OWNERS

Volume 3, Issue 4

Newsletter Index In This Issue...

Computer Club

Southern Maine TRS-80 Group
15 Mountain View Road
Cape Elizabeth, ME 04107
207/767-2351 or 207/797-4898

RADIO SHACK ANNOUNCES AGRICULTURE'S FIRST ELECTRONICS MARKETING INFORMATION SERVICE

Tandy Corporation/Radio Shack, in a joint announcement with Professional Farmers of America, in Chicago recently revealed plans for agriculture's first electronic marketing information service.

Called Instant Update, the service provides farmers and agribusinessmen immediate access to the market-making events that affect commodity prices, crop yields and other data important to improving farmers' business activities.

Instant Update information will be transmitted via telephone lines to VIDEOTEX terminals specially made by Tandy Corporation/Radio Shack for the Professional Farmers of America program.

"The Radio Shack VIDEOTEX system was selected because of its technology and cost effectiveness," stated Charles Phillips, senior vice president of Special Markets for Radio Shack. This Radio Shack terminal utilizes standard telephone lines and a standard television set

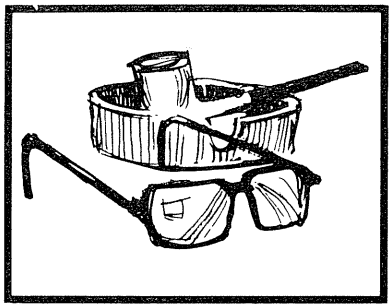
(Continued on Page 20)

Notes on Previous Newsletters

Mr. J. Michael Healy of Birmingham, AL recently sent us this note:

"I would like to tell you how much I enjoy your NEWSletter and I am enclosing a program change for the Base Conversion program (12/80) which makes the binary printout much easier to read. This change inserts a space between each four digits (nibble) of the binary conversion. It involves changing only one line [300] and adding lines [301-306] and [611-613]. I use this program for programming EPABX's (Telephone Computers) and it has made the printout much easier to work with. I have a Model II and a Line Printer IV and am very pleased with both. Remember our battle cry: WE HAVE ONLY BEGUN TO PROGRAM!!!"

(Continued on Page 20)



View From the 7th Floor

by Jon Shirley, Vice President Computer Division

I am going to break my own rule and announce a new product before it's available. Soon we will offer disk drive head cleaner kits for both the 5¼ and 8 inch drives. They will sell for $29.95 each and contain two cleaning diskettes and a bottle of cleaner. Each diskette can be used 13 times so at once a week you get a half years life per kit. There are some other head cleaning kits on the market but we picked this one by getting our 3 major drive suppliers to test every brand we could find and tell us which one absolutely would not damage the head or the pad. They certified only one and that's the one we will offer. I strongly recommend that every disk drive user get a kit and use it. If you are a light user clean the heads every two weeks and get a full year's protection of your data for only thirty bucks. That is cheap protection.

There seems to be a little confusion about the disk drives used in the Model III, judging from some articles I have read and some of your letters. The drives in the Model III are based on the Tandon design but they are manufactured from the ground up at Texas Peripherals in Odessa, Texas. The T.P. drives are 40 track and use a band actuator for fast head stepping. Texas Peripherals is a joint venture (50/50) of Datapoint Corporation and Tandy Corporation. There are 150 employees at Texas Peripherals and it was the first new industrial plant opened there in over 20 years. In case you have never been to west Texas, the whole place floats on OIL and almost all the local industry is concerned with oil exploration, drilling, storage etc.

While on the subject of the Model III I have noticed some enterprising types are now selling built-in drive kits for the Model III. When a Model III is expanded to one (or two) internal drives a lot of stuff is packed inside. Included are a disk controller board, another switching power supply, molded brackets to hold the drives, shielding to protect the drives and to maintain FCC compliance, cables to hook the whole thing up and, of course, the drive(s). If one of these non-Radio Shack kits tempts you to buy, you should be aware of two potential problems. First of all none of these kits are FCC certified as of February 14 when I am writing this. None of them have even been submitted to the FCC. So if you have one of these installed it is very likely that your Model III will not continue to meet FCC regulations. The FCC will enforce its rules and either require compliance or stop the seller but it will take time.

The second problem that you will face is that we will not service your Model III with a brand X controller and drives and power supply inside. We are not being anti-free enterprise, mean or greedy, just practical. When a disk controller is installed it becomes an integral part of the entire computer and can effect its performance even in non-disk operations. Adding a disk controller is NOT like putting different tires on a car, its like replacing the entire intake manifold, carburetor and all. Do it wrong and you will burn valves (or worse). Since all our diagnostics for a disk system are disk based we would have no way of knowing what is creating the problem and of course we do not stock parts for non-Radio Shack products. So if you must buy a brand X kit be sure you can get service!

We hear about all sorts of different applications for the various TRS-80's and from time to time I would like to share some of them with you. For instance there is a talented programmer in Canada who has produced a very good scoring program for sailboat regattas. It runs on a 32K 2 disk Model I and really does it all. We used it here in Fort Worth this fall to score the World Youth championships which were sailed on Lasers and Laser 2s. (Bet you did not know we have a world class Yacht Club in the middle of Texas!) If any of you would like information on this program drop me a line... it's a personal recommendation as I set up the system here.

Another interesting sports application came to our attention from a news release put out by the Milwaukee Bucks of the National Basketball Association. They tell the story as follows.

The Milwaukee Bucks of the National Basketball Association have stepped into the computer age with a microcomputer that continuously compiles, displays, and prints player and team statistics during the course of the game.

The Bucks organization has taken a big step into the modern age this season with the acquisition of a new computerized statistical scoring system. California based Sports Stats U.S.A. has designated their system the SS100 Basketball Scoring System. The SS100 system consists of a standard off-the-shelf office computer purchased from Radio Shack (TRS-80 — Model III) and a special basketball data entry computer keyboard and computer software programs from Sports Stats U.S.A.

The system has virtually replaced pen, pencil and paper; strained eyes and cramped fingers with the miracle of micro-circuitry. For years, keeping stats meant hash marks on a scorecard, adding columns, and filling out a box score by hand. The next morning, after the game, it was back to the office, pencil in hand, erasers aplenty, to add up team and individual highs and lows, check career marks, type up the results, and otherwise spend hours in the pursuit of a peculiar American sports pre-occupation known as "statistics."

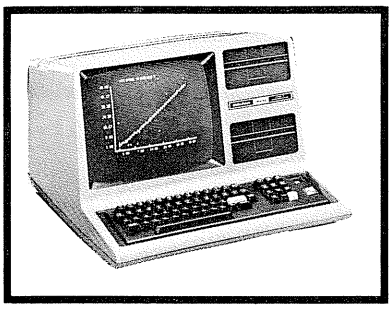
This year, in contrast, all of the above is being done with the touch of a button. Stats are totaled and printed in box score form seconds after the game-ending horn. Later, the computer can produce all the cumulative and comparative data, as well as season highs and lows, in printed form suitable for reproduction and use in future press notes and news releases. Cumulative statistics are now compiled and printed in the time it used to take to complete a box score. Accurate up-to-the-second game statistics are displayed by the computer on video monitors to assist radio and television broadcasters in calling the game. During timeouts, and between quarters, quickie stats are printed and distributed. Comprehensive box scores are printed at half time and post game.

"The system is light years ahead of how we used to compile statistics," said Bucks' Publicity Director Bill King II. "The amount of information immediately available both on video monitors and through printed reports was heretofore not possible. But with the aid of the Sport Stats system we can virtually overwhelm even the most statistical oriented sports reporters and fans."

"The system has a very high profile courtside while it saves valuable office time for me," continued King. "The computer has allowed us to redefine the role and importance of keeping statistics by minimizing the vast amount of time required to compile statistics."

According to Sport Stats spokesman, Jim McCabe, the real advantage and revolutionary nature of computerized scoring lies in the inherant speed, accuracy, and video display capabilities of the computer.

(Continued on Page 17)



Model I/III

Product Line Manager's News

When you read my first comment that the Model III included the "best" of Model I plus a lot more it was a cold January. We started looking then at the additions to the Model III and now we are already into the Spring. I will have to stop this discussion soon because we have a stack of new products to introduce to you, but let's cover some more Model III additions.

DISK BASIC

Editing your programs is a snap with:

↑	Prints the previous program line on the screen
↓	Prints the next line
.	Prints the current line
,	Edit the current line
(SHIFT) ↑	Prints the first line
(SHIFT) ↓ Z	Prints the last line (left (SHIFT) key only)
Lxx	List line xx
Exx	Edit line xx
Dxx	Delete line xx
Axx, inc	Auto insert beginning at line xx

DOS

The DO file is a time saver and is also one of the most versatile tools that we have added for the applications programmer. You will find plenty of uses for this special file which is first created with the BUILD command. Put TRSDOS library commands or applications programs in the DO file in the order that you want them to execute. Then kick off the sequence with DO"filename" where "filename" is a DO file previously developed with BUILD. It's great for things like preparing a series of listings while you are at lunch, killing transaction files, initializing a communications channel before starting a session, beginning an application by setting FORMS for the printer, or initiating WP for protecting program files. Of course the major benefit of the feature is to insulate an operator from the processing procedures that would normally be required to use a business application. DO files also insure that the procedures are followed properly, cutting down on frustration and lost time. Here is an example of the way it would work:

First from TRSDOS Ready, type BUILD EXAMPLE. You will then be prompted to type in up to 63 characters on each line to describe commands, or to provide text that will be used for input to a running program. When you are finished exit with the (BREAK) key and the commands will be saved in a "DO file" (which in this case is called EXAMPLE). You could have next typed AUTO DO EXAMPLE — Now when you reset or power up, a DO activity will begin. The "DO file" EXAMPLE provides automatic command input and will execute each command shown without any intervention from the keyboard (other than stopping for your INPUT requests from BASIC). Note that BASIC applications can be run from a DO file including automatic input of information normally entered from the keyboard. In those cases, the responses to the BASIC applications questions thru use of BASIC INPUT statements are saved in the DO file along with the name of the applications to be run.

Take a look: (For clarity, we underlined the parts which you would enter from the keyboard.)

```
BUILD EXAMPLE (ENTER)
Hit BREAK to exit
Type in up to 63 Characters
SETCOM (BAUD=1200,WAIT) (ENTER)
Type in up to 63 Characters
FORMS (WIDTH=80) (ENTER)
Type in up to 63 Characters
PAUSE INITIALIZING SERIAL INTERFACE AND
  PRINTER (ENTER)
Type in up to 63 Characters
BASIC (ENTER)
Type in up to 63 Characters
(ENTER)
Type in up to 63 Characters
(ENTER)
Type in up to 63 Characters
RUN"HISTOGRA/BAS" (ENTER)
Type in up to 63 Characters
(BREAK)
TRSDOS Ready
```

Now, to see how it works type:

```
DO EXAMPLE (ENTER)
TRSDOS Ready
SETCOM (BAUD=1200,WAIT)
BAUD = 1200, WORD = 8, STOP = 1, PARITY =
  NONE, WAIT MODE
TRSDOS Ready
FORMS (WIDTH=80)
TRSDOS Ready
PAUSE (INITIALIZING SERIAL INTERFACE AND
  PRINTER)
INITIALIZING SERIAL INTERFACE AND PRINTER
Press <ENTER> to continue
(ENTER)
TRSDOS Ready
BASIC
How Many Files?
Memory Size?
TRS-80 Model III Disk BASIC Rev 1.2
(c)(P) 1980 by Tandy Corp. All Rights
  Reserved.
Created 5-Jul-80
21706 Free Bytes   3 Files
READY
>RUN"HISTOGRA/BAS"
                    HISTOGRAM
```

...(The program would continue from here, with inputs from the keyboard as usual.)

MODEL I/III Potpourri

A few comments... In answer to some of the predominant questions coming in about VisiCalc. You can use the Model III CONVERT utility on a Model I VisiCalc disk to transfer VisiCalc data files that were originally set up on your Model I. But transfer them to a Model III VisiCalc disk. Model I VisiCalc was not designed to run on a Model III and it won't. Accessing VisiCalc data files from BASIC is described in a document available at no charge as 700-2220. It is a VisiCalc addendum describing the Data Interchange Format... The LPC printer driver is now on the Model III TRSDOS disk as well as available for Model I (700-2007). Make sure that you read the Peripheral Product Line Manager's page this month for an understanding of the purpose of the LPC driver... A recent comment was made that RS software packages include instructions on user changes to make them run on Model III's. Not so — We have an exchange program available if you get a package without a III compatible tape or disk (if it is needed). In fact we have reworked our stock to include these tapes and disks so you won't need to get involved in any "user" changes. We do include

(Continued on Page 4)

Model I/III (From Page 3)

a pamphlet with your Model III entitled "Instructions for Converting Specified Model I Programs for use on TRS-80 Model III." The changes in that pamphlet are "optional," since tapes or disks are available (with the changes already made) for those programs which are shown. The changes are given for your convenience since many of you already own our most popular cassette software and may prefer to modify these yourselves . . . If you have used the SAVE command with the ASCII option to save a BASIC program—SAVE filespec, A—rather than the compressed format then you may have gotten the error message " DIRECT STATEMENT IN FILE ." This is caused by lines which exceed 255 bytes. Make sure you save your program first without the ,A option as you are liable to lose your program otherwise. To avoid the problem remember that the 255 byte maximum includes the line number digits and the space following the line number, all the characters and spaces in the line itself and 2 bytes for each line feed.

