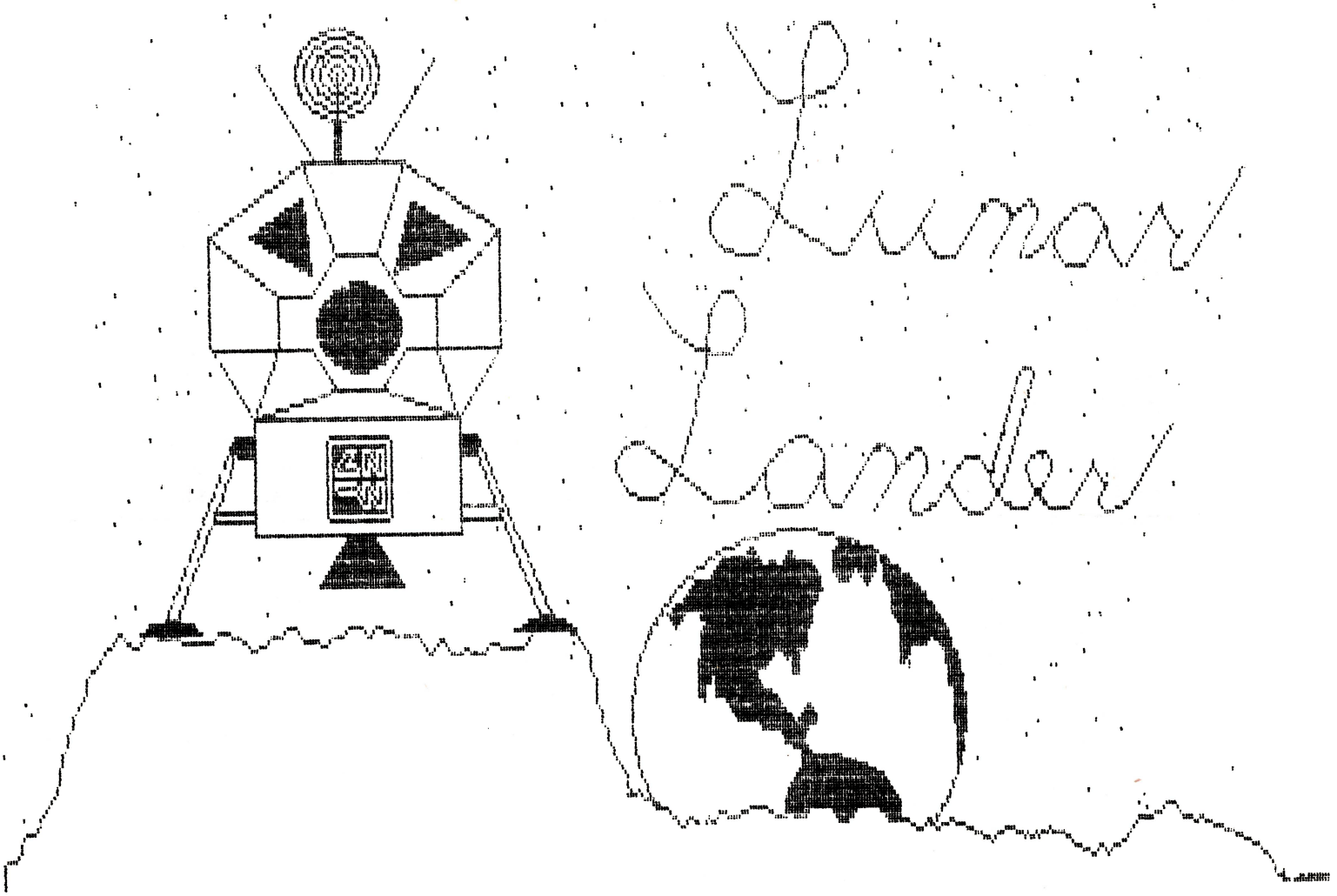
FLASH - WE ARE ABOUT TO BECOME INCORPORATED

Must Re-elect All Officers At The Next Meeting (Aug. 7)

See Details On Page 20



Voice of the '80

Volume 5 Number 3

August 1984

DIALER/CMD

by Dave McGlumphy

Editors Remarks This program by Dave, will autodial your telephone for you using the cassette port. Note the remarks in the source code for hookup instructions. The DEFM (Defined Messages) in the source code can be easily changed to your names and numbers. After all Dave doesn't want you to be calling up all his friends, now does he? By the way this program was sent to me via MCI Mail, an inexpensive message passing service outlined in Northern Bytes, by Jack Decker, Volume 5, Number 2.