Model I/III Bugs, Errors and Fixes

Model III TRSDOS 1.2

On page 79 of the Model III TRSDOS manual, it states that the number given to the "LINES = 'bbb'" option is "the maximum number of lines to print before an automatic form feed . . .". This is not so. In Model III TRSDOS 1.2, when the maximum number of lines is reached, the line counter restarts at one and NO automatic eject is done. The lines counter serves only to tell the printer driver how many lines have been printed since the last top-of-form. It is the responsibility of the programmer to examine the line counter at location 4029 Hex (16425 decimal) to determine if it is time to send the printer driver a top-of-form character (decimal 12) to eject the paper.

The manual is also in error when it states, about the LINES option, ". . . If omitted, 60 is used . . .". If no entry is made, whatever the maximum lines counter has in it will remain unchanged. The power-up default is 60, but after power-up, if the value is changed for any reason, you will need to specify 60 if that is the number of lines you need.

Also in the Model III TRSDOS 1.2 manual, under the section on Fundamental TRSDOS I/O Calls, pages 124 and 125, it should be noted that the OPEN and INIT TRSDOS I/O calls destroy the contents of the IY register. This is a bug in TRSDOS 1.2 and is scheduled to be corrected in the next release of Model III TRSDOS. Until then, if you are using the TRSDOS I/O calls in your machine language routines, remember to save and restore the contents of IY before and after a CALL to INIT or OPEN.

Tape Mailing List (26-1503)

In versions 3.0 and earlier, when printing labels the linefeeds are off and the labels print incorrectly.

To correct this problem, make the following changes to the program:

```
19010 LPRINTFI$(I,2):LPRINTFI$(I,3); "   ";
      FI$(I,4);"-";FI$(I,5): LPRINT" ":
      LPRINT" "
19015 IFE=0THENLPRINT" ":ELSEIF
      FI$(I,1)=""THENLPRINT" "
```

Standard & Poor's Stockpak™ (26-1507)

You may receive a "FILE FULL" message when you try to add the 31st stock.

To correct this, follow this procedure:

1) Go to the security entry part of the program and DELETE the 31st ticker that you just added.

2) ADD a ticker XXX, with a dividend rate of 0.00, X-date 00/00/00, current price 0.000, PE Ratio 0.00, S & P index 0.00

3) After entering this dummy ticker, you should be able to proceed as usual.

Accounts Payable (26-1554)

In version 3.0 only, the program will not print or subtract the discount from the amount of the Cash Requirement Report. The checks are subtracted when printed.

To correct this problem, make the following changes to the REPORTS program:

```
173 IV$=I1$:D=ABS(CVI(I2$)):GOSUB61:DA$=D$:D=
    ABS(CVI(I3$)):DU=D:GOSUB61:DU$=D$:D=CVI
    (I4$):DD=D:
    GOSUB61:DD$=D$:V$=I5$:GOSUB57:
    A1=A:V$=I6$:GOSUB57: A2=A:V$=
    I7$:GOSUB57:A3=A:V$=I8$:GOSUB57:A4=A:
    V$=I9$:GOSUB57:A5=A
```

Disk Payroll (26-1556)

Computer Services has received new state tax tables for Mississippi, Nebraska and New York. If you need these new tax tables, contact Computer Services (Address and phone numbers on the back page of the Newsletter).

Business Mailing List (26-1558)

New version 3.0 copies of this program should be available. These new versions include the ability to print two "up" (across) labels. The Model I version is 700-2211 and the Model III version is 700-2212. This new version is available free to current owners of 26-1558. You will be required to exchange your ORIGINAL program disk for the new version diskette.

Real Estate Vol. I (26-1571)

In the "Rate of Return" program, the modified internal rate incorrectly equals regular internal rate of return.

To correct this, the following change should be made to the "Rate of Return" program:

```
800 R1=R1+R2:D9=0:D0=C:D0=D0-(D0*R3):FORI=
    1TOD1-1: ONAGOTO810,820
```

Advanced Statistical Analysis (26-1705)

In Time Series II, a "subscript out of range error" may occur.

To correct this, make the following changes:

DELETE line 1090

```
1060 GOSUB6000:JJ=0:FORJ=1TO8:JJ=JJ+1:IFZ
     (J)="@" N(K)=M+JJ-1:JL=J:GOSUB7000:JJ=
     0:K=K+1:IFMT=3M=M-JL
10090 E=0:KL=1:FORK=1TONT:E=E+1:IFZD(K)=
      "@"THEN KL=KL+1: E=E-1:GOTO10100:ELSE
      IFZI="Y"LPRINT "ELEMENT #";E,"GROUP #";
      KL,ZD(K)
10095 PRINT"ELEMENT #";E, "GROUP #"
      ;KL,ZD(K):GOSUB9900
```

Note that we added a large section in the middle of line 10090 and "chopped off" the end of line 10095.

Johnson/Caire Screen Print Routine

Mr. John P. Caire, Jr. of Hamden, CT recently purchased a Line Printer IV for his Level II 48K Model I TRS-80. Mr. Caire has provided us with a modified version of Myron F. Johnson's Screen Print Routine (Jan/Feb 1980). Mr. Caire's modification corrects a small problem which the original version had, allows it to print lower case, automatically puts the printer in 132 column mode for the graph, then returns the printer to 80 column mode.

Here is the Johnson/Caire routine:

```
2200 K%=0
2210 FOR Y%=0 TO 47
2220 P$=""
2230 FOR X%=0 TO 127
2240 IF POINT(X%,Y%) THEN P$=P$+ "#" ELSE
     P$=P$+ " "
2250 NEXT X%
2260 IFINT((Y%/3)-.1)=(Y%-1)/3 THEN GOTO 2330
     ELSE 2270
2270 LPRINT CHR$(27); CHR$(20); P$
2280 NEXT Y%
2290 K%=0
2300 LPRINT CHR$(27); CHR$(19)
2310 RETURN
2320 END
2330 K%=K%+1: Q$="": R$=""
2340 FOR L%=0 TO 63
2350 A%=PEEK(15360+ L%+ (64* (K%-1)))
2360 IF A%<=32 THEN A%=32:GOTO 2380
2370 IF A%=>128 THEN A%=32
2380 Q$=Q$+ CHR$(A%)+" "
2390 NEXT L%
2400 FOR M%=1 TO 128
2410 IF MID$(Q$, M%, 1) <> " " THEN R$=R$+
     MID$(Q$,M%,1) ELSE R$=R$+ MID$(P$,M%,1)
2420 NEXT M%
2430 LPRINT R$
2440 GOTO 2280
2450 END
```

If you have an upper-case only printer, Mr. Caire suggests you add line 2365:

```
2365 IF A%=> 97 AND A%<= 127 THEN A%=A%-32:
     GOTO 2380
```

Happy graphics!

CLOAD KBFIX

Mr. Bob Pollock sent us this information:

"For those of us that still have problems with the old ROM Level II keybounce problem, here is your KBFIX program rewritten in BASIC. On power-up, respond to MEMORY SIZE with 32712. CLOAD this BASIC program, then run it. If you have keyed in the program correctly, the computer will go directly into SYSTEM mode. Activate the debounce program by responding /32713.

"This is (a) faster loading program than the machine language version."

Here is Mr. Pollack's program for 16K Level II Model I's:

```
5 REM KBFIX PROGRAM FOR 16K, LOAD AND RUN TO
  FIX KEY BOUNCE.
10 DEFINT A-Z
20 FOR I=1 TO 55
30 READ J
40 POKE 32712+ I, J
50 K= K+ J
60 NEXT I
70 IF K<> 4864 THEN PRINT "BAD LOAD" ELSE
   CLS: SYSTEM
80 END
90 DATA 175, 17, 10, 0, 205, 11, 0, 25, 34,
   22, 64
100 DATA 205, 97, 27, 195, 25, 26, 33, 54,
    64, 1
110 DATA 1, 56, 22, 0, 10, 95, 174, 115,
    163, 32, 8
120 DATA 20, 44, 203, 1, 242, 226, 127, 201,
    95, 197
130 DATA 1, 220, 5, 205, 96, 0, 193, 10,
    163, 200, 195
140 DATA 251, 3
```

Sort Modification

Mr. Konrad Kuzmanoff of Ohio State University sent us this modification to our machine language sort routine (July 1980):

"I found the machine language sort published in the July issue of the TRS-80 newsletter most useful. I especially appreciate the additions that you mention in the November issue.

"You may be interested in printing the following modification which allows one to load the sorting routine from the DOS. The modification protects the sort routine by changing the high memory pointer at &H4049. To put the sort on disk:

1. Load and run the BASIC program.
2. Return to the DOS. (CMD"S")
3. Call up DEBUG. (DEBUG (ENTER))
4. Use the memory modification to insert the following code at &HFFCB (MFFCB):

   21 FF FE 22 49 40 C3 2D 40

5. Return to the DOS (G402D (ENTER))
6. Call up TAPEDISK. (TAPEDISK (ENTER))

   Note: You will still be in DEBUG. To execute TAPEDISK, press (G) (ENTER). The TAPEDISK prompt (a question mark '?') will appear.

7. Under TAPEDISK type:

   F SORT/CMD:0 FF00 FFD5 FFCB (ENTER)

8. Return to the DOS. (E (ENTER))

"The sort routine is now saved as SORT/CMD, and can be loaded from the DOS simply by entering the command 'SORT.' The sort routine will be loaded, high memory will be set to protect it, and you will be returned to the DOS."

Editors Notes: This modification is for Disk based Model I computers only. Model III Disk users should check their TRSDOS manual for information on using the DUMP utility to transfer the machine code from RAM to disk. Please note that in the TAPEDISK command above, 'FF00' is the START address, 'FFD5' is the END address, and 'FFCB' is the TRAnsfer address. There is no relocation of the code.

The modification given above works for a 48K system. If you have a 32K system change the third pair of values in the memory modification from "FE" to "BE." You should also change the 'FF's in the TAPEDISK command to 'BF's. For a 16K system, change the 'FE' to '7E,' and change the 'FF's to '7F's.

The modification allows you to load the machine language program directly from DOS, without having to use the DATA statements. To see that this works, modify the sample program given in the July News by DELETEing line 60 and all lines from 1000 to the end of the program (make sure you save a BACKUP of the original program, just in case something goes wrong). This will allow you to execute the sort demonstration program, but the program will never have to call the sort load subroutine.

From DOS, follow this procedure after you have created SORT/CMD and modified the demo program:

1. Type: SORT (ENTER)
2. When DOS READY appears type: BASIC (ENTER)
3. Press (ENTER) for both the Files and Memory Size questions.

(Continued on Page 6)

Sort Modification (From Page 5)

4. Type: RUN"SORT/BAS" (ENTER) (use your program name!)

I used this change in a 48K Model I and the program handled 1500 strings very well. This change lets you forget about Memory Size, which I find to be a very useful feature.

If you would like to use this procedure with the two dimensional sort routine we published in March 1981, here is the information you need:

For a 16K machine, the code to add using M7FEE is:

```
21 DE 7E 22 49 40 C3 2D 40
```

For 32K change the '7E' to 'BE' and modify memory beginning at Hex BFEE. For 48K change the '7E' to 'FE' and modify memory beginning at Hex FFEE.

The TAPEDISK command for 16K would be:

```
F SORT2D/CMD:0 7EDF 7FF7 7FEE
```

For 32K, change the '7E' to 'BE' and the two '7F's to 'BF.' For 48K change the '7E' to 'FE' and the two '7F's to 'FF.'

PLANTING SEEDS

The random number generator in the TRS-80 Model I and Model III can be reinitialized or "reseeded" so that the RND function will return a repeatable sequence of random numbers.

One typical use for a predictable sequence of random numbers is computer generated tests. After initializing the random number generator, the sequence of test questions would be determined using the RND function. The correct answer list for each test could then be generated by reseeding the random number generator with the same value used to initialize it.

Reseeding the random number generator simply requires POKEing three memory locations with a number from 0 to 255. All three locations must be POKEd to produce a repeatable sequence. The memory locations are 16554, 16555, and 16556.

The example below shows how it's done:

```
10 INPUT "PLEASE ENTER YOUR NAME";N$
20 IF LEN(N$)<3 THEN PRINT "NAME MUST BE AT
   LEAST 3 CHARACTERS": GOTO 10
30 GOSUB 300: 'Initialize random number
   generator
40 FOR X=1 TO 20
50 PRINT RND(1000),
60 NEXT X
70 PRINT: PRINT
80 GOSUB 300: ' Re-initialize random number
   generator
90 FOR X=1 TO 20
100 PRINT RND(1000),
110 NEXT X
120 END
300 'Routine to reseed random number
    generator
310 FOR Z=1 TO 3
320 POKE 16553+ Z, ASC(MID$(N$,Z,1))
330 NEXT Z
340 RETURN
```

Quick Adder

Here is a portion of a letter we received recently from Richard Moskovitz of Carnegie, PA:

"...I just had need to add several short columns of figures, and went to the adding machine to do the job. After adding only 3 of the 50 or more columns, I thought there must be a better way. I went to my trusty TRS-80 and wrote the short program attached. I have the 10-key pad attached so input is very quick. While there is nothing spectacular in the program, it filled a need that I had right then. The program is as follows:

```
100 CLS
110 LINEINPUT"ENTER A NUMBER ";A$
120 IF A$="" THEN B=0 ELSE B= B+ VAL(A$)
130 PRINTTAB(35) "SUBTOTAL"; USING
    "######,###.###";B
140 GO TO 110
150 END
```

"As you can see from line 110, I have Disk BASIC. (I also have SCRIPSIT which I find to be fantastic).