DIALER/SRC

```
00010 ;DIALER - Dave McGlumphy 4429 Paula Ln
00020 ;Chattanooga, Tn. 37415   03/26/84
00030 ;MCI# 181-7759
00040 ;This program has model III defaults but also runs on a
00050 ;model 1 without changes.  I don't know about model IV
00060 ;It controls the "REMOTE" plug of the cassette cable
00070 ;Use that plug to control a normally-closed relay which
00080 ;is in series with the red or green wire of the phone
00090 ;line to achieve pulse dialing.  If it dials too fast
00100 ;for your phone system, increase the pause duration
00110 ;at the two places in the code that say "WITHIN DIGIT"
00120          ORG     7000H
00130 BRKROW   EQU     3840H        ;KEYBOARD ROW WITH BREAK
00140 BYTOUT   EQU     33AH         ;DISPLAY SINGLE BYTE
00150 CKKEY    EQU     2BH          ;CHECK FOR KEY-PRESS
00160 CLS      EQU     1C9H         ;CLEAR SCREEN
00170 CURSOR   EQU     4020H        ;CURSOR POSITION LSB
00180 DODOS    EQU     4405H        ;EXIT & DO DOS COMMAND
00190 DOS      EQU     402DH        ;DOS EXIT
00200 DSPLY    EQU     4467H        ;DISPLAY MSG ON SCREEN
00210 LBUFPT   EQU     40A7H        ;LINE INPUT BUFFER PTR
00220 LIPT     EQU     361H         ;BASIC LINE INPUT RTN
00230 PAUSE    EQU     60H          ;(BC) DETERMINES DURATION
00240 PORTF1   EQU     403DH        ;MODEL 1 CASS PORT FLAG
00250 PORTF3   EQU     4210H        ;MODEL III CASS PORT FLAG
00260 START    EQU     $
00270          LD      HL,41A6H     ;PLUG THE DISK BASIC
00280          LD      B,15H        ; LINKS FROM 41A6H
00290 BASLP    LD      (HL),0C9H    ; TO 41E2H
00300          INC     HL
00310          INC     HL
00320          INC     HL
00330          DJNZ    BASLP        ; WITH RETURNS
00340          LD      A,(54H)      ;CHECK MODEL 1/III
00350          DEC     A            ;Z IF MODEL 1
00360          CALL    Z,MODEL1     ;INITIALIZE FOR MODEL 1
00370          LD      HL,LBUFPT    ;SET UP THE
00380          LD      (HL),0CAH    ; LINE
```

```
00390          INC     HL            ; INPUT
00400          LD      (HL),64H      ; BUFFER
00410 MENU     EQU     $             ;DISPLAY THE OPTIONS
00420          LD      HL,SCREEN     ;PT AT MENU SCREEN
00430          CALL    DSPLY         ;PRINT IT
00440 MENGET   EQU     $             ;WHAT FUNCTION?
00450          CALL    GETKEY        ;GET UPPERCASE REQUEST
00460          CP      'A'
00470          JR      Z,ATC         ;YES. PLAY AIR TRAFFIC
00480          CP      'B'           ;B?
00490          JR      Z,BASIC       ;BRING UP BASIC
00500          CP      'C'           ;C?
00510          JR      Z,MODM80      ;BRING UP MODEM80/CMD
00520          CP      'D'           ;D?
00530          JR      Z,DIL68M      ;YES. CALL 6800 BOARD
00540          CP      'E'           ;E?
00550          JR      Z,GETPHN      ;YES. GET KEYED PHONE#
00560          CP      'F'           ;F?
00570          JR      Z,DILCRB      ;YES. CALL CRABAPPLE
00580          CP      'G'           ;G?
00590          JR      Z,DILMCI      ;YES. CALL MCI
00600          CP      'H'           ;H?
00610          JR      Z,DILDCS      ;YES. CALL CHRIS SMITH
00620          CP      'I'
00630          JR      Z,DILWSM      ;CALL BUTCH
00640          CP      'J'
00650          JR      Z,DILPET      ;CALL PETE
00660          CP      'K'
00670          JR      Z,BOOT        ;A HALT INSTRUCTION
00680          CP      'L'
00690          JR      Z,DILWOM      ;DIAL WOMACK'S BBS
00700          CP      'M'
00710          JR      Z,HNG         ;HANG UP
00720          CP      'N'
00730          JR      Z,PIC         ;PICK UP THE PHONE
00740 MENUXT   EQU     $             ;EXIT FROM MENU FUNCTION
00750          JR      MENGET        ;GO TO GET NEXT FUNCTION
00760 ATC      LD      HL,ATCPGM     ;POINT AT ATC COMMAND
00770          JP      DODOS         ;GO DO IT
00780 BASIC    LD      HL,BASPGM
00790          JP      DODOS         ;GO INTO BASIC
00800 MODM80   LD      HL,MODPGM
00810          JP      DODOS         ;EXECUTE MODEM80/CMD
00820 DIL68M   LD      HL,M68PHN
00830          CALL    DIAL          ;DIAL 68MICRO JOURNAL
00840          JR      MENUXT        ;START OVER
00850 GETPHN   EQU     $             ;GET PHONE# FROM KEYBOARD
00860          LD      HL,NBRMSG     ;POINT TO NUMBER? MSG
00870          CALL    DSPLY         ;PRINT IT
00880          LD      HL,(LBUFPT)   ;POINT TO INPUT BUFFER
00890          PUSH    HL            ;SAVE IT
00900          CALL    LIPT          ;GET THE NUMBER
00910          POP     HL            ;POINT AT PHONE #
00920          CALL    DIAL          ;DIAL IT
00930          CALL    GETKEY        ;PAUSE FOR A KEYPRESS
00940          JR      MENUXT        ;START OVER
```

```
00950 DILCRB  LD      HL,CRBPHN   ;POINT TO CRABAPPLE PHONE
00960         CALL    DIAL        ;DIAL IT
00970         JR      MENUXT      ;START OVER
00980 DILMCI  LD      HL,MCIPHN   ;POINT TO MCI PHONE#
00990         CALL    DIAL        ;DIAL IT
01000         JR      MENUXT      ;START OVER
01010 DILDCS  LD      HL,DCSPHN   ;POINT AT CHRIS' PHONE#
01020         CALL    DIAL        ;DIAL IT
01030         JR      MENUXT      ;START OVER
01040 DILWSM  LD      HL,WSMPHN   ;POINT AT BUTCH'S PHONE
01050         CALL    DIAL        ;DIAL IT
01060         JR      MENUXT      ;START OVER
01070 DILPET  LD      HL,PETPHN   ;POINT AT PETE'S PHONE#
01080         CALL    DIAL        ;DIAL IT
01090         JR      MENUXT      ;START OVER
01100 BOOT    HALT                ;CAUSES BOOT
01110 DILWOM  LD      HL,WOMPHN   ;POINT AT WOMACK'S PHN
01120         CALL    DIAL        ;DIAL IT
01130         JR      MENUXT      ;START OVER
01140 HNG     CALL    HANGUP      ;HANG UP THE PHONE
01150         JR      MENUXT      ;START OVER
01160 PIC     CALL    PICKUP      ;PICK UP THE PHONE
01170         JR      MENUXT      ;START OVER
01180 DIAL    EQU     $           ;HL POINTS TO PHONE#
01190         PUSH    HL          ;SAVE IT
01200         CALL    HANGUP      ;HANGUP THE PHONE
01210         LD      BC,0        ;DO A LONG
01220         CALL    PAUSE       ; PAUSE
01230         CALL    PICKUP      ;PICK IT UP. (DIALTONE)
01240         LD      BC,0        ;DO A LONG
01250         CALL    PAUSE       ; PAUSE
01260         POP     HL          ;RESTORE PTR TO PHONE#
01270         DEC     HL          ;POINT TO BYTE BEFORE
01280         PUSH    HL          ; PHONE# & SAVE IT
01290 GETDIG  EQU     $           ;GET PHONE DIGIT
01300         CALL    CKBRK       ;BREAK EXITS TO DOS
01310         POP     HL          ;RESTORE PHONE# PTR
01320         INC     HL          ;POINT TO NEXT DIGIT
01330         PUSH    HL          ;SAVE PHN# PTR
01340         LD      A,(HL)      ;PUT IN A
01350         CP      0           ;END?
01360         JR      Z,EXIT      ;YES
01370         CP      13          ;C/R?
01380         JR      Z,EXIT      ;YES
01390         CP      30H         ;< 0 ?
01400         JP      M,GETDIG    ;YES
01410         CP      39H+1       ;> 9 ?
01420         JP      P,GETDIG    ;YES
01430         PUSH    AF          ;SAVE A
01440         CALL    BYTOUT      ;PRINT IT
01450         POP     AF          ;RESTORE IT
01460         SUB     30H         ;CHG ASCII# TO HEX#
01470         CP      0           ;ZERO?
01480         JR      NZ,HAVDIG   ;NO
01490         ADD     A,10
01500 HAVDIG  EQU     $           ;HAVE DIGIT TO DIAL
```

```
01510          LD      (PULSCT),A   ;SAVE IT
01520          CALL    DIGIT        ;DIAL THE DIGIT
01530          LD      BC,25000     ;PAUSE BETWEEN
01540          CALL    PAUSE        ; DIGITS
01550          JR      GETDIG       ;DO NEXT DIGIT
01560 DIGIT    EQU     $
01570          LD      A,(PULSCT)   ;# PULSES LEFT
01580          CP      0            ;DONE YET?
01590          RET     Z            ;YES. GOTO NEXT DIGIT
01600          DEC     A            ;SUBTRACT 1 &
01610          LD      (PULSCT),A   ; SAVE IT
01620          LD      A,(PORT)     ;PUT CASSETTE PORT #
01630          LD      C,A          ; IN C
01640          LD      A,(MODEL)    ;MODEL 1/III?
01650          CP      1            ;1?
01660          JR      Z,DIGM1      ;YES
01670          LD      A,(PORTF3)   ;GET PORT SETTINGS
01680          JR      DIGXOR       ;TURN OFF CASSETTE
01690 DIGM1    LD      A,(PORTF1)
01700 DIGXOR   EQU     $            ;RESET PROPER MODEL
01710          LD      HL,CASBIT    ; CASSETTE BIT
01720          XOR     (HL)         ;TURN OFF
01730          OUT     (C),A        ; CASSETTE MOTOR
01740          LD      BC,2200      ;PUT DURATION IN BC
01750          CALL    PAUSE        ;WITHIN DIGIT
01760          LD      A,(PORT)     ;WHICH PORT TO CHANGE
01770          LD      C,A          ;PUT IT IN C
01780          LD      A,(MODEL)    ;MODEL 1/III?
01790          CP      1            ;1
01800          JR      Z,DIGM1A
01810          LD      A,(PORTF3)   ;MODEL III PORT FLAGS
01820          JR      DIGAND       ;TO TURN ON CASSETTE
01830 DIGM1A   LD      A,(PORTF1)   ;MODEL 1 CASSETTE FLAGS
01840 DIGAND   LD      HL,CASBIT    ;MODEL CASS MTR BIT
01850          AND     (HL)         ; TO TURN IT ON
01860          OUT     (C),A
01870          LD      BC,2200      ;PUT DURATION IN BC
01880          CALL    PAUSE        ;WITHIN DIGIT
01890          JR      DIGIT
01900 EXIT     EQU     $            ;HOUSEKEEP & EXIT
01910          POP     HL
01920          RET
01930 MODEL1   LD      A,0FFH       ;LOAD PORT WITH
01940          LD      (PORT),A     ;MODEL 1 ADDRESS
01950          LD      HL,MODEL     ;PT TO MODEL BYTE
01960          LD      (HL),1       ;INDICATE MODEL1
01970          LD      HL,CASBIT    ;PT TO CASBIT
01980          LD      (HL),4       ;PUT 4 IN IT
01990          LD      HL,14308     ;SELECT CASSETTE
02000          LD      (HL),1       ; #2
02010          RET
02020 HANGUP   EQU     $            ;HANG UP THE PHONE
02030          LD      A,(PORT)     ;PUT CASSETTE PORT#
02040          LD      C,A          ;INTO C
02050          LD      A,(MODEL)    ;WHICH 'PUTER?
02060          CP      1            ;MODEL 1?
```

```
02070          JR      Z,HANG1       ;YES
02080          LD      A,(PORTF3)    ;GET CASS PORT FLAGS
02090          JR      HANXOR        ;GO TURN OFF
02100 HANG1    LD      A,(PORTF1)    ;GET MOD1 CASS PORT FLAGS
02110 HANXOR   LD      HL,CASBIT     ;MODEL CASS MTR BIT
02120          XOR     (HL)          ; TO TURN OFF
02130          OUT     (C),A         ;TURN OFF MOTOR
02140          RET                   ;FROM HANG-UP
02150 PICKUP   EQU     $             ;PICK UP THE PHONE
02160          LD      A,(PORT)      ;PUT CASSETTE PORT#
02170          LD      C,A           ;INTO C
02180          LD      A,(MODEL)     ;WHICH 'PUTER?
02190          CP      1             ;MODEL 1?
02200          JR      Z,PICK1       ;YES
02210          LD      A,(PORTF3)    ;GET CASS PORT FLAGS
02220          JR      PICAND        ;GO TURN ON
02230 PICK1    LD      A,(PORTF1)    ;GET MOD1 CASS PORT FLAGS
02240 PICAND   LD      HL,CASBIT     ;MODEL CASS MTR BIT
02250          AND     (HL)          ; TO TURN ON
02260          OUT     (C),A         ;TURN ON MOTOR
02270          RET                   ;FROM PICKUP
02280 GETKEY   CALL    CKBRK         ;<BREAK> EXITS TO DOS
02290          LD      A,0           ;CLEAR KEY BUFFER
02300          CALL    CKKEY         ;KEY PRESSED?
02310          OR      A             ;CONDITION THE FLAGS
02320          JR      Z,GETKEY      ;NO KEY PRESSED
02330          PUSH    AF            ;SAVE A
02340          LD      IX,CURSOR     ;SET UP PRINT@ POSITION
02350          LD      A,58H         ; AT
02360          LD      (IX),A        ; END
02370          LD      A,3FH         ; OF
02380          LD      (IX+1),A      ; MENU
02390          LD      A,31          ;ERASE EOF
02400          CALL    BYTOUT        ;DO IT
02410          LD      A,14          ;TURN CURSOR
02420          CALL    BYTOUT        ; ON
02430          POP     AF            ;GET A AGAIN
02440          PUSH    AF            ;SAVE IT AGAIN
02450          CALL    BYTOUT        ;PRINT IT
02460          POP     AF            ;RESTORE A
02470          CP      'Z'+1         ;LOWERCASE?
02480          RET     M             ;NO
02490          SUB     20H           ;MAKE IT UPPERCASE
02500          RET                   ;WITH KEY IN A
02510 CKBRK    PUSH    AF            ;SAVE A
02520          LD      A,(BRKROW)    ;BREAK KEY
02530          AND     4             ; PRESSED?
02540          JP      NZ,DOS        ;YES
02550          POP     AF            ;RESTORE A
02560          RET                   ;NO
02570 NBRMSG   DEFB    8             ;BACKSPACE CHARACTER
02580          DEFM    'NUMBER? '
02590          DEFB    3             ;END LINE WITH NO C/R
02600 SCREEN   DEFB    1CH           ;HOME CURSOR
02610          DEFB    1FH           ;EREOF (CLEAR SCREEN)
02620          DEFM    'A. ATC                    (<BREAK> EXITS '
```

```
02630          DEFM    'TO DOS.)
02640          DEFB    10
02650          DEFM    'B. BASIC'
02660          DEFB    10
02670          DEFM    'C. MODEM80'
02680          DEFB    10
02690          DEFM    'D. 68MICRO JOURNAL.... '
02700 M68PHN   DEFM    '842-6809'
02710          DEFB    0
02720          DEFB    10
02730          DEFM    'E. ENTER A NUMBER TO DIAL'
02740          DEFB    10
02750          DEFM    'F. CRABAPPLE........... '
02760 CRBPHN   DEFM    '875-6035'
02770          DEFB    0
02780          DEFB    10
02790          DEFM    'G. MCI................ '
02800 MCIPHN   DEFM    '1-800-323-7751'
02810          DEFB    0
02820          DEFB    10
02830          DEFM    'H. CHRIS SMITH........ '
02840 DCSPHN   DEFM    '899-5377'
02850          DEFB    0
02860          DEFB    10
02870          DEFM    'I. BUTCH............... '
02880 WSMPHN   DEFM    '1-614-695-3056'
02890          DEFB    0
02900          DEFB    10
02910          DEFM    'J. PETE............... '
02920 PETPHN   DEFM    '870-1324'
02930          DEFB    0
02940          DEFB    10
02950          DEFM    'K. BOOT THE SYSTEM'
02960          DEFB    10
02970          DEFM    'L. WOMACK BBS......... '
02980 WOMPHN   DEFM    '891-0136'
02990          DEFB    0
03000          DEFB    10
03010          DEFM    'M. HANG UP THE PHONE'
03020          DEFB    10
03030          DEFM    'N. PICK UP THE PHONE    '
03040          DEFB    03
03050 ATCPGM   DEFM    'ATC'
03060          DEFB    13
03070 BASPGM   DEFM    'BASIC'
03080          DEFB    13
03090 MODPGM   DEFM    'MODEM80'
03100          DEFB    13
03110 CASBIT   DEFB    2            ;MIII CASSETTE MTR BIT
03120 MODEL    DEFB    3            ;MIII (1=MODEL 1)
03130 PORT1    DEFB    0FFH         ;M1 CASSETTE PORT
03140 PORT     DEFB    03CH         ;MIII CASSETTE PORT
03150 PULSCT   DEFB    0            ;PULSE COUNT
03160          END     START
```