The reason I use A$ as a string for the input is so I can test for a null enter.

"I have to add, at different times, items that have thousands only and also items that have dollars and cents. I find that the ### following the . in line 130 allows for the inputting of the thousands without repeating all the zeros. In other words, 10,000 is input as 10. A figure like 10,245 would be input as 10.245. Now the decimal point in the answer is used for the comma of the thousands. When I am adding dollars and cents, the extra zero (third place of the decimal value) is not annoying.

"The purpose of testing for a null string (in line 120) is to allow for the totaling of one set of figures and the start of the next. A (ENTER) only, without any figure input, will cause the value of B to be reset to zero allowing for the start of a new column of figures..."

Editor's Note: For non-Disk BASIC users, change line 110 to read:

```
110 A$="": INPUT"ENTER A NUMBER "; A$
```

String Along With STRING$

The following article is reprinted from the Jan. 1981 newsletter printed by the Radio Shack Computer Center in South Rochester, NY:

One statement, which is often overlooked in LEVEL II BASIC, is STRING$. The format of STRING$ looks like this:

STRING$(n, c)

where "n" is a number which indicates how many times you want a character repeated and "c" describes the character. The character can be identified in two ways. It can be identified by using the ASCII code number for that character (par ex: A = 65) or it can be the character itself, enclosed in quotes ("A"). Thus, either of these expressions would print the character "A" 48 times:

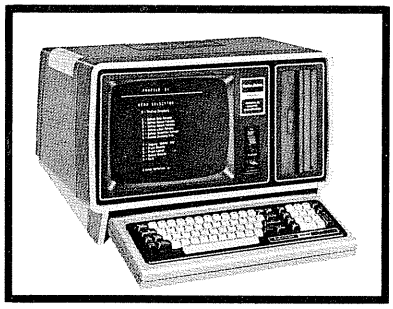
```
PRINT STRING$(48, 65)
PRINT STRING$(48, "A")
```

The statement can be put to very practical use if you wish to draw a bar chart which illustrates numeric data. For example, try this:

```
10 CLS: CLEAR 64
20 INPUT "HOW MANY HOURS DO YOU WORK PER
   DAY"; W
30 INPUT "HOW MANY HOURS OF TV"; T
40 W=W*2: T=T*2: S=48-(T+W)
50 CLS: PRINT@ 448, STRING$(S, "Z");
   STRING$(W, 140); STRING$(T,176)
60 PRINT "SLEEP"; TAB(S) "WORK"; TAB(S+W)
   "TV"
```

Obviously this little program does not accurately graph what you do all day, but it illustrates a point — that STRING$ can be used to draw clever graphs. In this program, the "day" has been scaled up so that 48 positions on the screen represent the 24 hours of the day. Thus in line 40, each one of your entries is multiplied by 2 to bring them up to scale. The remainder of the day, which we assume is devoted to sleep, is calculated by subtracting the sum of your (doubled) entries from 48. The numbers 140 and 176 which appear inside the parentheses in line 50 produce graphics characters which print narrow, solid lines.

Editor's Note: STRING$ is available in Model I Level II and Disk BASIC, Model III BASIC and Disk BASIC, Model II BASIC, Model II Compiler BASIC, and Extended Color BASIC.



Model II

Product Line Manager's News

INTERFACING PROFILE AND SCRIPSIT.

If you have at least two disk drives on your MOD-II and own either PROFILE or SCRIPSIT, do yourself a favor and look into the capabilities of the other. If you do not have an external drive, you may find that the combination of PROFILE / SCRIPSIT could make the purchase of a second drive a worthwhile investment.

PROFILE has the ability to spin off a SELECT file of merge data from your data base. A SELECT file is a special file which can be used by SCRIPSIT for merging information into standard forms or letters. To do this you specify the header names that you want to use as SCRIPSIT keywords, then the field numbers of the data you want blend into a SCRIPSIT form or letter.

Now you select the records from PROFILE by up to sixteen criteria and create a disk file that contains the SCRIPSIT key words and the data to be merged.

An example? OK. Lets use PROFILE as a membership, dues, mailing list. Your PROFILE records might contain a FIRST NAME, LAST NAME, ADDRESS, CITY, STATE, ZIP, MEMBERSHIP NUMBER, DATE DUE, DATE LAST PAID, AMOUNT DUE and AMOUNT LAST PAID. We don't want or need all of this information to appear in the letter so the SELECT format would be:

```
FNAME
LNAME
ADDRESS
CITY
STATE
ZIP
DATE DUE
AMOUNT DUE
```

When you specify which data is to be spun-off, you would select on AMOUNT DUE GT 0 (greater than zero). If PROFILE finds any records with the AMOUNT DUE greater than zero it will write the items specified in the SELECT format to the MERGE file. The MEMBERSHIP NUMBER, AMOUNT LAST PAID, DATE LAST PAID etc. would not be in that MERGE file.

Now, to SCRIPSIT for the form letter. The letter would be typed into SCRIPSIT as follows:

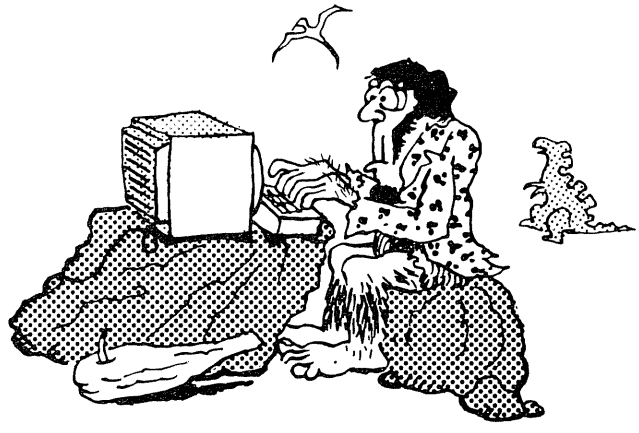
```
MR. {LNAME}
{ADDRESS}
{CITY}, {STATE} {ZIP}
Dear Mr. {LNAME}
    Our records indicate that your membership dues of
{AMOUNT DUE} were past due on {DATE DUE}.
    Please remit your payment to keep your membership in
good standing.
THANK YOU,
Club Treasurer
```

Printing the letter will give you a proof copy exactly as you type it in, including the braces and LNAME. These are the same keywords you used in the selection format of PROFILE.

Instead of printing the letter, you can ask for a merge. SCRIPSIT will ask for the filename of the MERGE file. In this case, it would be the PROFILE select file but could be the MAILING LIST (26-4507) or a data file from a program you have written.

After you enter the file name and press (ESC), SCRIPSIT will scan the merge file, match the first block of keywords with the form letter in SCRIPSIT for position within the data block then read in the next data block, the keywords in the form letter are replaced with data from PROFILE, the letter is reformatted to compensate for the different lengths of data, such as a long or short last name, and the letter is printed.

This advantage becomes even more obvious if the names are used in the body of the letter. Each letter will appear as if it were custom typed for each person it is sent to, but you only typed the letter one time. That same letter is still available to be used next month. Updating your PROFILE data base and running the SELECT program will provide you with the next current set of data for the letter. Also, the letters could be printed in alphabetical or zip code order if you requested the data to be sorted.



Model II Bugs, Errors and Fixes

TRSDOS 2.0 AND 2.0a

The following sets of patches should be made to Model II TRSDOS 2.0 and 2.0a:

1. For some ranges of values, the "RAM Directory" Supervisor call (#53) has returned an erroneous value for the number of records in the file. This is caused by an invalid offset in one of the operands used in the calculation.

The following patch is for TRSDOS 2.0 and 2.0a only and will correct the code involved in the calculation. All three patches must be applied:

```
PATCH SYSTEM/SYS R=59 B=103 F=B52811 C=CD9926
PATCH SYSTEM/SYS R=59 B=120 F=CF5DEB C=CD9F26
PATCH SYSTEM/SYS R=60 B=154
                 F=E5E5E5E5E5E5E5E5E5E5E5
                 C=B5CA7A252BC9CF5D13EBC9
```

2. In some cases, when using the PATCH utility to modify a byte, an erroneous 'String not found—Abort' message would be returned, even when a LIST of the file would confirm the presence of the byte to be patched at the address specified. This was caused by an invalid compare when the target byte was at the beginning of a sector.

To observe the problem, attempt the following patch:

```
PATCH PATCH A=2989 F=7E C=7E
```

(Continued on Page 8)

Mod II Bugs, etc. (From Page 7)

(The '7E' byte is the first byte in the second sector of PATCH.)

The following patch is for TRSDOS 2.0 and 2.0a only and will correct the symptom described above.

```
PATCH PATCH A=3263
   F=B79932A5323AA432321933CD
   C=9132A5323AA432321933B7C4
```

When a file uses the last (96th) directory entry, it is possible, in some cases, to be able to see the file listed in the directory, but any attempt to access the file by name will result in an invalid error 24: 'File Not Found.'

The following patch is for TRSDOS 2.0 and 2.0a only and will correct the symptom described above. Both patches must be applied:

```
PATCH SYSRES/SYS A=1682 F=5F C=60
PATCH SYSRES/SYS A=1699 F=03 C=00
```

General Ledger (26-4501)

In version 1.2, an extra Top-of-Form may occur when you are doing a two page asset and liability balance sheet.

To correct this, change line 1275 of the "Glbalsht" program to read:

```
1275 IFM=2ANDL>4THENP=1:GOSUB430:P=60
```

In versions 1.0 — 1.2 of General Ledger, if you wish to change the month that your system is running in, use the following procedure:

1) Go into Accounts Maintenance
2) Press the (BREAK) Key
3) Type FM= the # of the month (ENTER)
4) Type CONT (ENTER)
5) Press the (F1) key
6) Go to System Status and check the month to be sure it is correct.

If the General Ledger's year is closed out, and it was not in the 12th month, System Status does not reset to one. To fix this problem, follow this procedure:

1) LOAD"Income" (ENTER)
2) Type in 1490 FM=0 (ENTER)
3) SAVE"Income" (ENTER)

Inventory Management (26-4502)

After one year of running Inventory Management, the suggested order list is supposed to be automatically switched to the zero base ordering. We have found that it is two months off in versions 1.0 and 1.1

Make the following changes in "ORDER/BAS":

Version 1.0—

Line 7710 should read:

```
7710 Y=P:IFNY=12THENOP=PH+1ELSEOP=PH-NY:
     IFOP<1THENOP=OP+12
```

ADD line 7715:

```
7715 X=OP
```

Version 1.1—

Line 7710 should read:

```
7710 Y=P!:IFNY=12THENOP=PH+1ELSEOP=PH-NY:
     IFOP<1THENOP=OP+12
```

ADD line 7715:

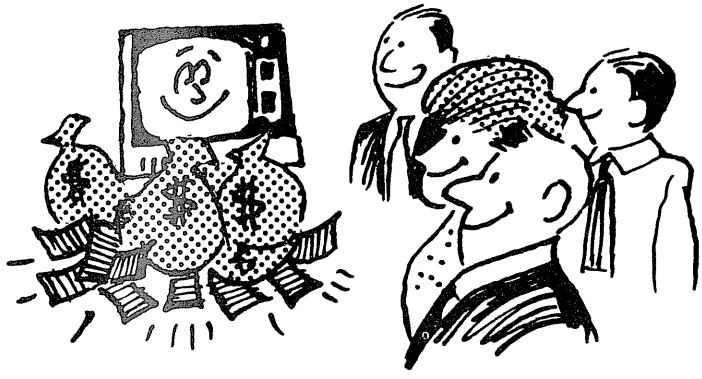
```
7715 X=OP
```

Payroll (26-4503)

Computer Services has received new state tax tables for Mississippi, Nebraska and New York. If you need these new tax tables, contact Computer Services (Address and phone numbers on the back page of the newsletter.)

In version 1.1 of the Payroll program, the employee's subject FICA wages will be incorrect in some cases. In the version 1.1 "W2" program, change line 1700 to read:

```
1700 LPRINT:LPRINTTAB(6);NA$;TAB(40);
     USINGFI$; RE#(1);:LPRINTTAB(55);
     USINGFI$;RE#(1)-RE#(5)
```



Accounts Payable (26-4505)

When a vendor is completely paid off, the balance in the account should be zero. Instead, it may be one cent. Check all invoices and make sure that they are correct.

To correct the problem, change the following lines in the "APCHECK/BAS" program:

```
700 W#=ABS(N#)*100:V$="":X=W#/D1#:W#=W#-
    X*D1#: V$=V$+CHR$(X-(N#<0)*128)
702 X=W#/D2#:W#=W#-X*D2#:V$=V$+CHR$(X)
704 X=W#/D3#:W#=FIX(W#-X*D3#):
    V$=V$+CHR$(X)+CHR$(W#):RETURN
```

Scripsit (26-4530)

If you have version 1.0 of Scripsit, there are two sets of changes which should be made. The first problem is that the Repaginate function may wipe out and/or scramble your documents. The second problem is that when an alignment tab is terminated with an (ENTER) instead of a period, Scripsit will not accept the (ENTER) as a valid command.