MAINTENANCE TIPS

by Dave Owen

Reprinted from Micro Info Exchange

In my work as a technician, I have encountered a number of Model III's whose Drive 0 would not even boot., let alone run the diagnostic software that I normally use to pinpoint problems. I decided to let the Floppy Disk Controller (FDC) tell me about its problems. The solution (printed below) is a BASIC program that can be easily loaded from tape or disk. It is written for the Western Digital 1793 FDC chip with port values used by Radio Shack's Models III and 4. You are free to use and modify this program as you see fit.

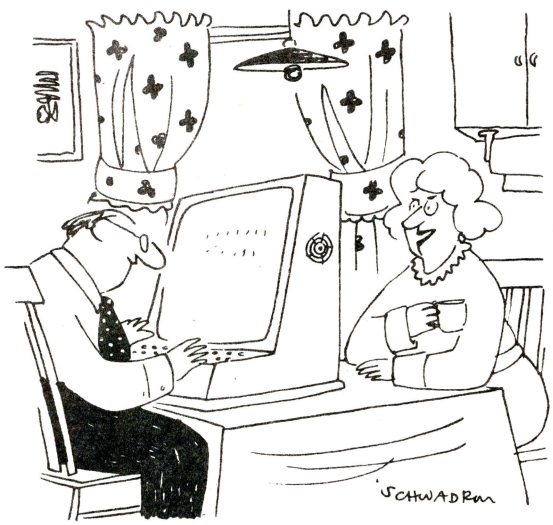
FDCIII/BAS

```
1 'FDCIII/BAS: WESTERN DIGITAL 1793 FLOPPY DISK CONTROLLER
2 'TESTER by Dave Owen, c/o CCS, POB 3032, Camarillo, CA,
3 '93011. From "Micro Info Exchange" (Cabrillo Computer
4 'Society Newsletter), August '83, page 5.
5 'If drive 0 on a Model III/4 does not boot, run this test,
6 'It checks the FDC chip--not the disk drives (except to
7 'learn if they are present and active).
8 'If the FDC passes this test, then run Floppy Doctor,
9 'which will extensively exercise the drives themselves.
10 CLS : PRINT"TEST 1: Disk Controller Ready Check...Verify
Status Bit 7 set":PRINT
20 OUT 240,208
30 GOSUB 620
40 IF ST = 255 THEN PRINT"No power applied to disk controller
chip!"
50 GOSUB 830
60 PRINT"TEST 2: Disk Status Checks...Verify Status Bit 2 set":
PRINT
70 PRINT"    ...DRIVE 0: ":B=129 : GOSUB 940
80 PRINT"    ...DRIVE 1: ":B=130 : GOSUB 940
90 PRINT"    ...DRIVE 2: ":B=132 : GOSUB 940
100 PRINT"    ...DRIVE 3: ":B=136 : GOSUB 940
110 GOSUB830
120 CLS: PRINT"TEST 3: NMI Register Checks...Verify no
errors":PRINT
130 IR = INP(228) : PRINT"   ...CONTROLLER: ";
140 IF IR = 62 THEN PRINT"No interrupt--not ready" : GOTO 190
150 IF IR = 126 THEN PRINT"No interrupt--ready" : GOTO 190
160 IF IR = 190 THEN PRINT"Interrupt requested--not ready": GOTO
190
170 IF IR = 254 THEN PRINT"Interrupt requested--ready" : GOTO
190
180 PRINT TAB(10)"ERROR--interrupt failure"
190 PRINT"   ...DISK DRIVE :0 ";:OUT 244,129 : GOSUB 880
200 PRINT"   ...DISK DRIVE :1 ";:OUT 244,130 : GOSUB 880
210 PRINT"   ...DISK DRIVE :2 ";:OUT 244,132 : GOSUB 880
```

```
220 PRINT"   ...DISK DRIVE :3 ";:OUT 244,136 : GOSUB 880
230 GOSUB 830
240 PRINT"TEST 4A:  Track  Register  Checks...Verify no errors":
PRINT
250 FOR A = 0 TO 255
260 OUT 241,A : B=INP(241) : PRINT@132,"...TRACK ";A
270 IF A <> B PRINT@196,"ERROR TRACK ";A
280 NEXT A
290  PRINT@320,"TEST  4B:  Sector  Register  Checks...Verify  no
errors": PRINT
300 FOR A = 0 TO 255
310 OUT 242,A : B=INP(242) : PRINT@452,"...SECTOR ";A
320 IF A <> B THEN PRINT@516,"ERROR SECTOR ";A
330 NEXT A
340  PRINT@640,"TEST  4C:   Data   Register  Checks...Verify  no
errors": PRINT
350 FOR A = 0 TO 255
360 OUT 243,A : B=INP(243) : PRINT@772,"...DATA = ";A
370 IF A <> B THEN PRINT@836,"ERROR AT DATA VALUE ";A
380 NEXT A
390 GOSUB 830
400 PRINT"TEST 5: Track Seeking Checks...Verify Status Bit 4 NOT
set" : PRINT
410 PRINT"   ...DRIVE 0: " : B=129 : GOSUB 470
420 PRINT"   ...DRIVE 1: " : B=130 : GOSUB 470
430 PRINT"   ...DRIVE 2: " : B=132 : GOSUB 470
440 PRINT"   ...DRIVE 3: " : B=136 : GOSUB 470
450 GOSUB 830
460 GOTO 10
470 OUT 240,B
480 OUT 244,3
490 FOR X = 1 TO 500
500 OUT 244,B
510 NEXT X
520 OUT 244,B
530 OUT 241,0
540 OUT 243,39
550 OUT 240,23
560 FOR X = 1 TO 500
570 OUT 244,B
580 NEXT X
590 OUT 244,B
600 GOSUB 620
610 RETURN
620 DN = INP(240) : ST = DN
630 IF DN - 128 >=0 THEN DN = DN - 128 : D7 = 1 ELSE D7 = 0
640 IF DN - 64 >=0 THEN DN = DN - 64 : D6 = 1 ELSE D6 = 0
650 IF DN - 32 >= 0 THEN DN = DN - 32 : D5 = 1 ELSE D5 = 0
660 IF DN - 16 >=0 THEN DN = DN - 16 : D4=1 ELSE D4 = 0
670 IF DN - 8 >=0 THEN DN = DN - 8 : D3 = 1 ELSE D3 = 0
680 IF DN - 4 >=0 THEN DN = DN - 4 : D2 = 1 ELSE D2 = 0
690 IF DN - 2 >=0 THEN DN = DN - 2 : D1 = 1 ELSE D1 = 0
700 IF DN - 1 >=0 THEN DN = DN - 1 : D0 = 1 ELSE D0 = 0
710 PRINT TAB(32)"S7 S6 S5 S4 S3 S2 S1 S0"
720 PRINT TAB(10)"FDC STATUS  REGISTER  =";  D7; D6; D5; D4; D3;
D2; D1; D0;
```

```
730 PRINT" = ";ST
740 IF D7 = 1 THEN PRINTTAB(10)"...Bit 7 set: Drive motor not up
to speed" : RETURN
750 IF D6 =  1  THEN  PRINTTAB(10)"...Bit  6 set: Diskette write
protected" : RETURN
760 IF D5 = 1  THEN  PRINTTAB(10)"...Bit  5 set: Disk drive door
open/No diskette"
770 IF D4 = 1 THEN PRINTTAB(10)"...Bit 4 set: Seek error - track
not verified"
780 IF D3 = 1  THEN  PRINTTAB(10)"...Bit  3 set: CRC error in ID
field"
790 IF D2 =  1  THEN  PRINTTAB(10)"...Bit  2  set: Disk drive at
Track 00"
800 IF  D1  =  1  THEN  PRINTTAB(10)"...Bit  1  set:  Index mark
detected"
810 IF D0 =  1  THEN  PRINTTAB(10)"...Bit  0 set: No diskette in
system"
820 RETURN
830 PRINT
840 PRINT"*** Press any key to continue ***"
850 A$ = INKEY$ : IF A$ = "" THEN 850
860 PRINT: CLS
870 RETURN
880 FOR X = 1 TO 500 : NEXT X
890 IR = INP(228)
900 IF IR = 126 THEN PRINT"Ready flip-flop set" : GOTO 930
910 IF IR = 254 THEN  PRINT"Interrupt and ready flip-flop set" :
GOTO 930
920 PRINTTAB(10)"ERROR: A problem exists--check further"
930 RETURN
940 OUT 244,B
950 FOR X = 1 TO 500
960 OUT 244,B
970 NEXT X
980 OUT 244,B
990 OUT 240,3
1000 FOR X = 1 TO 500
1010 OUT 244,B
1020 NEXT X
1030 GOSUB 620
1040 RETURN
```

"Do you have a software package on how to run a lemonade business?"



"But Bertrand! Before it was just you and your newspaper at the breakfast table."

How SPEEDSW/CMD Works

by Pete Fratus

Reprinted from SBUG, March 1984

There have been about a dozen letters and hints sent to the TRS-80 magazines about how to make your Model 4 run at 4MHz in Model III mode. None has given a solution to the occasional disk crashes caused by running at a faster speed during disk I/O. Ian Webb wrote a small program to handle the disk problem. From Ian's earlier work, I borrowed the code to make an enhanced version that handles another problem -- how to get back to III speed without booting.

Why would I want to run in the slower Model III speed anyway? Several times I have run into an application that doesn't work well in the faster speed, such as the BASIC full screen editor from CAU. XEDIT allows configuration of key repeats. When a fast repeat is chosen and I am running at 4 speed, it is difficult to move the cursor only one character at a time.

That makes editing tough!

Cassette loading and some games can also be a problem. Just about anything that runs off the clock works differently and sometimes not at all.

SPEEDSW/CMD is a small assembly language program that switches back and forth between Model III and Model 4 speed. It uses the same technique described in the Model III/4 Disk System Owner's Manual on page 80, Operating Speed.

By PEEKing or POKEing location 16912,(ORing) or (ANDing) a 64 or 191 to it, and OUTputting to port 236, the clock speed is changed back and forth from III to 4. The switching function of the program relies on the fact that the speed is always going to be one or the other. Just find out what it is now, and make it the opposite.

The disk timing part was a little trickier. Ian's program was loading a value into part of TRSDOS. I spent quite a while figureing out just what he was doing. In the book, TRSDOS COMMENTED, the code at location 4400H is labeled "Twiddle some electronic thumbs". This is a delay routine for disk I/O. When the speed is changed to 4MHz, the time spent executing these instructions is cut by roughly one half and, on occasion -- the disk becomes garbaged.

The method used to fix this was to substitute instructions that would take about twice as long, so that cutting the time in half would get it back to normal even though the CPU work was still faster.

To switch back to Model III speed, I replaced those original instructions. Examine the source listing for SPEEDSW/CMD. At lines 180-200, the code at the disk timing location (4400H) is loaded and compared to the value of the longer instructions. If the longer instructions are already there, they need to be replaced with the original shorter ones, so go to MKSHRT at line 330.

When the timing codes are the long ones, they are overlayed with shorter codes, two at a time, in lines 330-420. The count starts at eight, and is decremented by one, twice during each loop.

The only other possibility is that they are already the original ones, so make them longer at lines 210-270. It takes eight of them to cover up the old code. Notice that the count starts at eight and is decremented by one for each loop until the count is zero.

The assembly language equivalent of the PEEKing, ANDing and or ORing, and POKEing is done in two places. Depending on whether I am speeding up (ORing) or slowing down (ANDing), I execute the code at 280-320 or 430-470, respectively.

Each piece gets the old value, makes the changes, puts it back, outputs to the port, and returns to the calling program.

The /CMD file method has worked well for me, but I wanted to be able to change the speed on the fly from within BASIC. The load address of ED10H was a good place for my use, but the code is location independent, so you can move it anywhere.

The program can be loaded and jumped to in BASIC with:
CMD "L" , "SPEEDSW/CMD" , DEFUSR0 = &HED10 , X = USR0(0)
Remember to protect high memory for the location you choose. It loads and runs a lot faster if it is resolved into decimal numbers and POKEed into a magic array. Magic arrays are strings you can execute. They have been explained in many 80-MICRO, Computer User, and Basic Computing articles, as well as BASIC Faster and Better.