To fix these problems, follow this procedure:

1. Reset the computer and insert a Scripsit diskette.
2. Answer the DATE prompt with the current date.
3. Press the (HOLD) key.
4. Answer or skip the TIME prompt.
5. The repaginate problem is fixed with the following four patches:

```
PATCH SCRIPSIT/6 A=E094 F=0F C=05
PATCH SCRIPSIT/6 A=E235 F=C6566F C=C310DC
PATCH SCRIPSIT A=DC10 F=0000000000000000
                      C=BF2006210057C3
PATCH SCRIPSIT A=DC17 F=0000000000000000
                      C=3DE2C65667C338E2
```

6. The alignment tab problem is fixed by the following two patches:

```
PATCH SCRIPSIT A=9CD8 F=FDBE16 C=C320DC
PATCH SCRIPSIT A=DC20 F=0000000000000000
                      C=FDBE16CAEB9CFE0D
PATCH SCRIPSIT A=DC29 F=000000000000
                      C=CAEB9CC3DD9C
```

Model II Ideas

Kenneth Willoughby of Fairacres, NM sent us the following letter and program:

"Dear Editor,

"I am enclosing 2 software programs that may interest your readers. These are for the TRS-80 MODEL II. One is a cheap WORD-PROCESSOR by which I am writing this letter using the EDITOR functions to correct mistakes and so on.

"Each instruction line could have 2½ lines of instruction to fill out the logical record but this would goof up the automatic page and line count in the FORMS control when running the program and printing on the line printer (I have lineprinter II). To begin my letter, I use only the video until satisfied with the letter program. There it can be listed, deleted, and merged with the subject matter of previous letters in cannibal fashion.

"The second program is a MODEL II video graphics program that I developed. It seems to teach the MODEL II graphics ability as well as being fun to play to test your skill. I was disappointed at (the) TRS-80 graphics book. It did not give me any inkling of the power built into the MODEL II system. I had to stumble through letting the powerful ERROR MESSAGES of MODEL II teach what to do next. This is what makes me love the MODEL II more than any computer that I have ever used, from the IBM 610 to the UNIVAC 1108 and CDC 6600. This second program is a simple shooting gallery or a radar search type problem . . .

"P.S. Why doesn't RADIO SHACK package and market more systems catering to helping handicaps conquer their mobility and communication barriers through electronic means? You have all the components in your RADIO SHACK catalog. Combine a computer, ultrasonic burglar detector, and speech synthesizer to help a newly blinded person learn room-object locations. Combine a computer, voice synthesizer, and VOXBOX to help a deaf person to speak over the phone as well as receive speech. Combine a computer with a telephone modem to help a paraplegic work out of his home minimizing his mobility problem which is what I am doing."

Program 1

```
100 LPRINT
110 LPRINT"TRS-80 Microcomputer NEWS"
120 LPRINT"c/o Editor"
130 LPRINT"P.O.BOX 2910"
140 LPRINT"FORT WORTH,TEXAS 76101"
150 LPRINT
. . .
550 SYSTEM"FORMS T"
```

Note: What Mr. Willoughby has done is to create a BASIC program, in which each line of the program is a line of text in his letter. By making each line fit the video display, he is assured that his letter will fit his LP II, and he knows in advance exactly how the letter will look, which allows him to make effective use of hyphenation. As he pointed out in his letter, BASIC's editor is very powerful and should make editing this type of letter fairly easy.

Program 2

Here is Mr. Willoughby's second program, with a couple of minor modifications which make it play faster and fit our print format:

```
10 PRINT"---TARGET GAME FOR ONE PLAYER ---"
20 REM: THIS TARGET GAME IS ONE PLAYER WHICH
        WILL HAVE 5 TARGETS (O) AND
30 REM: HAVE 10 SHOTS WITH ELEVATION= 0-20 &
        AZIMUTH= 0-60 CHOSEN EACH SHOT
40 REM: TARGETS (O) AND SHOTS (X) WILL BE
        DISPLAYED AS WELL AS SHOT COUNT
50 PRINT"LIST 10 TO 90 FOR INSTRUCTIONS"
60 SYSTEM"PAUSE"
100 CLS: RANDOM
110 FOR Z= 2 TO 7
120   REM ROW
130     M(Z)=INT(RND(20)+.5)
140   REM COLUMN
150     N(Z)=INT(RND(60)+.5)+10
160   REM TARGET DISPLAYED
170     PRINT@(M(Z),N(Z)),"O"
180   NEXT Z
190 FOR J=0 TO 20
200   REM RIGHT VERTICAL
210     PRINT@(J,10), CHR$(151)
220   NEXT J
230 FOR K= 10 TO 70
240   REM BOTTOM HORIZONTAL
250     PRINT@(20,K),CHR$(143)
260   NEXT K
270 FOR J=0 TO 20
280   REM LEFT VERTICAL
290     PRINT@(J,70), CHR$(151)
300   NEXT J
310 C=0: D=0: R2=10
320 PRINT@(0,0), "GUN SET?"
330 PRINT@(2,0),"ELEV?";R1;
340   PRINT@(2,0),;
350   INPUT"ELEV";R1
360 PRINT@(3,0),"AZ?";R2-10;
370   PRINT@(3,0),;
380   INPUT"AZ";R2
390 R2=R2+10
400 PRINT@(R1,R2),"X"
410 PRINT@(5,0),"SHOT #";C+1
420 FOR Z=2 TO 7
430   REM SHOT HIT
440     IF R1=M(Z) AND R2=N(Z) THEN
          PRINT@(M(Z),N(Z)), CHR$(159)
450   NEXT Z
460 C=C+1
470 IF C=10 THEN 490
480 GOTO 330
490 PRINT"AMMO GONE!"
500 END
```

Mysterious Line Feed

Mr. Ray B. Blessum of South Pasadena, CA sent us this letter concerning a problem with our Quick Printer II:

"The following program (throws) in an unwanted line feed:

```
1 LPRINT"123456789012345678901234
456789012
2 LPRINT"See what I mean?
RUN
123456789012345678901234567890123456789012

See what I mean?
```

"But this works OK:

```
1 LPRINT"123456789012345678901234
456789012See? No unwanted line feed.
RUN
123456789012345678901234567890123456789012
See? No unwanted line feed.
```

"I thought you and your readers would like to know.

"Or is there a better way. There might be several 32 character lines in succession."

Mr. Blessum is the victim of a "hidden" linefeed. Here is what happened (in the first example):

1. The printer correctly printed the first 32 characters in line 1.

2. As soon as the 32nd character was printed, the PRINTER issued an automatic carriage return since it was at the end of its line.

3. The computer then sent the carriage return which follows every LPRINT statement.

(Continued on Page 20)

Screen Dump, Revisited

Here is a portion of a letter from L.M. Goddard of Exeter College in England:

"...I noted with interest the screen dump routine (for Model II), written by Mr. Merrill E. Eastcott, Jr., which appeared in your September 1980 issue.

"I enclose a variation of his program which I believe to be a little simpler and more convenient to use. It can be used at any time, whether in BASIC mode *or* DOS mode. It does not require to be MERGEd with a resident BASIC program..."

SCREEN DUMP UTILITY FOR TANDY TRS80 MODEL II

"The following machine language program provides a very convenient means of obtaining a screen dump to printer. It can be used when the machine is in either BASIC mode *or* DOS mode.

```
                                ORG     F000H
F000    21 20 F0                LD      HL,F020H
F003    16 4F                   LD      D,4FH
F005    1E 00                   LD      E,00H
F007    3E 0B           LOOP    LD      A,0BH
F009    43                      LD      B,E
F00A    0E 00                   LD      C,00H
F00C    CF                      RST     8
F00D    3E 13                   LD      A,13H
F00F    06 4F                   LD      B,4FH
F011    0E 0D                   LD      C,0DH
F013    CF                      RST     8
F014    1C                      INC     E
F015    7B                      LD      A,E
F016    D6 18                   SUB     18H
F018    C2 07 F0                JP      NZ,LOOP
F01B    C9                      RET
```

1. The program is most easily entered using DEBUG, but can, of course, be entered in assembly code and assembled.

If using DEBUG to enter the program, use DUMP to store it on disc with START= F000 and END= F01B (Program name —SCRDMP/CMD).

2. Enter the following program in BASIC.

```
100 SYSTEM "LOAD SCRDMP/CMD"
110 DEFUSR=&HF000
120 X=USR(0)
130 END
```

Save program with the name SCRDMP/BAS.

3. The utility can be used at any time, as follows:

(i) BASIC mode RUN"SCRDMP/BAS" (ENTER)
(ii) DOS mode SCRDMP/CMD (ENTER)

UNPATCH/BAS

Mr. Ken Snapp, Jr. of Denver, Colorado sent us this information:

"I have been using the Model II PATCH utility extensively and soon discovered the need for a method of reversing PATCHes without laboriously retyping the find and change strings. Consequently, I developed this short BASIC program which reads a BUILD-file containing PATCH commands and changes the F= sequence to C= and vice versa..."

Instructions for the use of UNPATCH/BAS:

1. Rather than enter PATCH commands directly from the keyboard, create a BUILD-file containing these commands. This procedure is required if you want to later reverse the PATCHes but is best in any case.

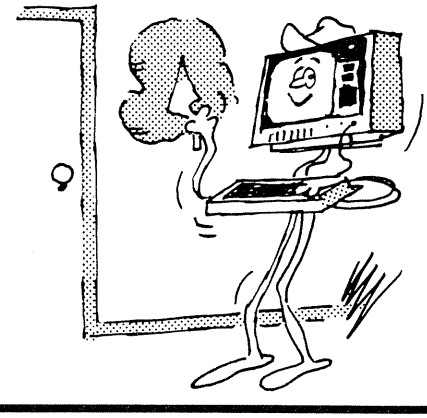
2. Type in the BASIC program from the listing and SAVE it (I use the filename UNPATCH/BAS).

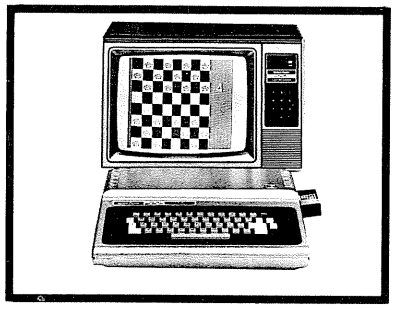
3. Enter the command line:
BASIC UNPATCH/BAS-F:1 (ENTER)

4. The program will ask for the Filespec of the BUILD-file which contains the PATCHes to reverse, simply enter the desired filespec.

```
1 REM ****************************************
2 REM UNPATCH/BAS - REVERSES THE FIND AND
                    CHANGE STRING
3 REM               SEQUENCES IN A BUILD-FILE
4 REM               CONTAINING PATCH COMMANDS
5 REM BY KEN SNAPP, JR.
6 REM   DENVER, COLORADO
7 REM ****************************************

10 CLS: DEFINT R: DEFSTR A
20 DIM A1(4)
30 A1(0)="P": A1(1)="A": A1(2)="T":
   A1(3)="C": A1(4)="H"
40 INPUT"Enter Filespec of BUILD-file
   containing PATCHes to Reverse ";AF
50 RL=0: RN=0: R1=1: R2=0: R3=0
60 OPEN"D",1,AF,1
70 FIELD 1, 1 AS CH$
80 RE= LOF(1)
90 GOSUB 4000
100 IF RC=34 THEN R3=R1: R2=0: GOSUB 3000
110 IF RC=13 THEN R1=1: R2=0: GOTO 150
120 ON R1 GOSUB 1000, 2000
130 RL=RN
140 AL=CH$
150 IF RN< RE THEN 90
160 CLOSE 1
170 INPUT"PATCH Reverse Complete, Restart it?
    (Y/N)"; AA
180 IF AA="Y" THEN 40
190 END
1000 REM TEST FOR "PATCH" COMMAND
1010 IF CH$=A1(R2) THEN R2=R2+1 ELSE R2=0
1020 IF R2=5 THEN R1=2: R2=0
1030 RETURN
2000 REM TEST FOR FIND/CHANGE SEQUENCE
2010 IF CH$<>"=" THEN 2060
2020 IF AL="F" THEN LSET CH$="C": GOTO 2050
2030 IF AL<> "C" THEN 2060
2040 LSET CH$="F"
2050 PUT 1, RL
2060 RETURN
3000 REM BYPASS DATA ENCLOSED BY QUOTES
3010 GOSUB 4000
3020 IF RC<>13 AND RC<>34 THEN 3010
3030 R1=R3
3040 RETURN
4000 REM GET NEXT CHARACTER
4010 RN=RN+1
4020 GET 1,RN
4030 IF CH$=" " THEN 4010
4040 RC=ASC(CH$)
4050 RETURN
```





Color Computer

Product Line Manager's News

Before we get into the meat of the fun "stuff" this month, here are a few facts for the hungry mind . . .

If you happen to be using the new Line Printer VII with your color computer and are playing Chess (26-3050), Checkers (26-3055), or the new Backgammon (26-3059), be sure that the printer is turned off. It seems that the coding for the programs sends "garbage" across the serial port which the printer picks up as data. When the buffer is full, the printer prints 80 characters of "garbage." Nothing is hurt, just your nerves when you're in hot pursuit of the opponents King and suddenly the printer starts chattering at you.

The next piece of news is the Square Root function in standard Color BASIC (or should I say the lack of one). The RSC-4 catalog claimed a SQR function was available in standard Color BASIC, but it is only available in Extended Color BASIC. This being the case, here is a short subroutine you can use to figure square roots in your programs:

A= your number
B= square root (A)

```
10 B=A/3
20 B=.5*(B+A/B)
30 C=B*B
35 C=INT(C*10000+.5)/10000
40 IFC<>A THEN GOTO 20
```

In last month's article, I gave you a peek at the wonderful world of Radio Shack's Extended Color BASIC. Now I would like to continue with a little more in-depth explanation of some of the instructions contained in the Extended Color BASIC.

Before we get into the heart of the program (yes, we're going to write a program) we have to tell the computer what to do. These will be the first program statements in our program.

```
5 CLS            'This instruction clears the video screen.
10 PCLEAR 4      'This instruction clears 4 "pages" of
                  memory for graphics.
15 PMODE 3,1     'This instruction sets the computer into
                  the 128 x 192 graphics mode with four
                  colors available. It also tells the computer
                  to start the display on the first page of
                  memory.
17 PCLS          'Even though we have already cleared
                  the screen in line 10, we need to clear
                  the "screens" on the memory "pages"
                  that we are going to use. That is what we
                  are doing here.
20 SCREEN 1,1    'With this line, we are telling the
                  computer to switch to the graphics
                  display mode and use the alternate color
                  set (buff, magenta, cyan, orange).
```

This is what your program should look like so far:

```
5 CLS
10 PCLEAR 4
15 PMODE 3,1
17 PCLS
20 SCREEN 1,1
```

If you were to run the program now, all you would see is a quick flicker of the screen and the OK printed at the top of the screen. Not very impressive, is it? Well, hold onto your hats, it's going to get better. Try putting an endless loop at the end of your program. (i.e. an endless loop like `30 GOTO 30`; get the picture?). Now run the program again. The green background is replaced with an off-white background and nothing else showing on the screen. For the time being, suffice it to say that what you are seeing is the background color (cyan, in this color set) which you wanted PCLS'd in line 17 of your program. To get back to your text screen and the balance of the "yet-to-be-written program," simply press the (BREAK) key. That is the red key in the upper right hand corner of the keyboard. Do a LIST and your program is listed on the screen.

Starting at line # 30 (we don't need the endless loop any more) we can begin to create our own dream house using the LINE command. Think of the screen as a grid having 256 spots across (X coordinates) and 192 spots down (Y coordinates). Using the various combinations of X and Y coordinates and connecting them using the LINE command, you can draw a house, a garage, driveway, sidewalk, the horizon, windows, doors (both for the house and for the garage), bricks on the house, shingles on the roof, . . . get the idea?

The general format for LINE is:

```
LINE (X1,Y1)-(X2,Y2),a,b
```

X1 and Y1 are the X, Y coordinates for the start of the line and X2, Y2 are the end point of the line. "a" is either PSET if you want to draw the line using the foreground color (be able to see it once it is drawn) or PRESET (draw an invisible line) using the background color. The "b" is for those of you who want to get fancy. It can be B or BF. "B" will use the two X, Y coordinates as diagonal corners and create a box, while the "BF" will use the two X, Y coordinates as diagonal corners and create a box, then fill the box with the foreground color.

Note: If X1 and X2 are the same or if Y1 and Y2 are the same then all you will get is a straight line with no box.

(You can control the foreground—background colors by using the COLOR statement, but that is another article.) Again, start at line # 30 and begin to develop your "dream house." If you're having trouble, look at the program listing at the end of this article. There are a lot of program lines there, but if you stop between line 190 and 200, put in an endless loop like, `195 GOTO 195`, you should get the idea. Please notice that I haven't used the "B" and "BF" functions in my program. Why don't you try it, since I don't want to spoil all the fun of programming in Extended Color BASIC.

```
5 CLS
10 PCLEAR 4
15 PMODE 3,1
17 PCLS
20 SCREEN 1,1
30 LINE(0,185)-(255,185),PSET
40 LINE(50,135)-(200,135),PSET
50 LINE(50,135)-(50,85),PSET
60 LINE(150,135)-(150,85),PSET
70 LINE(200,135)-(200,105),PSET
80 LINE(50,85)-(150,85),PSET
90 LINE(150,105)-(200,105),PSET
100 LINE(70,65)-(170,65),PSET
110 LINE(170,85)-(220,85),PSET
120 LINE(50,85)-(70,65),PSET
130 LINE(150,85)-(170,65),PSET
140 LINE(150,105)-(170,85),PSET
150 LINE(200,105)-(220,85),PSET
160 LINE(170,65)-(190,82),PSET
170 LINE(220,85)-(240,102),PSET
175 LINE(190,82)-(190,85),PSET
180 LINE(240,102)-(240,127),PSET
```

(Continued on Page 12)

Color (From Page 11)

```
190 LINE(200,135)-(240,127),PSET
200 FOR X=62 TO 77:
    LINE(140,X)-(145,X),PSET:NEXT X
210 FOR X=62 TO 77:
    LINE(145,X)-(148,X-3),PSET:NEXT X
220 LINE(140,62)-(145,59),PSET
230 LINE-(148,59),PSET
240 LINE(155,135)-(155,115),PSET
250 LINE(195,135)-(195,115),PSET
260 LINE(155,115)-(195,115),PSET
270 FOR X=115 TO 135 STEP 5:
    LINE(155,X)-(195,X),PSET:NEXT X
280 FOR X=122 TO 123:
    LINE(160,X)-(170,X),PSET:
    LINE(180,X)-(190,X),PSET:NEXT X
290 LINE(174,127)-(176,127),PSET
300 Y=85:FOR X=50 TO 70 STEP 4:
    LINE(X,Y)-(X+100,Y),PSET:
    Y=Y-4:NEXT X
310 Y=105:FOR X=150 TO 170 STEP 4:
    LINE(X,Y)-(X+50,Y),PSET:
    Y=Y-4:NEXT X
320 LINE(95,135)-(95,118),PSET
330 LINE(105,135)-(105,118),PSET
340 LINE(95,118)-(105,118),PSET
350 PSET(103,126,8)
360 LINE(55,130)-(90,130),PSET
370 LINE(55,130)-(55,118),PSET
380 LINE(90,130)-(90,118),PSET
390 LINE(55,118)-(90,118),PSET
400 LINE(110,130)-(145,130),PSET
410 LINE(110,130)-(110,118),PSET
420 LINE(145,130)-(145,118),PSET
430 LINE(110,118)-(145,118),PSET
440 LINE(63,100)-(67,100),PSET
450 LINE(63,100)-(63,105),PSET
460 LINE(67,100)-(67,105),PSET
470 LINE(63,105)-(67,105),PSET
480 LINE(80,100)-(85,100),PSET
490 LINE(80,100)-(80,105),PSET
500 LINE(85,100)-(85,105),PSET
510 LINE(80,105)-(85,105),PSET
520 LINE(108,100)-(113,100),PSET
530 LINE(108,100)-(108,105),PSET
540 LINE(113,100)-(113,105),PSET
550 LINE(108,105)-(113,105),PSET
560 LINE(126,100)-(131,100),PSET
570 LINE(126,100)-(126,105),PSET
580 LINE(131,100)-(131,105),PSET
590 LINE(126,105)-(131,105),PSET
600 LINE(155,135)-(105,145),PSET
605 LINE(82,150)-(20,165),PSET
610 LINE(195,135)-(60,165),PSET
620 LINE(0,165)-(20,165),PSET
630 LINE(60,165)-(255,165),PSET
640 LINE(95,135)-(30,150),PSET
645 LINE(105,135)-(65,145),PSET
650 LINE(65,145)-(105,145),PSET
660 LINE(30,150)-(82,150),PSET
670 LINE(0,115)-(50,115),PSET:
    LINE(240,115)-(255,115),PSET
680 GOTO 680
```

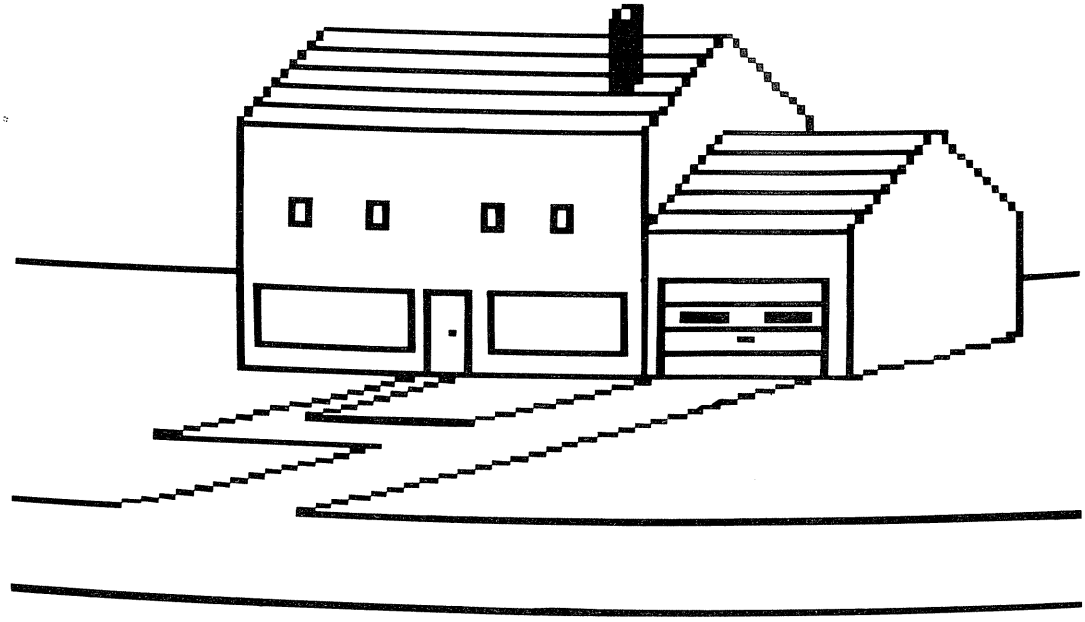
Lines 5-20 set up the computer to display graphics.
Lines 30-190 outline the house and attached garage.
Lines 200-230 draw the chimney.
Lines 240-290 create the garage door with windows and handle. (I decided my garage will have a four-section door.)
Lines 300-310 put shingles on the house and the garage.
Lines 320-350 give me a front door with a handle.
Lines 360-590 draw in the picture windows on the ground floor and the bedroom windows upstairs.
Lines 600-670 create the sidewalk, driveway, and the horizon behind the house.

Now, for the creative and fancy readers. Since lines 360-590 are drawing squares and rectangles, each using four program lines, let's let Extended BASIC do some of the work for us. Delete those lines (360-590) and add these instead:

```
360 LINE(55,118)-(90,130),PSET,B
400 LINE(110,130)-(145,118),PSET,B
440 LINE(63,100)-(67,105),PSET,B
480 LINE(80,100)-(85,105),PSET,B
520 LINE(108,100)-(113,105),PSET,B
560 LINE(126,100)-(131,105),PSET,B
```

*For the real fancy reader, add a "F" after the "B" in each of these lines to color the windows.

In just 6 program lines we have accomplished what 24 program lines did in the original program. Now, how about a workshop . . ., or a storage shed . . ., or another wing to the house . . ., or a three car garage . . ., or . . ., well enjoy LINE and Extended BASIC. 'Til next month . . .



Color Castle

We recently received a letter from Mr, Ken Clause of Metuchen, NJ. The letter contained two parts, the first part was a modification to Dec. 1980's Christmas Tree program, and the second part was a rewrite of the Level I Castle Shot Program (Fire When Ready, Gridley; Page 212 of the Model I Level I manual). Here are portions of his letter:

"The December, 1980 issue of "Microcomputer News" listed a program (page 14) "A Tree for your Color Computer." This program did not run on my color computer or the unit in my local computer center.

"I have re-written lines 10 to 110 to correct the problem, and modified line 470."