SPEEDSW/CMD

```
0010;SPEEDSW - Switch Model 4 (In III Mode) to 4MHz
0020;  Pete Fratus                    January 27, 1984
0030;Using the current timing code as a reference, this
0040;program toggles between III speed and 4 speed.
0050;I use Ian Webb's method of disk slow down going
0060;to 4MHz.  At 4400H, there is a series of instructions
0070;to provide disk timing.  He replaces them  with slower
0080;instructions.  When going back to III speed, I replace
0090;back the original instructions.
0100;*** NOTE *** The format overlay contains the same set
0110;of timing instructions; I would not run in 4 speed
0120;when formatting!
```

```
0130;
0140;This program is relocatable because no local storage
0150;is used.
0160;
0170          ORG     0ED10H        ;Start location
0180 START    LD      A,(DISK)      ;Get timing instr
0190          CP      LONGER        ;Is it long?
0200          JR      Z,MKSHT       ;If yes, make shorter
0210          LD      A,LONGER      ;Else, get long code
0220          LD      HL,DISK       ;Set HL to code address
0230          LD      B,EIGHT       ;Set loop count
0240 LLOOP    LD      (HL),A        ;Put in long code
0250          INC     HL            ;Next address
0260          DEC     B             ;Count - 1
0270          JR      NZ,LLOOP      ;Loop until 8 are done
0280          LD      A,(SPEED)     ;Get old speed
0290          OR      BIT6          ;Turn 4 speed bit on
0300          OUT     (PT236),A     ;Tell CPU what you did
0310          LD      (SPEED),A     ;Put new speed back
0320          RET                   ;Go back to DOS or BASIC
0330 MKSHRT   LD      DE,SHRTER     ;Get short code
0340          LD      HL,DISK       ;Get code address
0350          LD      B,EIGHT       ;Set loop count
0360 SLOOP    LD      (HL),D        ;Put short code in address
0370          INC     HL            ;Next address
0380          DEC     B             ;Count -1
0390          LD      (HL),E        ;Put other short code in
0400          INC     HL            ;Next address
0410          DEC     B             ;Count - 1 again
0420          JR      NZ,SLOOP      ;Loop until eight are done
0430          LD      A,(SPEED)     ;Get old speed
0440          AND     NOT6          ;Set 4 speed bit off
0450          LD      (SPEED),A     ;Put new speed back
0460          OUT     (PT236),A     ;Tell CPU you did it
0470          RET                   ;Go back to BASIC or DOS
0480 EIGHT    EQU     0008H         ;Counter value
0490 BIT6     EQU     0040H         ;Mask 01000001
0500 NOT6     EQU     00BFH         ;Mask 10111111
0510 LONGER   EQU     00E3H         ;OPCODE EX (SP),HL
0520 SHRTER   EQU     0F5F1H        ;OPCODES PUSH AF, POP AF
0530 PT236    EQU     00ECH         ;PORT 236
0540 SPEED    EQU     4210H         ;Location of speed bit
0550 DISK     EQU     4400H         ;Location of timing code
0560          END     START
```

Two new products from Tandy
One good and one not so good
By George Saladino

This month I would like to mention two new products that I have used. The first one is the new CCR 82 tape recorder from Radio Shack. It is much smaller than the previous recorders it replaces and has some very nice features for the computer user. It measures only 7" long by 4 1/2" wide and 1 1/4" thick! A neat little package

Features that the recorder has that I like are:

1. A override of the remote connection. A switch lets one advance the tape in play mode while the remote plug is in from the computer.

2. A switch that lets you turn on the speaker as a monitor.

3. Appears to be built better than the previous recorders and does better with loading computer tapes the first time!

4. As I said before, it is much smaller than the previous recorders and will fit nicely in a brief case with the Model 100.

The second product I have used is Scripsit 100. Yes, a type of Scripsit for the Model 100! This package catalog number is 26-3830. I must note here that it is used in conjunction with the "Text" software built into the Model 100. It is an add on package. This new product I rate fair. More on this later.

The manual states that it can -

Set margins
Print headers and footers
Set line spacing
Print page numbers
Print more than one copy of your text automaticlly
Set the lines per page
Send the output to the printer jack or RS232 jack.
Set global printer commands.
Underline
Indent paragraphs
Center text

These are just some of the features that this Scripsit for the Model 100 has. They are set in a "DO" file called "WSPEC" or put into the text with special control codes.

The program comes on tape, but can be used on the disk system. The manual does not say this, but I have it running on a disk system. More on this in awhile. The programs that come on tape are the following.

WSPEC - Mentioned above. This is a "DO" file that is used to control the printer output. You modify it for your own needs once loaded into the 100.

CORTNS - A basic program that loads in one of the two machine language programs that gives the package its speed.

CORTN1 - One of the two machine language programs mentioned above. This is the one to use with 32K 100's. (Not the one to use with the disk system however, since it will zap your disks!)

CORTN2 - The second machine language program to be used in 100's that are less than 32K have at lease 16K. (And with the disk system)

WRITE - A basic program that prints out your text files.

To use, Load in the WSPEC file and change it to your needs. Then load in the basic "Cortns" program and it will load in the right machine language program for your 100.

Lastly, load in "Write" and you are ready to print out your text. You enter your text in the Model 100 with the "TEXT" program built into the 100 and then run the "Write" program.

For disk users, you should only use the 2nd machine language program called "Cortn2" so you do not zap your disks. Once this file is loaded in, you do not need the basic program on your disk. By the way, you can load the machine language program on disk from memory by doing the following.

1. Type in LoadM"Cortns" (Note when the basic program loads in "Cortn2" save it to memory as per your manual, but call it "Cortns")

2. After loading in the file type in
SaveM"0:Cortns.Co",58073,58262
and the file will be saved to disk! You can reverse the process and take it off disk and put it into memory any time you want.

I stated before that I rate this Scripsit only fair. I found it fine printing out short text, but a longer text, the program would put several lines between the text at unexplained times. I have not figured out why yet. Also, If I tried to take a file right from disk instead of loading it into the memory of the 100 first, the program would start printing the file and then stop with a "Input Past End of File" error. This same file would print out if loaded into the memory of the 100 first!

I have not tried all the features, but I feel that there are more bugs in the software than the ones I found.

Buy the tape recorder! It is a nicely made recorder, I would wait with Scripsit.

A s k A l a n

by Alan Abrahamson

Being well known as somewhat of a deviot, this months column shall deviate from the normal Q & A format. I have for some time now, been doing lots of work with text editing or word processing as some may know it better. I also have had the occaision to use my text from the word processor within my Basic program. This type of situation was normally used as a sequential ASCII file, saved in ASCII from the word processor, and then used from within the Basic program as text for directions or instructions within the program. I beleive that I have explained the method of doing this in a prior column.

Well this time the "end" was different and the "means" to achieve this "end" was entirely different. I needed a method to create text. Well that was easy and straight forward using my trusty word processor program. But now I wanted to take that sequential ASCII data and transform it into a random access file so that I could select any "block" (record) of text from the file at will. I mused the problem for a bit and came up with the following program to accomplish the task.

The program is heavily commented and should not be difficult for any of you to decipher. I don't know if others need this capability. I do know that it was useful to me to create help screens with my word processor, block the text into records of 256 bytes (4 lines of text of 64 characters per line included in each record), and then use this utility to move the sequential data into a random record access file. I could now select any record in the file with a standard random file OPEN and GET. On to the program.....