We re-entered the tree program here, using the listing from the Dec. newsletter, and had no problems with it. We suspect that the program ran correctly, and Mr. Clause decided that "there must be some mistake." Which only goes to show that each of us has slightly different tastes in programs. Here are the "corrected" lines which Mr. Clause provided (They work very well):

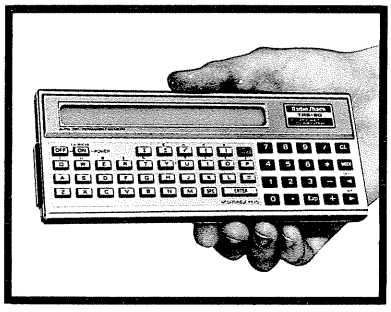
```
5 CLS
10 FOR X=3 TO 479 STEP 96
20 PRINT@ X,"MERRY CHRISTMAS FROM US ALL!"
25 FOR T=1 TO 300: NEXT T
30 NEXT X
40 FOR X=35 TO 510 STEP 96
50 PRINT@ X,"HAPPY NEW YEAR TO YOU ALL!"
55 FOR T=1 TO 300: NEXT T
60 NEXT X
70 FOR X=75 TO 460 STEP 96
80 PRINT@X,"FROM TRS-80"
85 FOR T=1 TO 300:NEXT T
90 NEXT X
'00 FOR X=1 TO 1000
 10 NEXT X
```

Lines 120 to 460 were unchanged.

```
470 GOTO 470
```

Here is Mr. Clause's version of "Fire When Ready, Gridley" for the 4K Color BASIC computer:

```
5 CLS(0)
10 INPUT"ENTER YOUR INITIALS"; A$
20 CLS(0)
30 Z=42
40 FOR Y=12 TO 31
50   FOR X=Z TO 63
60     SET(X,Y,2)
70     NEXT X
80   IF Y<14 THEN Z=Z+2
90   NEXT Y
100 FOR X=42 TO 63 STEP 6
110   SET(X,11,3)
120   SET(X+1,11,3)
130   NEXT X
140 Q=0
150 PRINT@ 280,A$ "'S";
160 PRINT@ 312, "CASTLE";
170 FOR Y=16 TO 19
180   SET(63,Y,2)
190   NEXT Y
200 FOR Y=16 TO 19
210   SET(62,Y,2)
220   NEXT Y
230 FOR X=46 TO 49
240   SET(X,9,7)
250   SET(X,10,7)
260   NEXT X
270 FOR X=40 TO 54
280   SET(X,7,7)
290   SET(X,8,7)
300   NEXT X
310 FOR Z=1 TO 2
320   FOR X=2 TO 12
330     FOR Y=26 TO 29
340       SET(X,Y,4)
350       NEXT Y
360     NEXT X
370   FOR X=2 TO 12 STEP 2
380     RESET(X,27)
390     NEXT X
400   FOR X=7 TO 8
410     RESET(X,29)
420     NEXT X
430   FOR X=1 TO 100:NEXT X
440   RESET(40,7)
450   RESET(40,8)
460   RESET(41,7)
470   RESET(41,8)
480   SET(54,7,7)
490   SET(54,8,7)
500   SET(55,7,7)
510   SET(55,8,7)
520   FOR X=1 TO 200:NEXT X
530   PRINT@ 87," BOOOM!!!";
540   RESET(54,7)
550   RESET(54,8)
560   RESET(55,7)
570   RESET(55,8)
580   SET(40,7,7)
590   SET(40,8,7)
600   SET(41,7,7)
610   SET(41,8,7)
620   SOUND 175,1
630   FOR X=38 TO 2 STEP -1
640     P=X-41
650     Y=P*P/63+7
660     SET(X,Y,8)
670     SET(X-1,Y,8)
680     RESET(X+1,Q)
690     RESET(X,Q)
700     Q=Y
710     NEXT X
720   PRINT@ 354,"KAPOW!";
730   SOUND 10,2
740   RESET(8,26)
750   RESET(7,28)
760   RESET(5,29)
770   RESET(10,27)
780   RESET(11,29)
790   RESET(9,28)
800   RESET(11,26)
810   SET(10,25,4)
820   SET(14,27,4)
830   SET(3,24,4)
840   SET(15,29,4)
850   SET(8,30,4)
860   SET(9,31,4)
870   SET(11,30,4)
880   SET(13,31,4)
890   FOR X=1 TO 1700:NEXT X
900   NEXT Z
910 GOTO 5
```



Pocket Computer

Product Line Manager's News

This month, rather than telling you about one of our Pocket Computer programs, I would like to share with you the programs which we have received from readers over the last several weeks:

Programmable Timer

Here is part of a letter from Larry C. Crump of Mineral Wells, Tx.:

"...This is a program to turn the (pocket) computer into a programmable delay timer. I use it as a reminder when I have to go to a meeting later in the day. A time delay from 1 to 999 minutes is entered and at the end of this period the beeper goes off and "announces" the number of minutes chosen have elapsed. The timer has less than 3 seconds error per hour on my computer. I am interested to know if other computers will run the same speed close enough with varying battery condition, etc.

Programmable Timer

```
10 PRINT" PROGRAMMABLE TIMER"
20 PAUSE"INPUT DELAY IN MINUTES"
30 PAUSE" THEN HIT ENTER."
40 INPUT T
50 BEEP 1
60 C=0: M=0
70 IF T<1 THEN 170
80 IF T>999 THEN 180
90 FOR A=1 TO 49
100 PAUSE" "
110 NEXT A
120 C=C+1: M=M+1
130 IF M=4 THEN 200
140 IF C<T THEN 90
150 BEEP 10
160 IF T=1 PRINT" ONE MINUTE IS UP."
170 PRINT T,"MINS. ARE UP."
180 PRINT"MAXIMUM DELAY 999 MINS."
190 GOTO 10
200 M=0: PAUSE " "
210 GOTO 140
220 END
```

"The angular symbol blinks to show the program is running. The clock is based on the fact that about 49 FOR-NEXT loops run per minute. This isn't accurate enough so a "fudge" factor is inserted every time M = 4 by adding another .85 second pause in line 200. Line 70 is included to negate the fact that 0 or negative minutes look like 1 or positive minutes. Line 80 limits the maximum time to a reasonable battery time and machine limits. Two hits (3 for 1 minute input) of the (ENTER) key after the program runs brings it back ready to enter another time delay. I hope this program will time out right on all pocket computers. Thanks for a wonderful product that is a true innovation for the future."

TRS-80 Pocket Computer Telephone Book

Mr. Randy Bonin of Port Arthur, TX sent this program for the Pocket Computer:

"I wrote the following program to keep track of names and telephone numbers I use regularly. It will hold sixty-six entries. Lines 30-50 (first loop) search for the name, if found in files, program exits loop and name/number is printed in line 80. If not in file, a message is printed in line 60. New info is input in line 100. Lines 110-150 (second loop) will check for a name duplication and also check for the first open memory location. If no memory is left, the message in line 160 will be displayed. If a memory location is open the program will then ask for the number in line 190, then the message in line 220 is displayed.

"The program will start in "DEF" mode by pressing (SHFT) (L). New data entry is made by pressing (SHFT) (N).

"Program will hold 66 names and numbers. Hope you enjoy it.

"Note: Name file will only hold the first seven characters."

```
010 "L" REM TELEPHONE LOG 1-6-81 RANDY BONIN
020 I, "ENTER NAME "; A$
030 F, B=27 TO 159 STEP 2
040 IF A$=A$(B) THEN 70
050 N, B
060 P, "SORRY NOT ON FILE"
070 C=B+1
080 P, A$, A(C)
090 "N" REM ENTRY OF NEW DATA
100 I, "ENTER NEW NAME "; A$
110 F, B=27 TO 159 STEP 2
120 IF A$=A$(B) THEN 170
130 C= B+1
140 IF 0=A(C) THEN 180
150 N, B
160 P, "SORRY NO MEMORY LEFT"
170 P, "NAME PRESENTLY ON FILE"
180 A$(B)=A$
190 I, "ENTER NEW NUMBER "; A
200 C=B+ 1
210 A(C)=A
220 P, "DATA RECORDED"
230 END
```

Gas Mileage

I took my Pocket Computer with me to New Mexico recently, and developed this little program for computing cumulative gas mileage along the way:

```
10 "A": INPUT"MILEAGE - START ";M
20 T=0
30 "B": INPUT"GAS IN GAL. ";G
40 T=T+G
50 "C":INPUT"MILEAGE - ENDING ";E
60 N= E- M
70 N=N/ T
80 PRINT"MPG= ";N
```

At the beginning of my trip, I filled the gas tank and entered the starting mileage (run the computer in DEF, use (SHFT) (A)).

Each time I stopped for gas I used (SHFT) (B) to enter the amount of gas that was put in at that stop. To find my trip cumulative mileage, I answered the next question with the current mileage from the odometer.

At the end of the trip, I filled the gas tank and ran the program one last time. "MPG= 27.98" was the result. Looks like it is time for another tune-up.

Memory Aid

This program was written by Computer Marketing Representative Jeffrey W. Levin in our Rochester N.Y. Computer Center. Here are some of his comments:

"The program is called "Memory Aide." It's written to provide assistance in memory notation conversion for anybody who works with assembly language programming for the full-display TRS-80 models . . .

"What my program does is enable you to enter an address in one of the three main notations — decimal, hexadecimal, or MSB/LSB — and get back the other two notations. Additionally, if an address is above 32767, you automatically get the POKE value. You merely select the desired input from the keyboard in the DEF mode using the (SHFT) (D), (SHFT) (H), and (SHFT) (M) keys, and the pocket computer does all the rest . . .

MEMORY AIDE

```
10 "D":INPUT"DECIMAL?";D:A=100:Q=0
12 IFD>65535GOTO10
14 IFD<0GOTO10
20 U=INT(D/256):T=D-256*U
30 W=INT(U/16):X=U-W*16
40 Z=INT(T/16):Y=T-Z*16
50 BEEP1:GOSUBA+W:W$=V$
51 BEEP1:GOSUBA+X:X$=V$
52 BEEP1:GOSUBA+Z:Z$=V$
53 BEEP1:GOSUBA+Y:Y$=V$:IFQ=1GOTO440
54 GOSUB997
55 GOTO10
100 V$="0":RETURN
101 V$="1":RETURN
102 V$="2":RETURN
103 V$="3":RETURN
104 V$="4":RETURN
105 V$="5":RETURN
106 V$="6":RETURN
107 V$="7":RETURN
108 V$="8":RETURN
109 V$="9":RETURN
110 V$="A":RETURN
111 V$="B":RETURN
112 V$="C":RETURN
113 V$="D":RETURN
114 V$="E":RETURN
115 V$="F":RETURN
200 "H":A=300:PAUSE"ENTER FOUR HEX
    DIGITS":PAUSE "   ONE AT A TIME"
210 FORN=27TO30:INPUTA$(N):NEXTN
220 M=30:D=0:FORN=27TO30:GOSUBA:BEEP1:
    D=D+A(N+4)*16^(M-N):NEXTN
230 W$=A$(27):X$=A$(28):Z$=A$(29):Y$=A$(30):
    U=A(31)*16+A(32):T=A(33)*16+A(34)
240 GOSUB 997:GOTO 200
300 IFA$(N)="0"LETA(N+4)=0:RETURN
301 IFA$(N)="1"LETA(N+4)=1:RETURN
302 IFA$(N)="2"LETA(N+4)=2:RETURN
303 IFA$(N)="3"LETA(N+4)=3:RETURN
304 IFA$(N)="4"LETA(N+4)=4:RETURN
305 IFA$(N)="5"LETA(N+4)=5:RETURN
306 IFA$(N)="6"LETA(N+4)=6:RETURN
307 IFA$(N)="7"LETA(N+4)=7:RETURN
308 IFA$(N)="8"LETA(N+4)=8:RETURN
309 IFA$(N)="9"LETA(N+4)=9:RETURN
310 IFA$(N)="A"LETA(N+4)=10:RETURN
311 IFA$(N)="B"LETA(N+4)=11:RETURN
312 IFA$(N)="C"LETA(N+4)=12:RETURN
313 IFA$(N)="D"LETA(N+4)=13:RETURN
314 IFA$(N)="E"LETA(N+4)=14:RETURN
315 IFA$(N)="F"LETA(N+4)=15:RETURN
316 GOTO 200
400 "M":Q=1:A=100
410 INPUT"MSB?",U
412 IFU>255GOTO410
414 IFU<0GOTO410
420 INPUT"LSB?",T
422 IFT>255GOTO420
424 IFT<0GOTO420
430 D=U*256+T:GOTO30
440 GOSUB997:GOTO 400
997 PRINT" ";W$;X$;Z$;Y$;" ";U;" "; T;" ";D
998 IFD>32767LETP=-1*(65536-D): PRINT"POKE
    = ";P
999 RETURN
```

Simple Planetary

Here is a letter from Alan Sheidler of Moline, Illinois:

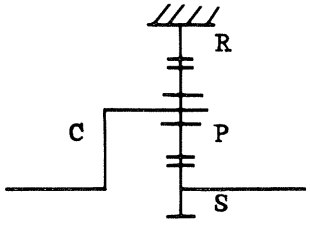
Several months ago I purchased a TRS-80 Pocket Computer. I have found the pocket computer to be a valuable design aid in my work as an engineer for a large machinery company. The pocket computer is a very versatile machine, well suited as a tool for the development of the initial concept of a design.

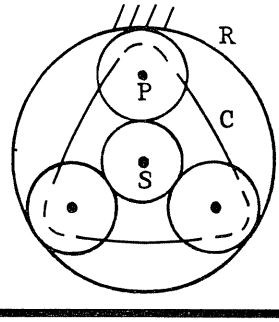
EXAMPLE: Before detailed work can begin on the design of a transmission, a great many gear train proposals must be developed, analyzed, and of course eliminated. Only one proposal will ultimately be adopted. For any particular transmission there are thousands of gear train configurations which are possible. Many parameters are used to judge the virtues of each proposal, but the most important of these is whether or not the available gear ratios match what is desired.

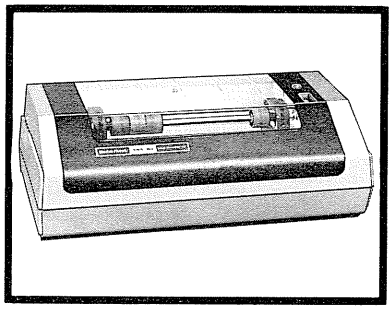
The calculation of gear ratios can be very difficult for some types of complicated gear trains. One type of gear train called a "planetary" gear train is a good example of this. I have written a short program for calculating the gear ratios obtainable with one type of planetary gear train called a "simple" planetary. . . .