SEQTORAN/UTL

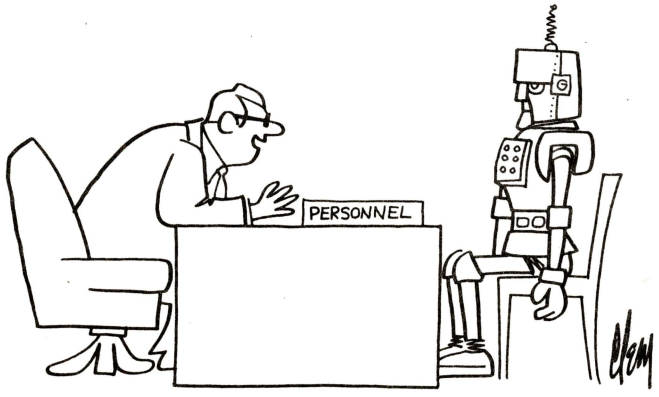
```
1 REMarkable     SEQTORAN/UTL by Alan Abrahamson (05/30/84)
                 Takes text file from word processor. Each line
                 must end with (C/R) & blocks it into a random
                 file record, 4 lines per each 256 byte record.
10 CLEAR 1000 : CLS
   'Clear string space, wipe off screen.
12 LINEINPUT"Text file name ";F1$ : LINEINPUT"Help file name
   ";F2$
   'Get word processor filespec & Help file filespec.
20 OPEN"I",1,F1$
   ' Open Text file in buffer 1.
30 OPEN"R",2,F2$
```

```
   ' Open Help file in buffer 2.
35 FORX=ØTO3 : FIELD2, (64*X) AS DU$, 64 AS B$(X) : NEXT
   'Field buffer for 4 screen lines of 64 characters each.
36 FORX=ØTO3
   'Start loop
38 IF EOF(1) THEN CLOSE : END
   'Check for EOF and end if true.
4Ø LINEINPUT#1,A$
   'Read sequential record to Carraige Return.
42 LSET B$(X) = A$ : NEXT
   'Set sequential line into buffer variable.
67 FORX = Ø TO 3 : PRINT B$(X); : NEXT : PRINT
   'Show what you are going to file to disk.
7Ø PUT 2, X1 + 1
   'Put record to disk. X1 will equal Ø initially.
8Ø X1 = X1 + 1
   'Increment X1
9Ø GOTO35
   'Go do it again at 35.
6ØØØØ SAVE"SEQTORAN/UTL
      'Alan's SAVE line
```

Now that you have the random help file created. It is a simple task to display that data from within your Basic program. I shall show you with the following code, one way to do this.

DISPLAY/BAS

```
1Ø CLEAR 1ØØØØ : CLS
   'Make some string room, clear CRT
2Ø OPEN"R",1,"HELPFILE"
   'Open your HELPFILE created with SEQTORAN/UTL
3Ø FIELD 1, 128 AS A$, 128 AS B$
   'Field 2 variables to receive text
4Ø LINEINPUT "Which record shall I get for you? "; RN$
   'Prompt for help file record number
5Ø RN = VAL(RN$) : GET 1, RN : PRINT A$ : PRINT B$ : GOTO 4Ø
   'RN=Record number, GET record, Print buffer data out of A$
    and B$ and then go back to 4Ø for next prompt.
```



"Your ETS scores are excellent, but we feel that your personality needs a bit more polish."

More Memory on Small Items

by Bob Schwabach

Reprinted from Knight-Ridder Newspapers

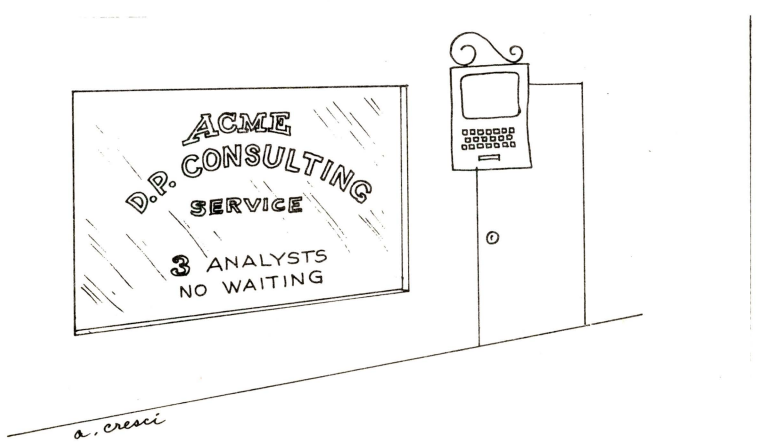
The hottest new item in the past year has been the lap-sized computer. Newspapers have been buying them for reporters in the field, and it is considered very chic for businessmen to pull out their Radio Shack Model 100's on the train or in the airport and jot down a few meaningful transmissions to the home office. Despite their light weight and ease of use, these machines are far more limited than a desk-sized personal computer. This is about to change. You knew it would.

Hewlett Packard is first off the mark with the announcement this month of the HP-110, a completely portable 8 1/2-pound computer the size of a notebook. It uses a 16-bit 8086 processor (much faster and more efficient than the IBM-PC 8088 processor), and has more than half a megabyte of memory. Memory is divided up into 272 kilobytes of user RAM (Random Access Memory) and 384 kilobytes of ROM (Read Only Memory).

The enormous amount of ROM (roughly 10 times what most computers have) is utilized by Hewlett Packard to provide major computer programs resident in the machine. Lotus 1-2-3, MemoMaker or Word-Star, MS-DOS and a Terminal communications program are already "on-board," for example, when you turn on the machine. A modem, for communication with other computers over phone lines, is built in.

The machine is not cheap, but has the normal ultra-high quality of HP equipment. Price is $2,995 for the computer itself, and $795 for an external disk drive if you want one for large-scale storage. HP's new ink jet printer, "ThinkJet," is $495, and plugs right in.

Contact a dealer, or Hewlett packard Personal Computer Group, Cupertino, CA., 95014.



INEXPENSIVE OVERVOLTAGE PROTECTION FOR ELECTRONIC EQUIPMENT

by Herman Parks

You've saved all your life to buy your computer, and in one millisecond, a spike on the power line reduces it to a smoking heap of charred silicon. Judging by the interest shown at the July meeting this is a sad but true tale for some of FCUG's members.

Probably the soundest advice given during the discussion was "Pull the plug whenever you are not operating." In addition there are protective devices for insertion between the outlet and the equipment. My discussion here will describe an inexpensive version of an overvoltage protector.

It began with my reading of an article by Steve Ciarcia in Byte Magazine, Dec. '83, "Keep Power Line Pollution Out of Your Computer". I tucked the thought away as a future project until a few months ago, Caldor's featured a sale of Electripak Multiple Outlet Centers, an outlet strip featuring six 15-Amp, 3-wire outlets, a switch, a circuit breaker, and an indicator lamp - all nicely housed in a steel box. These would a very good buy at their normal Caldor price of $18.99 and, to me, irresistible at $12.88. (They have been on sale since).