PROGRAM

```
10 INPUT "# OF SUN GEAR TEETH ";S
20 INPUT "# OF RING GEAR TEETH ";R
30 INPUT "INPUT GROUNDED ELEMENT ";X$
40 INPUT "WHAT IS INPUT ELEMENT ";Y$
50 IF X$= "R" THEN 90
60 IF X$= "C" THEN 120
70 IF X$= "S" THEN 150
80 BEEP 2:INPUT "REENTER GRNDED ELEMENT "
   ;X$:GOTO 50
90 IF Y$="S" G=R/S+1:GOTO 180
100 IF Y$= "C" G= 1/(R/S+1):GOTO 180
110 BEEP 2: INPUT "REENTER INPUT ELEMENT "
    ;Y$:GOTO 90
120 IF Y$= "S" G= -R/S:GOTO 180
130 IF Y$= "R" G= -S/R:GOTO 180
140 BEEP 2: INPUT "REENTER INPUT ELEMENT "
    ;Y$:GOTO 120
150 IF Y$= "R" G= S/R+1:GOTO 180
160 IF Y$= "C" G= 1/(S/R+1):GOTO 180
170 BEEP 2: INPUT "REENTER INPUT ELEMENT "
    ;Y$:GOTO 150
180 PRINT "GEAR RATIO=", USING"###.######";G
190 GOTO 10
```







Peripherals

Product Line Manager's News

There have been a number of problems related to printer code response involving several Model I/III software packages. I will try to explain here the sources of the problems, the resulting errors and possible corrective actions which might be taken.

The printer drivers which are a part of Model I ROM were tailored to fit the only printer available at the time (LP I), and the Model III drivers were fashioned after the Model I drivers to maintain compatability and assure support for Model I products. That first printer was a very dumb printer and had a rather unusual response to certain codes. Because of this, the ROM printer driver intercepts certain codes normally passed on to the printer and substitutes others designed to achieve the desired response.

The early Radio Shack printer did not have a built-in "top of form" (advance to next page) function. Therefore, the only way to advance to the top of the next page was to issue the proper number of line feed commands to the printer. To date, Radio Shack software does not support any automatic TOF action by a printer.

In order to achieve proper response, both the TOP OF FORM command, CHR$(12) and the vertical tab CHR$(11) are intercepted. The computer compares the number of lines already printed on the current page with the maximum number of lines to print on that page (plus one), and a suitable number of linefeeds are sent instead of the CHR$(11) or CHR$(12). In both the Model I and the Model III, the maximum lines per page is set to 67 at power-up. Note that this value is the maximum number of lines for 11 inch paper (66) PLUS one.

All R/S printers, upon receipt of a Carriage Return (CR or CHR$(13)), not only issue a return to the beginning of the current line but also advance to a NEW LINE (LF). In addition, early models (LP I, II, IV, and some versions of LP III (26-1156A)) ignore any CR issued if there is no text in the print buffer. Thus, the printer ignores all but the first in an unbroken series of CR codes. For this reason, those confangled early printers refuse to respond to a null string, (i.e. — just LPRINT). Since the printer already issues a LF when a CR is received it seems to assume that in a string of line feeds (issued to reach Top of Form for instance) the last LF is a mistake and ignores it. Actually, what really occurs is that no action is taken upon receipt of the line terminator (The last CHR$(13) in a string of CHR$(13)s).

The code for "line feed" is decimal 10 (CHR$(10)). This is supposed to cause the printer to advance one line without changing the position of the carriage. However, the BASIC Interpreter intercepts the "10" (LF) in the expression CHR$(10) and REPLACES IT with a CR (CHR$(13)). (Some RS printers issue a CR as well as a LF when they receive a 10.) IN ADDITION, the "NULL" code CHR$(0) is simply ignored; it is not passed to the printer at all.

This should tell you that if you have a printer which happens to respond "correctly" to the CHR$(10), or reacts to *all* CR codes it receives, you are in trouble!

Clear as mud? There's more to come!

Our first printers only had seven data lines active; the eighth bit (Most Significant Bit) was simply not connected. In order to slip a code "10" past the watchful eye of the BASIC interpreter, some intrepid programmer discovered that CHR$(138) looked like a code "10" to the printer because the missing eighth bit was the difference between the binary forms of 10 and 138.

This was great until a printer came along which used that eighth bit for something. Real trouble! To correct this, we have undertaken the task of establishing codes which can be implemented on all our new printers. Our new standards, now being finalized, reflect a straightforward approach (more about these in future issues). This means, however, that software written for the earlier machines might not print the way they were designed.

ENTER LPC/CMD

LPC is a system (machine language) program which modifies the ROM printer driver in Model I/III. It is designed to allow the system to emulate the response of the first generation printers on a printer having the new revised code response. In other words, LPC throws away all but one CR in a series of carriage returns (ignores LPRINT statements which do not have any text) and converts 138's to 10's.

LPC *might* be necessary with some software when using these current ("new") printers and/or any future RADIO SHACK printer product:

Line Printer III	(26-1156)
Line Printer V	(26-1165)
VI	(26-1166)
VII	(26-1167)

All Daisy Wheel Printers

When do you need LPC? Well, if you run your application and you keep adding a line every time you do a Top of Form then you need it (you are having "Top of Form trouble"); or if you lose line feeds (you are having problems with CHR$(138)).

At this writing to say exactly which Model I/III programs need the driver is like trying to hit a moving target. Many times it has been possible to fix the problem before release of a new version. In many cases, both "fixed" and original versions exist in the field and there may not be an easy way to know which is which. This list of Model I/III software which requires LPC/CMD is current at this time. It might (will!) change:

26-1504 Level II Payroll
26-1506 Cassette Portfolio
26-1507 Standard and Poor's STOCKPAK™
26-1552 General Ledger
26-1553 Inventory Control
26-1556 Disk Payroll
26-1557 Concrete Take-Off
26-1558 Mailing List
26-1559 Manufacturing Inventory Control
26-1560 Fixed Asset Accounting
26-1562 Profile
26-1571 Real Estate Vol. 1
26-1572 Real Estate Vol. II.
26-1573 Real Estate Vol. III
26-1574 Real Estate Vol. IV
26-1603 Budget Management

If you have these programs, and you are having the problems we are describing, use LPC/CMD. If the problems do not appear, LPC is not needed.

MOD I to MOD III conversions which you do yourself may also need LPC. These conversions are those which you might do to move an existing program (yours or ours) from Mod I to Mod III.

LPC is a very shy beast. Once loaded (usually by an AUTO command on your application disk) it burrows into the system and pulls the sand over it's head and disappears. That is, when you load LPC, it checks the computer, moves itself to just below the top of memory, and then resets the Top of Memory pointers in the

(Continued on Page 17)

Peripherals (From Page 16)

odel I to just below itself. When this process is complete, the computer does not know that the program is there! Since the program "disappears," if you attempt to load a second copy of LPC because you forgot the first copy was there, LPC will load a second copy, just below the first one (remember, LPC changed the Top of Memory pointers)! Do this often enough and you will run out of memory.

LPC/CMD for Model I Disk systems is available free from your RADIO SHACK dealer. Order number 700-2007. For Model III Disk users, LPC/CMD is included with TRSDOS 1.2 and will be included with all future Model III TRSDOS releases.

To help you avoid the problems we have been discussing in your own programs, here are a few tips from our application programmers which should eliminate the need for LPC:

1. At the beginning of the program insert the following statement to preset the line count:

```
POKE 16425,1
```

2. To do a TOF, follow this procedure:

```
LPRINT CHR$(12);:POKE
16425, 1
```

(Please note that the semi-colon (;) following the CHR$(12) is extremely important)

3. Do NOT use CHR$(138). If you have used CHR$(138), change it to CHR$(32). CHR$(32) is the code for a blank or space. (If you know that your program will be used ONLY with a "new" printer, you can use LPRINT by itself.) These steps will give you a CR/LF. There is currently no way to send a character 10 to an eight bit printer from BASIC.

Our application programmers assure us that these procedures will work with both Models I and III, all ROM versions and all "known" printers.

There are a few more loose ends which need to be addressed. A glance at some Radio Shack printer manuals will reveal an attempt to provide CR/LF response to CHR$(138). This can of worms should not have been opened and such response is no longer supported.

Because the operating system in MODEL II is in RAM it has been a little easier to address the code response problems there. Normally the Model II BASIC Interpreter completely ignores the CHR$(10). Also, TRSDOS 2.0 has a transparent mode to allow all codes written in BASIC to reach the printer (remember Model II BASIC ignores code 10, so you cannot send a 10 to the printer, even with the transparent mode on). A similar "transparent" driver is being prepared for Model I/III so that the full capabilities of our new graphic printers can be utilized. More about that in a future article.

Is it all clear now? Let me stir it all up again. There is a way, in the case of several R/S printers, to eliminate some of these problems without changing software or installing drivers. Some "new" R/S printers are designed so that they can easily be changed by your service representative. These changes cause the printer hardware to emulate the early machines; these newer printers can be set to allow or dissallow the automatic line feed as well as set or reset the "buffer empty response—throw away a LF" syndrome. The automatic New Line is still a necessary response for Radio Shack software. The second option is of value only in cases where only the older software are going to be used. If you use it, newer software could have problems.

To help make things a little clearer, look at the chart, which shows how our different printers react, and whether their response is changeable.

Note that if the printer response is changeable, the chart shows the way the printer is shipped from the warehouse.

These changes are documented in our Service Manuals. You should let the R/S service center adjust the machine as your warranties are voided when the machine is opened by unauthorized personnel. The ability to make these changes will be a feature of all future Radio Shack Dot Matrix printers as long as there is a perceived need. I'll grant you the value of the ability to defeat the automatic LF but pray for the day we can wave goodby to that "buffer empty" ordeal.

I know that some of this may be confusing. It has been giving the folks here in Ft. Worth a fine time, but we "bit the bullet" and are going with a standard for future products. A little confusion and inconvenience now will buy us a proper standard to work with in the future.

Next month I will have juicy details of our new Plug N Power Controller. And in an upcoming issue (maybe next month) I hope to feature a chart outlining the code response of all our printers, our new printer code response standards, as well as more complete information on the Model I, II, and III printer drivers.

View 7th (From Page 2)

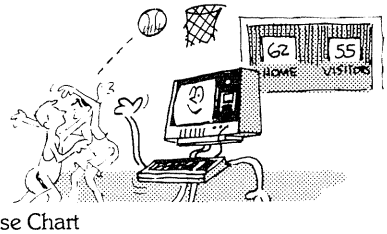
Since the heart of the system is a general purpose office computer, it can also perform important office functions. The Bucks will be utilizing the system in such areas as season ticketholder mailing list, and word processing while another club will be utilizing it for general ledger and payroll.

The Sport Stats SS100 Basketball Scoring System took two years to develop working with several NBA teams. Sport Stats has just completed six months of extensive field testing of the system with the help of the Milwaukee Bucks and is now offering the SS100 for sale to NBA, college and high school teams.

The "Sport Stats SS100 Basketball Scoring System" has been used at each of the Bucks home games this season. Two other NBA teams, the Golden State Warriors and the Phoenix Suns, also have the system installed in their arenas. Eventually, it is hoped that computer stats will be universally accepted on a league-wide basis, helping the NBA step into the computer age.

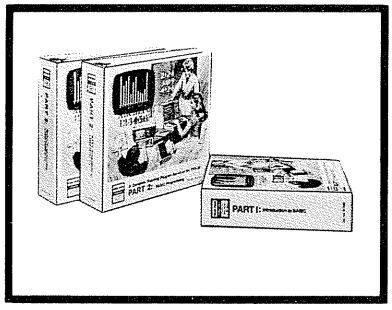
If any of you know of other applications that would be of interest to our readers please send them in. I would like to hear more about the man who has a multi-thousand mailing list on a Model I —on cassettes.

Until next month.



Printer Code Response Chart

	Option 1		Option 2 If buffer empty		Changeable ?
	CHR$(13) is CR/LF	CHR$(13) CR only	CR ignored	All codes executed	
LP I	X		X		No
LP II	X		X		No
Early LP III	X			X	No
Later LP IIIA	X		X		Yes
LP IV	X		X		No
LP V	X			X	Yes
LP VI	X			X	Yes
LP VII	X			X	No
WP 50	X			X	No
DW II	X			X	No



Education

Educational Products News

RADIO SHACK: A COMMITMENT TO EDUCATION

We at Radio Shack believe that along with the development of a practically priced microcomputer, there exists an attendant obligation to make the system and its potential available to the educational community.

To meet this obligation, Radio Shack has undertaken an extensive development effort to produce instructionally sound, effective, and properly validated microcomputer-based educational materials for use in the classroom.

COURSEWARE

What is courseware? It is a special kind of computer program designed to teach. Any computer or microcomputer is useless without a program—a special set of instructions written in a computer language—that tells the computer how to solve a problem, such as calculating a payroll, alphabetizing a list of names, or printing mailing labels. For an instructional application, a computer program must incorporate an additional level of design defining the scope and sequence of the materials to be taught. It must implement some acceptable instructional strategy.

Special courseware such as this is expensive and difficult to develop, and has been a limiting factor in applications of microcomputers in schools.

Radio Shack is addressing the problem of courseware. We have involved the experts — the teachers and curriculum developers in the schools—in designing and field testing our new courseware products. Products such as the Radio Shack K-8 Math Program represent a growing commitment to courseware by Radio Shack.

Our courseware products are divided into three categories: Computer Assisted Instruction, Educational Management Systems, and the Computer Education Series.

The following paragraphs describe some of these new materials.

COMPUTER ASSISTED INSTRUCTION

Computer Assisted Instruction (CAI) is based on the use of the computer as a medium of instruction—as a means of assisting the teacher in teaching subjects such as reading, mathematics, language arts, physics, and chemistry. The Education Division of Radio Shack has developed and made available a variety of CAI materials, and many more are under development.

All of the programs described below require a minimum configuration of a 16K TRS-80 Model I (with Level II BASIC), or a 16K TRS-80 Model III (with Model III BASIC), unless otherwise noted. Each program includes a teacher's manual containing complete instructions. For more information on these programs, see your local Radio Shack store or Computer Center.