Back to Ciarcia's article. After reading many words of theory and discussion, I came to the conclusion that the first, easiest, and most beneficial thing to do was to install MOV's (metal oxide varistors) across the three supply lines, as Ciarcia recommends. Other topics included lightning rods, proper grounding of the power system, and protection against RFI (radio frequency interference).

The MOV is a ceramic disk about the diameter and thickness of a nickel. It has two leads coming out of the edge. It looks just like a ceramic capacitor. When connected across a voltage source, an MOV will act as a very high resistance up to its voltage rating. Above this voltage rating it quickly starts to conduct, and continues to do so until the voltage once again falls below the rated value. So if you install an MOV across the lines supplying your computer, any transients or spikes coming in will be shorted out before they enter the equipment.

Ciarcia recommends that three MOV's be installed - between the hot-line and neutral; hot-line and ground; and neutral and ground. (Some commercial devices such as the Radio Shack Model 61-2790 in-line Voltage Spike Protector only put an MOV between line and neutral.

Obtaining a supply of MOV's (I converted three strips) proved a small difficulty. Ciarcia suggested using Radio Shack 276-370 MOV's, but with RS's new computerized stocking system, store manager's get ample warning to move stock out of sight before I get in the door. Three stores and not one MOV, so I ordered a

package of 10 Panasonic "ZNR" Type D Transient/Surge Absorbers (Digi-Key Part No. P7064) from Digi-Key Corp. Highway 32 South, P.O. Box 677, Thief River Falls, MN, 56701, (a long time electronic parts house, which really knows how to do mail order, and doesn't seem to sneer at chicken orders). One to nine are $1.35 each, ten are 11.50, add $2.00 for handling and shipping up to 9.99, 2.75 up to 24.99. There are cheaper ones with lower surge ratings, but I say, "Go first class".

Installation was easy using the Electripak outlet strip. The wiring face of the dual-outlets have holes into which the connecting wires are pushed (the ground wire attaches under a green screw head). The third dual-outlet uses only one of four quick-connect holes provided for each of the hot-line (black wire) and neutral (white wire) connections. Into these I inserted short lengths of bare No. 14 solid copper wire (this spec is important to the working of the quick-connect mechanism) so that I had two upward pointing connection posts. A solid wire was attached to the ground screw so that it also pointed up. The three MOV's were then soldered to each pair of these posts as close to the surface of the outlet as possible.

The posts were then cut close to the solder points. Finally, I checked for adequate clearance to the cover by laying a straight-edge over the top of the unit's mounting ears.

The following sketch should make the arrangement clear.

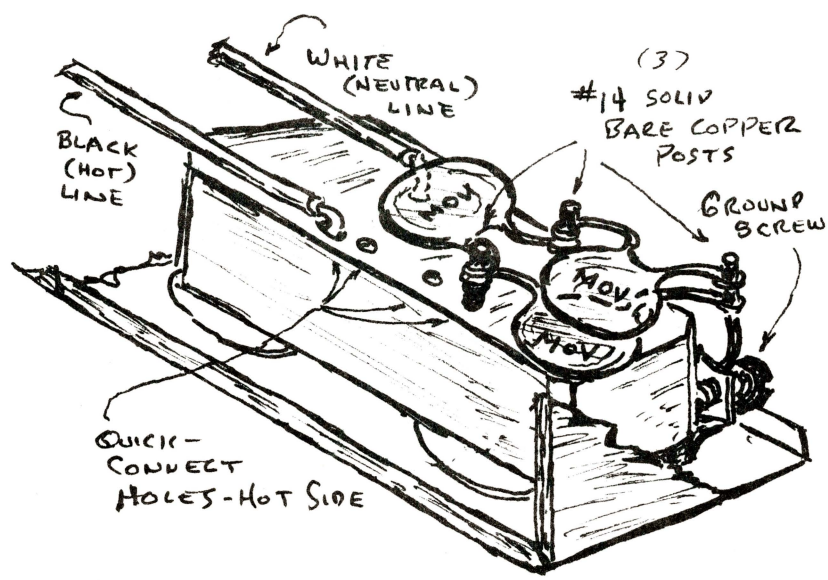


FIG. 1
DUAL RECEPTACLE WITH MOV's INSTALLED.

The President's Page

Date = Tuesday, August 7, 1984

Time = 7:30 PM

Place = New Canaan Country School

STOP THE PRESS - I've just been told that our application for incorporation has been granted and that by the August meeting date we should be:

FAIRFIELD COUNTY COMPUTER USERS GROUP, INC.

One of our first acts as an "Inc." will be to again nominate and elect officers. The same slate will be presented to the members, and additional nominations will be accepted from the floor. Our attorney, Charles Lee, will be at the meeting to brief us on our new status.

* * *

Our beloved president continues with his three month vacation - and I think it is a well-deserved one. The only argument I have with him is his choice of where to spend it. It appears that he has decided to spend his R & R time at the New Canaan Country School. So if you happen to see him lounging around the meeting room, try not to bother him with FCUG-stuff. You wouldn't want the office bugging you on your time off!

Two of our most knowledgeable lecturers will be featured in the August meeting. We have been sniffing about the topic of bulletinboards for several months now, and a full scale talk on the whole subject of setting up a club bulletinboard is overdue. Dick Carricato, who has volunteered to set up a BB, will tell us what advantages we can expect, how the bulletinboard operates, and what hard and software we need to make use of it.

Roger Giler found out a long time ago that graphic representation of business data gets the story across when tables and narratives fail. He will describe how he uses graphics generating programs, his own and commercial ones, to convert tabular data, and show us some of his impressive samples.

Herman

Voice of the '80
Index for August

DIALER ..... Page 1
MAINTENANCE TIPS ..... Page 7
Cartoons ..... Page 9
SPEEDSW/SRC ..... Page 11
Two new Tandy products Page 13
SEQTOTAN/UTL ..... Page 15
More Memory on HP-110 Page 17
The President's Page . Page 20
DIALER/SRC ..... Page 1
FDCIII/BAS ..... Page 7
How SPEEDSW/CMD Works Page 10
Cartoon ..... Page 12
Ask Alan ..... Page 15
DISPLAY/BAS ..... Page 16
Overvoltage Protection Page 18

The Voice of the '80 is a publication of the Fairfield County TRS-80 Users Group. Permission to reprint is granted for non-profit and non-commercial uses. Credit is appreciated. Submissions should be sent to the Editor at the address below:

PRESIDENT: Robert Jackson VICE PRESIDENT: Herman Parks
SECRETARY: Patrica Brinson TREASURER : Robert Gould
EDITOR : Alan Abrahamson ASSOC. EDITOR : Izzy Available

Ask Alan HOTLINE: Evenings (6PM-10PM) 203/866-7883

This Newsletter printed by Sir Speedy Printing Center, Wilton CT

{{{{{{{{ This Months cover art by Michael Ciferri }}}}}}}}

Licking, sticking, folding & stapling by: Herman and Jeff Parks

FAIRFIELD COUNTY TRS-80 USERS GROUP
% Alan Abrahamson
10 Richlee Road
Norwalk, CT 06851

FCUG

FIRST CLASS

TO:

First Class