AlphaKey™ is an alphabet and keyboard familiarization program for children four to six years old. Parts one and two can teach children to discriminate capital letters and to associate capital and lower-case letters. The letters are presented in sequence so the child becomes familiar with the order of the alphabet. When a letter is displayed on the screen, the child presses the key that matches that letter. Part three gives the child practice typing the letters of the alphabet in order with no hints. A report is given after each session. 26-1718 Suggested Retail Price: $29.95.

Radio Shack's K-8 Math Program, Volume One, is designed to be used as a supplement to regular classroom instruction from kindergarten through eighth grade. Part One of this program is a series of six computer programs containing skill building exercises and a test mode in numeration, addition, and subtraction concepts for use in grades K through 3. Part Two contains skill building exercises, a test mode, and a placement mode for addition, subtraction, multiplication, and division, appropriate for use in grades 1 through 8. Its features include automatic promotion and demotion, appropriate reinforcement messages, a self-pacing feature, a comprehensive reporting function, and more. 26-1715 Suggested Retail Price: $199.00.

The **K-8 Math Program with Student Management, Volume One**, consists of the original Radio Shack K-8 Math Program combined with a student management capability. This management system performs three main functions:

- Maintains a complete ongoing record of all lessons and tests taken by each student.
- Automatically keeps track of lesson promotions and demotions, sending the student to the correct lesson each time he/sh begins a session.
- Allows the teacher to review and print out student reports.

This set of programs requires a minimum configuration of a 32K TRS-80 Model I Disk System (with Level II BASIC) or a 32K TRS-80 Model III Disk System (with Model III BASIC). 26-1725 Suggested Retail Price: $199.00

Radio Shack's **K-8 Math Worksheet Generator** prints worksheets and answer sheets for exercises in addition, subtraction, multiplication, and division, using selected problem sequences from K-8 Math. The teacher can choose the subject, lesson number, and number of problems from each lesson. The worksheets can be identical, or the teacher can choose the number of different worksheets to be printed on the same level of difficulty. The minimum configuration required for this program is a 32K TRS-80 Model I (with Level II BASIC) or Model III (with Model III BASIC), and any Radio Shack Line Printer. 26-2162 Suggested Retail Price: $89.95.

Color Math is a self-paced drill and practice program in addition, subtraction, multiplication, and division for the TRS-80 Color Computer. It contains skill building exercises, a test mode, and a placement mode for each of these four programs. Intended for home use, Radio Shack's Color Math can provide hours of challenge in basic math concepts for students in grades K-8. The minimum hardware requirement for this program is a 4K TRS-80 Color Computer, Tape Recorder, and a color monitor or television. Available May, 1981. 26-3201 Suggested Retail Price: $39.95

The **Essential Math Program, Volume One**, is a self-paced drill and practice program designed to reinforce those math concepts introduced by the classroom teacher. It is a series of eight computer programs containing problems in addition, subtraction multiplication, division, and number concepts for grades seven through twelve. It features skill building lessons, a placement

(Continued on Page 19)

Education (From Page 18)

mode, optional promotion and demotion, reinforcement messages, a comprehensive reporting function, and more. 26-1716 Suggested Retail Price: $199.00.

Radio Shack's **Euclid Geometry Tutor** program is designed to reinforce basic geometry concepts introduced by the teacher. It allows students to practice constructing proofs using nine basic postulates of Euclidean Geometry. Four modes of operation make it possible for students to work at their own pace and to be challenged according to individual ability. The teacher's manual contains complete instructions, plus a Selected Exercise section that gives students an opportunity for drill and practice in a wide variety of problems. 26-1724 Suggested Retail Price: $29.95.

Advanced Graphics is a secondary math program that allows students to study equations in the form $y = f(x)$ and $r = f(\theta)$, or $x = f(T)$ and $y = f(T)$. The equations can be changed for a wide variety of applications. The Selected Investigations section of the teacher's manual gives extensive practice in problem solving. 26-1714 Suggested Retail Price: $29.95.

Vector Addition is a math and physics program for use at the secondary level. The student gives the computer the coordinates of the vectors, and the computer graphs them using either the tip-to-tail or common origin method. An edit feature allows students to add to the list of vectors or to change the vector components. The problem section of the manual covers seven physics topics which use vectors. 26-1720 Suggested Retail Price: $29.95.

Interpreting Graphs in Physics is a secondary math and physics program that allows students to solve problems in kinematics using position vs. time and velocity vs. time simulations. A graph of the data is shown on the video display, and the program generates questions about the graph for drill and practice. Helpful hints are provided. At the end of each session, a student report is displayed. 26-1721 Suggested Retail Price: $29.95.

Graphical Analysis of Experimental Data is a secondary math and physics program that allows students to analyze data using graphs. After the student types in data pairs, the program computes the line of best fit and graphs the data on a Cartesian graph. It also predicts the value of the dependent variable based on the values of the independent variable. A "stopwatch" timer feature for timing experiments is included. 26-1722 Suggested Retail Price: $29.95.

An **Investigation in Integral Calculus** is a secondary math program that graphs and computes areas bounded by any function $Y = f(X)$ and the X-axis. The student supplies the function, the end points, and the number of intervals. Available Summer, 1981. 26-2600 Suggested Retail Price: $29.95.

EDUCATIONAL MANAGEMENT SYSTEMS

Another group of products made available by the Education Division comes under the heading **Educational Management Systems**.

Many of these programs are authoring systems which allow teachers to create their own computer assisted instruction materials. Also included will be programs which assist teachers and administrators with clerical and record-keeping tasks.

Radio Shack's **Quick Quiz** allows teachers to create, store, administer, and print out multiple-choice exercises and tests using a TRS-80 microcomputer. A teacher can type up to forty questions with up to four answer choices per question into the computer and store them on a disk file. The program will then allow the teacher to administer the exercise or test at the computer to a maximum of fifty students and to store each student's score on a disk file. The teacher can also review the test results at any time. This program requires a 32K TRS-80 Model I or Model III disk system. 26-1728 Suggested Retail Price: $29.95.

TRS-80 PILOT PLUS is a command-oriented author language that allows the teacher to create computer assisted instruction materials. It is based on the PILOT computer language, but has extended capabilities for graphics generation and student file handling for daily student sessions. The manual includes a reference section, step-by-step instructions in non-technical language, a sample lesson section, and more. Available May, 1981. 26-2205 Suggested Retail Price: $49.95.

TRS-80/AUTHOR I is a screen-oriented authoring system that allows teachers with no previous computer experience to create sophisticated computer assisted instruction materials. It features full-screen editing and use of graphics, including the special Model III characters. The teacher can provide hints and a glossary to help the students when needed. Also, the teacher can design the lesson so that each student is shown appropriate material based on his or her performance. A student management module allows teachers to enroll students in lessons using assigned passwords. While students are working with the lessons, the program keeps track of overall and partial scores and response times. Special space compression provides efficient storage of lessons on a diskette. TRS-80/AUTHOR I requires a 32K Model I or Model III disk system. Available Summer, 1981. 26-1727 Suggested Retail Price: $150.00.

COMPUTER EDUCATION SERIES

Radio Shack's Education Division also produces materials which teach about the computer itself. The **Computer Education Series** is designed to be used at the secondary and college levels. Each set of materials in this series includes a teacher's manual, 25 student workbooks, and a set of overhead projector transparencies. (Additional workbooks are available separately.) Designed and thoroughly field-tested in schools, this series can be used by all teachers, regardless of their knowledge of microcomputers and programming.

Each lesson in the series has five distinct parts:

- The **OVERVIEW** is read to the students at the beginning of the class session to provide a general orientation on the topics to be covered.
- The **OBJECTIVES** function as a contract with the learners, identifying the skills they will learn if they do their part.
- The **NOTE-TAKING GUIDE** keeps the learner's focus on the important aspects of the lesson. As the material is presented by the teacher, the students complete the guide in the Student Workbooks and then have a set of notes for review.
- The **QUICK QUIZ** allows the students to assess their understanding of each lesson.
- The **ACTIVITY** allows the learners to apply the concepts and techniques in each lesson through a "hands-on" experience with the computer.

Volumes in the Computer Education Series which are currently available include:

PART 1: INTRODUCTION TO BASIC — an introduction to computer programming using the TRS-80 and the BASIC language. 26-2150 Suggested Retail Price: $159.00.

PART 2: BASIC PROGRAMMING — introduces more advanced elements of BASIC including editing, arrays, error messages, memory management, string manipulation, and more. 26-2152 Suggested Retail Price: $199.00.

PART 3: ADVANCED BASIC — introduces the INKEY$ statement, the ASCII character set, action graphics, cassette data files, mathematical and trigonometric functions, multi-dimensional arrays, and more. Available late Spring, 1981. 26-2154 Suggested Retail Price: $199.00.

Additional volumes in this series will be available soon.

We are committed to supporting educational uses of the TRS-80. Look for a growing list of educationally sound courseware from Radio Shack.

Radio Shack®

MICROCOMPUTER NEWS
P.O. BOX 2910
FORT WORTH, TX 76101

ADDRESS CHANGE
☐ Remove from List
☐ Change as shown
Please detach address label and mail to address shown above.

IF UNDELIVERABLE DO NOT RETURN

Elec. Marketing (From Page 1)

to receive and display the information 24 hours a day.

The Professional Farmers of America's version of the Radio Shack VIDEOTEX terminal contains 16K of memory that allows up to 32 screens of information to be stored for instant user recall and analysis. Phillips went on to say, "The Instant Update program is not a test but actually the first commercial application of this type and the largest program of its kind in the United States to date."

Radio Shack's TRS-80 Model II computer was chosen as the connecting link between the central data base and the information user. By operating in a Store/Forward mode the TRS-80 Model II links and dispenses agricultural marketing information in the central data base to the individual TRS-80 VIDEOTEX terminals at the subscriber's home or office. The use of TRS-80 Model II was a key element in the overall cost effectiveness of this system.

The TRS-80 VIDEOTEX terminal has the ability to access other data bases making it more flexible to the user.

Both Radio Shack and the agribusiness community are confident that the daily transmissions of Instant Update which began March 2, 1981 will formally usher in a new era of home/office information retrieval.

Information retrieval technology has been the subject of many small tests and trials throughout the United States. The most notable being the Green Thumb Project in Kentucky in which Radio Shack participated by designing the terminal used in the ongoing 200 farmer test.

Notes (From Page 1)

```
300 BO%=2: GOSUB 360
301 A$= LEFT$(NO$,4)
302 B$= MID$(NO$,5,4)
303 C$= MID$(NO$,9,4)
304 D$= MID$(NO$,13,4)
305 E$= MID$(NO$,17,4)
306 PRINT A$; " "; B$; " ";
    C$; " "; D$; " "; E$
610 NEXT J
611 IF LEN(NO$)=2 OR
    LEN(NO$)=6 OR
    LEN(NO$)=10 OR
    LEN(NO$)=14 THEN NO$=
    "   "+ NO$
612 IF LEN(NO$)=3 OR
    LEN(NO$)=7 OR
    LEN(NO$)=11 OR
    LEN(NO$)=15 THEN NO$=
    "  "+ NO$
613 IF LEN(NO$)=5 OR
    LEN(NO$)=9 OR
    LEN(NO$)=13 OR
    LEN(NO$)=17 THEN NO$=
    "    "+ NO$
```

Mr. Gene K. Lynch of Boulder, MT pointed out an error in the Roman Numeral Conversion Program (Jan. 1981). In line 270, change "D(I)=50" to "D(I)=5:". We appreciate Mr. Lynch's correction.

Line Feed (From Page 9)

4. The result is that two carriage returns were issued and the blank line is created.

5. The printer then went on to print the contents of program line 2.

The reason the second program worked correctly, is that the computer did not send a carriage return until the entire message was printed.

How can you stop this from happening? In the first program, at the end of line 1, add a quote mark (") to end the material to be printed, and then add a semicolon (;). The semi-colon at the end of the first line will suppress (prevent) the automatic carriage return and the print-out will be correct. Now line 1 in the first program looks like this:

```
1 LPRINT"12345678901234567890123
456789012";
```

And the print-out looks like this:

```
123456789012345678901234567890123456789012
See what I mean?
```

COMPUTER SERVICES ADDRESS AND PHONE NUMBERS

8 AM to 7 PM Central Time

Computer Services
900 Two Tandy Center
Fort Worth, Texas 76102

1-800-433-1679 (WATS Except Texas)
1-800-772-5914 (WATS Inside Texas)
1-817-390-3583 (Switchboard)

TRS-80 Microcomputer News is published monthly by Radio Shack, a division of Tandy Corporation. A single one year subscription is available free to purchasers of new TRS-80 Microcomputer systems with addresses in the United States, Puerto Rico, and APO or FPO addresses. Subscriptions to other addresses are not available. The subscription rate for renewals and other interested persons is twelve dollars ($12,00) per year, check or money order in U.S. funds. All correspondence related to subscriptions should be sent to: Microcomputer News, P.O. Box 2910, Fort Worth, Texas 76101.

Back issues of Microcomputer News are not available.

The TRS-80 Newsletter welcomes the receipt of computer programs, or other material which you would like to make available to users of TRS-80 Microcomputer systems. In order for us to reprint your submission, you must specifically request that your material be considered for reprinting in the newsletter and provide no notice that you retain copyrights or other exclusive rights in the material. This assures that our readers may be permitted to recopy and use your material without creating any legal hassles.